LINE (large-scale Information Network Embedding)

3个已有的Graph Embedding方法：MDS , IsoMap, Laplacian eigenmap 的局限性（话说找的方法还真是有点早啊，Laplacian eigenmap 那个方法01年的了）。说他们都是只能处理小规模网络，不能处理大规模网络。他们的时间复杂度至少是平方级别的。

* 局部信息是指，网络中的边信息，它是一个观测到的一阶相似度，之前很多方法就用了这个信息，如IsoMap，Laplacian eigenmap 。但是实际上在现实网络中，很多合法的边没有被观测到。所以一阶相似度并不能有效的表示全局信息。
* 所以作者就提出了，二阶相似度，不是通过节点周围点的连接强度，而是通过与其他节点间共享邻居节点的结构来表示信息。直觉上，节点之间share的邻居节点越多，可以认为他们越相似。

二阶相似度能完善一下只用一阶相似度带来的稀疏性问题。

所以优化的目标就出来了，但是怎么去优化它也是一个挑战。   
最近常用的方法就是梯度下降了，但是作者发现，直接用SGD会有一些问题。作者分析原因在于，如果网络带权的话，通常情况下，带权网络的权值是一个方差非常大的分布，而大的方差，导致了最后乘以梯度的时候，会导致梯度爆炸，而使结果不好。   
所以作者说他们提出了一个边采样的方法。按照概率分布采样，而概率是通过边权大小计算出来的，然后将边当成一个二元的边(只有src 和dst,不用考虑权重了)来处理。有了这样的处理之后，优化的目标函数还是一样，但是就没有梯度爆炸的问题了。

最近的一些工作，如graph factorization，它的问题在于

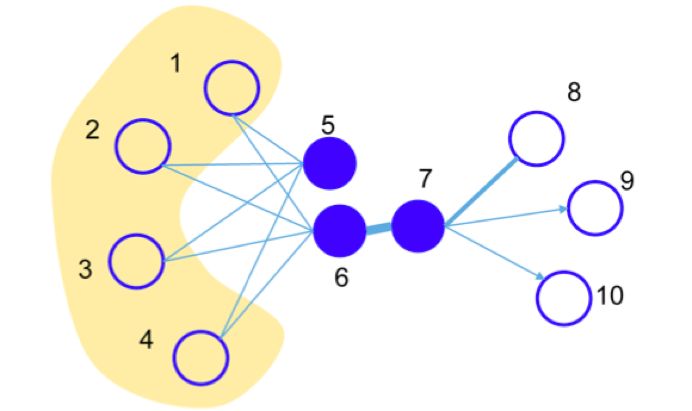
* 并不是专门为了网络而设计的
* 只有一阶相似度
* 只能处理无向图

然后另外一个相关的工作是Deep Work，它是用随机游走的办法，主要应用了二阶相似度，但是主要的问题在于：

* DeepWork用随机游走是一个DFS，而我们的方法是一个BFS，更能利用二阶相似度
* 只能用在不带权的图中

LINE算法利用图中已存在的边构造目标函数，该目标函数显式描绘了**一阶和二阶**的邻近关系。然后通过优化方法去学习点的表达向量，其本质上是一种关于边的平滑，即很多很可能存在的边实际上不存在，需要模型去学习和预测出来。这点类似于推荐，任何推荐的算法本质上是对于user-item关系矩阵的平滑。

**LINE能用于有向，无向图，带权重的和不带权重的。**



如上图，6，7两个点是相似的，是因为直接相连，而5，6两个点也是相似的，是因为共享了很多邻近点。故本文显式考虑了一阶和二阶邻近关系。

**一阶相似度只能用在无向图上**

**二阶相似度**

二阶相似度可以用在无向和有向图上