数据挖掘导论期末作业

题目：**标题**

姓名	赖培文、李赞辉、梁允楷
学号	18342041、18342053、18342055
专业	软件工程
授课老师	梁上松

目录

1 引言 **2**
1.1 图嵌入的背景和意义 2
1.2 问题描述 2
1.3 本文工作 3

2 DeepWalk 方法的简单介绍 **3**
2.1 对图嵌入问题的进一步讨论 3
2.2 DeepWalk 方法的介绍 4
2.2.1 DeepWalk 4
2.2.2 RandomWalk 4
2.2.3 word2vec 4

3 基于 DeepWalk 算法的进一步讨论 **5**
3.1 基于度数对 DeepWalk 的讨论 5
3.2 基于度数讨论的结果分析 6
3.3 基于路径对 DeepWalk 的讨论 7
3.4 基于路径讨论的结果分析 7

4 相关工作陈述 **8**
4.1 基于因子分解的方法 8
4.2 基于随机游走的方法 8
4.3 基于深度学习的方法 8

5 总结与展望 **9**

1 引言

1.1 图嵌入的背景和意义

众所周知，图是一种基础且常用的数据结构，它使用点和边的集合描述不同实体之间的关系。图广泛地存在于各种现实应用中，可以描述复杂的现实场景，如社交网络、交通网络、通信网络等，甚至生物中的蛋白质相互作用、一个句子都可以使用图结构进行描述。通过对图的分析，我们可以深入了解社会结构、语言和不同的交流模式，因此图一直是学界研究的热点。

使用图结构可以解决很多实际中的问题，如：社交网络中的关系预测、通信网络中异常节点的预测和识别、蛋白质功能的模拟等等。将这些任务做进一步的抽象，图分析任务可以大致分为以下四类：（a）节点分类，（b）链路预测，（c）聚类，以及（d）可视化 [1]。其中，节点分类旨在基于其他标记的节点和网络拓扑来确定节点的标签；链路预测是指预测缺失链路或未来可能出现的链路；聚类用于将相似节点聚合在一起；可视化有助于深入了解网络结构。

图由边和节点表示，如果直接使用这些关系表示图的信息，一般只能使用数学或统计的方式进行数据处理。如果将图转换到向量空间表示，则有更加丰富的方法可以对图进行处理。对于一般的应用，图使用 $|V| \times |V|$ 的邻接矩阵表示，但这样的方法对于现实生活中的大型图结构进行表征几乎是不可能的。一方面使用邻接矩阵表示图的信息，空间复杂度超过企业可以接受的范围；另外一方面，这样的编码方式并不能很好地反映图节点的信息。因此，使用图嵌入技术将图中的节点以低维稠密向量的形式进行表示，实际上将图的信息进行了压缩，打包在一个维度更小的向量中。这要求图嵌入的技术对节点映射为一个低维向量时，尽可能地保留节点的拓扑信息，在原图中接近的节点在低维特征空间也同样比较接近。

1.2 问题描述

对于图 $G = (V, E)$，节点集合 $V = \{v_1, \ldots, v_n\}$，边集合 $E = \{e_{ij}\}_{i,j=1}^{n}$。图嵌入是图节点的映射，映射 f 在将节点映射到低维特征向量的同时，尝试保留节点之间的拓扑信息。

$$f : v_i \to y_i \in \mathbb{R}^d,\ \forall i \in [0, n],\ where\ d \ll |V|$$

节点之间的拓扑信息有多种方法可以衡量，其中一种常用的方法为使用节点之间的距离表示节点之间的拓扑信息。即在原图中两个节点之间有邻接关系或者两个节点距离较近，在映射后在低维向量空间中对应两个向量的距离同样比较接近，在这里衡量两个向量距离的接近程度通常使用两个向量的内积。因此，当前节点对应的拓扑信息，可以使用其邻居节点进行表示。

对于这一问题，DeepWalk 是其中一个最经典的方法，这个方法的思路和上述使用邻居节点表示节点的拓扑信息有异曲同工的地方。DeepWalk 使用的随机游走策略将在当前节点的邻居中等概率地选择下一个要访问节点，这实际上仅利用了节点之间的连接信息来提取节点序列，在一定程度上保留了节点的拓扑信息。虽然 DeepWalk 是 KDD 2014 的工作，但它为我们实现图嵌入提供了很好的思路，是我们了解图嵌入无法绕过的一个方法。

表 1: Important Notations

u	DeepWalk 算法中随机游走讨论的当前节点
V_u	当前节点 u 对应邻接节点的点集
v_i	当前节点的邻接节点点集中下标为 i 的节点
$\|n\|$	节点 n 的度数，即 n 的邻接节点的个数

1.3 本文工作

本文主要讨论了图嵌入问题并深入了解 DeepWalk 算法的思路和实现，并且对 DeepWalk 进行一系列的讨论，尝试对不同类型的图改进 DeepWalk 中节点序列的提取的策略，根据节点以及连接之间更多的信息来指导节点的选择：

- 根据采样节点与起始节点的距离，调整返回上一节点的概率。
- 根据采样节点的邻居的度数，调整选择该邻居作为下一节点的概率。

本文的其余部分安排如下。在第二节中，我们将介绍图嵌入算法的发展。在第三节中，我们将讲述 Deepwalk 算法原理和改进。在第四节中，我们将展示和分析实验结果。在第五节中，我们将做出总结并给出未来的改进方向。

2 DeepWalk 方法的简单介绍

2.1 对图嵌入问题的进一步讨论

图嵌入的核心目标在于将图上的节点映射到一个低维向量上，这一向量可以保留节点的拓扑信息。这个问题本质上是将一个高维稀疏的向量映射到一个低维稠密的向量上，并且能够保留相应的信息量。并且值得一提的是，这样的映射成立有其对应的原因，对于现实中的稀疏图，使用邻接矩阵存储网络的信息可能存在大量的信息冗余。一个节点的信息体主要体现在和它的邻居节点，以社交网络为例，其中存在两个常见的现象：

- 社交平台上存在一些大 V，他们的影响力较大，和大量的节点存在邻接关系。
- 社交平台上存在不同的圈子：圈子内部的用户可能相互关联互相关注，甚至可能上述提到的大 V 就可能是圈子中的一个代表元；圈子之间的用户可能关联较少，甚至可能存在相互独立的子图。

上述提到的两个现象，尤其是圈子现象说明了使用邻接矩阵表示节点之间的关系可能存在大量的数据冗余。特别是两个相互独立的圈子（在一些关联较弱的圈子内部也可以近似看作两个相互独立的圈子）之间的用户代表两个相互独立的子图，在邻接矩阵中占据了冗余的信息空间。因此，使用节点的邻居节点的信息可以极大程度上表示节点的拓扑信息，在社交网络中可以对应为用户的圈子信息。

而且在上述的社交网络模型中，使用向量的内积作为衡量两个向量相似的指标，可以很好地解释上面的两个现象。通常而言，两个向量的夹角较小可以看作两个用户在同一个圈子内部；而向量的模比较大可以看作用户的影响力比较大，在其方向上和多个向量的内积都比较大，容易和多个用户产生关联。

综合上述两点，根据节点的邻接信息构建向量，进而描述图的相关信息有其的合理性；并且使用向量的内积作为衡量两个向量的相似程度，也十分具有解释性。

2.2 DeepWalk 方法的介绍

2.2.1 DeepWalk

DeepWalk 使用随机游走的方式在图中进行节点采样，得到节点与节点的共现关系，再使用 word2vec 的方法，用游走序列作为数据训练 skip-gram 模型，得到节点的向量表示。

Deepwalk 算法主要包含两个部分：一个随机游走序列生成器和一个更新过程。

- 随机游走序列生成器首先在图 G 中均匀地随机抽样一个随机游走 W_{v_i} 的根节点 v_i，接着从节点的邻居中均匀地随机抽样一个节点直到达到设定的最大长度 L
- 对于一个生成的以 v_i 为中心左右窗口为 w 的随机游走序列 $\{v_{i-w}, \ldots, v_{i-1}, v_i, v_{i+1}, \ldots, vi + w\}$，DeepWalk 利用 skip-gram 算法通过最大化以 v_i 为中心，左右 w 为窗口的同其他节点共现概率来优化模型：

$$\Pr(\{v_{i-w}, \ldots, vi + w\} \backslash v_i | \Phi(v_i)) = \prod_{j=i-w, j\neq i}^{i+w} \Pr(v_j | \Phi(v_i))$$

2.2.2 RandomWalk

随机游走从某个特定端点开始，等概率地游走到其邻接点之一，不断重复地随机选择游走节点直到得到指定长度的游走序列。使用随机游走去捕获图中节点的局部上下文信息，得到的游走序列可以反映节点在图中的局部结构特征。两个节点在图中共有的邻接点越多，则对应的游走序列就越相似，训练得到的向量之间的距离就越短。

随机游走在算法性能上具有优势，局部的游走在全局的网络中并不会相互影响，因此可以并行计算多个随机游走，且随机游走关心局部结构的特征，网络的局部变化只会影响局部范围内的随机游走结果，不需要重新计算全部特定端点的随机游走。

DeepWalk 采用截断随机游走，又称长度固定的随机游走。在实际实验中，随机游走的截断长度是非常重要的参数，长的随机游走在采样分布上会包含更大的范围，为模型训练提供更大范围的局部结构特征，但加大计算复杂度，同时使网络局部的变化可以影响更多的随机游走结果。

2.2.3 word2vec

word2vec 用于词嵌入，目的是让训练后的模型可以将每个单词映射到向量上，从而表示单词之间的关系；该方法基于词袋模型（Bag-of-words model），在该模型下，根据句子或文档的词出现频率进行训练，而不考虑文法和单词顺序。

在实验中，根据统计结果，节点共现频率和词汇共现频率相似，该结果表明：网络中随机游走的分布规律与语料库中句子序列出现的规律有着类似的幂律分布特征，游走的序列可以类比语料库中的句子，序列中的节点可以类比句子中的单词，随机游走序列中节点的共现情况类比于词汇的共现情况。在理论上，word2vec 不考虑文法与词序的特性也更符合网络节点以邻近性为特征采样而得到的结果。故 DeepWalk 将 word2vec 的方法应用于网络节点间，类比 word2vec 处理单词序列的方式来处理随机游走得到的节点序列。

3 基于 DeepWalk 算法的进一步讨论

在上面的讨论中可以得知，DeepWalk 使用随机游走的方式对邻居节点进行采样，实现的效果为距离当前节点较近的邻居节点被采样的概率比较大，距离当前节点比较远的邻居节点被采样的概率比较小。根据这一特性，我们可以对 DeepWalk 的采样策略进行详细的讨论，对不同类型的图作出进一步的讨论，依据不同类型的图结构，选择合适的采样方法使得 DeepWalk 的准确率尽可能提升。从采样策略和图的性质出发，我们提出了两个角度对采样策略进行讨论：基于图节点度数的改进方式、基于路径的改进方式。

3.1 基于度数对 DeepWalk 的讨论

考虑图的度数对 DeepWalk 采样策略的影响，由于随机游走策略是从当前节点随机选择一个邻接节点进行移动，然后在该邻接节点上继续迭代算法，我们可以得知不同的节点度数对 DeepWalk 的采样行为造成不同的结果。我们可以考虑如下场景：程序进行随机游走，如果新到达的节点的度数比较大，则程序随机游走选择往回行走，即前往上一个节点的概率比较小；相反如果新到达的节点度数比较大，则程序随机游走选择往回行走的概率比较小。

根据上述的性质，我们可以设计不同的采样策略对度数较大的邻接节点和度数较小的邻接节点进行一定的权衡比较。我们可以依据这一想法，设计出依据当前节点的邻接节点度数对选择下一个游走节点的概率进行调整的两种方法：优先选择度数更大的邻接节点、优先选择度数更小的邻接节点。

设计优先选择度数较大的邻接节点，可以很容易可以想到实现的思路。我们可以直接依据不同节点的度数占比作为选择节点的概率，可以使用以下公式进行表述，其中 $p(v_i)$ 为选择节点 v_i 作为随机游走的目标点的概率：

$$p(v_i) = \frac{|v_i|}{\sum_{j=1}^{|u|} |v_j|},\ \forall v_i, v_j \in V_u$$

设计优先选择度数较小的邻接节点，有多种概率映射的方式，只要最终达到概率和邻接节点的度数呈负相关即可。我们依据取相反数和取倒数两个角度，提出了以下三种映射方式：

以相反数的角度出发，我们可以使用常数 C 减度数占比的方式再进行归一化，求出各个邻接节点被选择的概率。以公式化的语言进行描述，可以得到以下表达式：

$$p(v_i) = \left(C - \frac{|v_i|}{\sum_{j=1}^{|u|} |v_j|} \right) / \left(C|u| - 1 \right),\ \forall v_i, v_j \in V_u,\ C > 1$$

在具体实验中我们取 $C=2$ 而不取 $C=1$，避免由于邻接节点只有一个邻居，即 $|u|=1$ 的情况，导致概率中分子项为 0 的情况，具体的概率表达式为：

$$p(v_i)=\left(2-\frac{|v_i|}{\sum_{j=1}^{|u|}|v_j|}\right)/\left(2|u|-1\right),\ \forall v_i,v_j\in V_u \tag{1}$$

当然，我们也可以取 $C=1$ 的情况，在这种情况下需要我们对分子为 0 的这一情况进行特判，具体的概率表达式为：

$$p(v_i)=\begin{cases}\left(1-\frac{|v_i|}{\sum_{j=1}^{|u|}|v_j|}\right)/(|u|-1), & |u|>1\\ 1, & |u|=1\end{cases} \tag{2}$$

从倒数的角度出发，我们可以将邻接节点的度数映射为 $\hat{p}(v_i)=\frac{\sum_{j=1}^{|u|}|v_j|}{|v_i|}$，这样同样可以达到选取的概率随着节点度数的增加而减少。因此，我们可以设计出以下的概率映射关系：

$$p(v_i)=\frac{\hat{p}(v_i)}{\sum_{k=1}^{|u|}\hat{p}(v_k)},\ \text{where}\ \hat{p}(v_i)=\frac{\sum_{j=1}^{|u|}|v_j|}{|v_i|} \tag{3}$$

3.2 基于度数讨论的结果分析

我们在上节通过度数对 DeepWalk 的采样方法进行了一系列的讨论，主要提出了依据邻接节点的度数选择不同的方案。我们猜想这两种方法在不同类型上的图会有各自擅长的表现。例如对一些大型社交平台，优先选择度数较大的邻接节点，可能会有更好的表现。依据我们的猜想，在大型的社交平台中人们更多的是表现自己的主要属性，在这样的网络上人们更多的是关注自己的感兴趣领域的一些大 V。因此，通过这些度数比较大的邻接节点可以更好地表现节点的信息。相反，对于一些小型的社交平台，特别是一些亚文化的社交平台，人们关注的更多是一些更具个性化、更具特色的圈子。这样的圈子可能没有明显的大 V，更多的是用户之间圈地自萌，主要体现在用户和用户之间相互关注，一般并不会主动跨圈关注其他大 V。因此，对于一些小型社交平台更适合使用度数较小的邻接节点作为采样对象。

为了验证我们的想法，我们使用了三个数据集对原方法和变体的方法进行了比较。每个方法在每一个数据集上运行了十次，并且最后求出平均值。得到的结果如表 2所示：

通过表格可以看出在机场数据集中，无论是 usa-airport 还是 europe-airport 数据集，优先选择节点度数大的方法效果比原本的方法要好，优先选择节点度数小的方法得到的效果比原方法效果要差或基本持平。我们可以使用上述的猜想进行解释。在机场数据集下，机场具有的标签更多是从属于比较大的机场，即小机场的飞机通常多飞往大机场，这个时候可以视作小机场在大机场的辐射范围内。因此，可以使用大机场的标签作为小机场的标签。因此，使用节点度数较大的邻接节点作为采样样本，在机场数据集中更加合理。

对于另一个 wiki 数据集，通过比较可以得知：优先选择节点度数小的方法效果比原本的随机采样要好，并且这样的优势在三种映射方式上都有所体现，而使用节点度数大的方法得到的效果比原方法的效果更差。使用我们前面提到的猜想同样可以对结果进行解释：wiki 上面的词条相互引用连接形成图，要想比较好得反映出词条的属性，使用的是和其关联的同一领域的

表 2: 基于度数的实验结果

数据集	随机游走	大度数优先	小度数优先法 1	小度数优先法 2	小度数优先法 3
wiki	0.6879	0.6343	0.7076	0.7042	0.6925
usa-airport	0.5370	0.5668	0.5240	0.5339	0.5297
europe-airport	0.4138	0.4338	0.4213	0.4142	0.4225

表 3: 基于路径对 wiki 数据集不同步长的实验结果

方法	10	20	30
随机游走	0.6879	0.6842	0.6800
对路径进行约束	0.6744	0.6848	0.6853

相关词条，而不是引用量高的词条。这样的词条往往没有被很多的词条所引用，只是领域内的相关词条引用，因此这样的词条度数一般较少，但却能更好的反映词条的属性。

3.3 基于路径对 DeepWalk 的讨论

从随机游走的方法出发，我们还可以对随机游走的路径长度进行分析。考虑随机游走的原理，作者使用随机游走的原因在于这样的行走本质上表现为一种采样，采样的概率密度函数分布随着和当前节点的距离拉开而逐渐减小。因此，实现了更多使用距离节点比较近的邻居节点表示当前节点的信息，而距离当前节点比较远的邻居节点被采样的概率减小，从而避免当前节点的属性存在过多偏差。

从这一角度出发，我们设计出一种新的采样方式，可以人为增强这种约束：随着采样路径的增长，我们适当增加往回行走的采样策略。和上面的讨论类似，我们建立了相应的概率映射函数，使用公式表示为：

$$p(v_i) = \begin{cases} \frac{l}{l+|u|-1}, & v_i \text{ is previous node} \\ \frac{1}{l+|u|-1}, & v_i \text{ is not previous node} \end{cases} \tag{4}$$

3.4 基于路径讨论的结果分析

我们猜想当随机游走行走的路径较长时，会出现类似于过拟合的现象，此时采样的节点可能不能很好地反映编码节点的信息。此时，使用上述的方法对路径进行限制会有比较好效果。

和前面的工作一样，我们使用不同的随机游走步长，对上面的数据集验证猜想。各个数据集的效果如表 3、4、5 所示：通过表格可以看出有两个数据集在 random walk 后达到过拟合后，使用对路径约束的方法仍能保持比较准确率继续上升的趋势。但对于 europe-airport 数据集，使用路径约束的方法并不能很好地反映方法的优势，效果甚至会有所下降。我们分析其中的原因有：有可能 europe-airport 数据集过小，不能很好地反映方法的优势；也有可能是我们的方法不具有普遍性，可能存在缺陷。

表 4: 基于路径对 usa-airport 数据集不同步长的实验结果

方法	10	20	30	40
随机游走	0.5370	0.5521	0.5605	0.5563
对路径进行约束	0.5373	0.5492	0.5554	0.5624

表 5: 基于路径对 europe-airport 数据集不同步长的实验结果

方法	10	20	30
随机游走	0.4138	0.4525	0.4138
对路径进行约束	0.4075	0.4063	0.4125

4 相关工作陈述

图嵌入算法是一种将图中节点、边及其特征转换为较低维度的向量空间，同时最大限度地保留图结构和信息等属性的方法。约二十年前，人们提出了图嵌入算法，算法思想是根据实际问题构造一个 D 维空间中的图，然后将图的节点嵌入到 d（d«D）维向量空间中，嵌入指的是在向量空间中保持邻接的节点彼此靠近。在过去的十年里，在图嵌入领域已经有了大量的研究，重点是设计新的嵌入算法。发展到现在，大体上可以将这些嵌入方法分为三大类：（1）基于因子分解的方法，（2）基于随机游走的方法，以及（3）基于深度学习的方法 [1]。

4.1 基于因子分解的方法

- Locally Linear Embedding，局部线性嵌入，局部线性假设每个节点都是相邻节点的线性组合。通过最小化目标函数 $\phi(Y) = \sum_{i=1}^{N} |y_i - \sum_{j=1}^{k} w_{ij} y_j|^2$ 得到嵌入矩阵 Y。
- Laplacian Eigenmaps，拉普拉斯特征映射，在权重 w_{ij} 较高时，如果在嵌入后被分割过远，则给予更高的惩罚。通过最小化目标函数 $\phi(Y) = \frac{1}{2} \sum_{ij} |y_i - y_j|^2 w_{ij}$ 进行求解。

4.2 基于随机游走的方法

- DeepWalk 受到 word2vec 的启发，选择某一特定点作为起始点，通过随机游走得到一系列点的序列，用 word2vec 来学习该序列，从而得到表示该点的向量。
- node2vec 和 DeepWalk 类似，但在随机游走这一步骤中使用有偏随机游走，即通过参数控制，使得随机游走反映出类似于 DFS 或 BFS 的采样特性，从而提高采样的效果。

4.3 基于深度学习的方法

- SDNE 使用深度自动编码器来保持一阶和二阶网络临近度，该方法通过自动编码器来寻找一个重构邻域节点的嵌入，并基于拉普拉斯特征映射来对嵌入结果进行评判反馈。
- DNGR 使用随机游走模型来生成概率共现矩阵，将该矩阵转化为 PPMI 矩阵，输入到叠加去噪自动编码器中得到嵌入结果。

5 总结与展望

参考文献

[1] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.