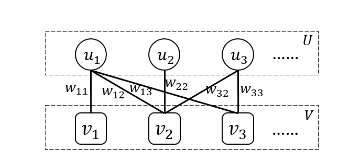
《BiNE: Bipartite Network Embedding》

《二部网络嵌入》

**摘要：**

1. **本文的背景：**在图神经网络早期研究中，大部分有效的方法（如DeepWalk、LINE等）都集中在同质网络的问题处理上，而对于像二部图这种特殊的异质网络，直接将这些方法应用在其上取得的效果并不理想。因此，本文的目的是找出一种框架将图嵌入学习更好地应用在二部图（bipartite network）上。
2. **本文的贡献：**提出了一种新的图神经网络框架——BiNE （Bipartite Network Embedding）来学习二部网络的节点嵌入表示。
3. **主要创新点：**设计了一种有偏的随机游走（random walk）来生成节点序列；并提出了一种同时考虑节点显式关系（explicit relations）和隐式关系（implicit relations）的联合优化框架来更好地学习节点的嵌入表示。
4. **实验结果：**在链接预测（link prediction）和推荐（recommendation）的应用实验中，BiNE相比当时的baseline模型取得了更好的效果，这说明通过BiNE训练得到的节点嵌入表示可有效用于各种实际的应用任务。此外，其他一系列的定量和定性的分析也都证明了BiNE的有效性。

**二部图的定义：**



一个二部网络的结构如上图所示。对于一个二部图，其中，分别表示两种类型的节点的集合，而代表二部图内连接两类节点的边的集合。并且，每一条边都有一个非负的权重，用于表示两节点与关系的强弱。

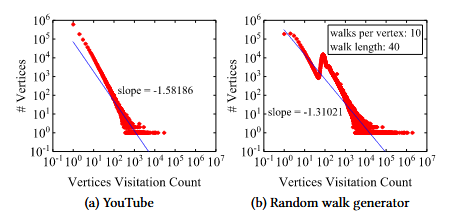
而二部图嵌入表示学习的目标是，给定一个二部图及其边的权重，将网络中所有的节点（）映射到一个低维的向量空间中，其中表示每个节点嵌入向量的维度。

**传统图嵌入方法的缺陷：**（以DeepWalk为例）

二部图（bipartite graph）是一种现实中常见的数据结构，它被广泛应用于各种场景，如推荐系统、搜索引擎、问答系统等。而常见的图嵌入方法，如DeepWalk，通常分为两步：1）通过随机游走得到顶点序列；2）使用词嵌入方法进行计算得到嵌入表示。

然而，将这些传统的方法应用在二部图上（一种特殊的异构图）它们的效果会是次优的，主要原因有以下两点：

1. **未考虑节点的类型信息**，虽然边只存在于不同类型的节点之间，但是同类型的节点之间也可能存在着某种重要的隐含关系。
2. **通过随机游走生成的序列并不能很好的保持二部图的特征**。因为二部图的分布通常遵循幂律分布，但是像DeepWalk等的随机游走的设计存在不足，难以反映二部图的这种分布情况。具体情况如下图所示：



可见，由于DeepWalk随机游走生成器的设计不当，生成的语料库（序列）没有显示出二部图所需的幂律分布。

而本文提出的BiNE有效解决了传统图嵌入方法上述的两点不足：

1. 对于网络中的隐式关系和显示关系，BiNE分别设计了对应的目标函数，使得被忽略的隐式关系也能被显式的建模出来，并且实验证明这样做是有助于提高模型的表现成绩的。
2. 虽然BiNE同样采用了随机游走的策略，但与DeepWalk不同，它是一种有偏的随机游走策略。一是根据节点的重要性，来决定有多少序列以该节点为起点。二是生成的序列长度并不是完全相同的，而是设置了一个概率来控制游走的停止。

**显式关系建模：**

与中的一阶相似性建模类似，通过考虑两个连接顶点之间的局部相似性来建模显式关系。所以，顶点与之间的联合概率被定义为：

而度量两个节点的嵌入表示在映射空间内的相近性则参考了取向量内积的思想，定义如下：

其中，，分别代表节点与的嵌入向量。

最后，我们再通过散度来衡量上述两个分布的差异，以此定义显式关系的目标优化函数，即最小化下式：

**直观理解：**对于两个相连紧密的节点，如果学习到的两节点的表示在低维的向量空间中也是彼此靠近的，则保持了所谓的局部相似性。

**隐式关系建模：**

隐式关系定义：

对于二部图，两个相同类型的节点，尽管它们之间不存在直接相连的边，但如果存在一条从到的路径，我们就认为这两个节点之间存在着某种隐式关系。**也就是说，本文所指的节点之间的隐式关系，是相对于两个同类型的节点来说的**。

构建节点序列的语料库：

为此，为了学习到二部图同类型节点上的隐式关系，我们需要把一个二部图网络拆分为两个同质网络，而定义两相同类型的节点的隐式关系（二阶相似度）的公式如下：

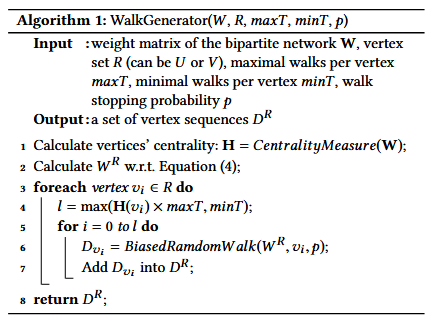
于是，我们就可以得到两个同质网络的权重矩阵和：

其中，，，即它们的维度大小分别为和。而代表原二部图的权重矩阵。

接着，我们就需要在这两个新的同质图上进行随机游走，以获取对应节点类型的序列来学习相关的隐式关系。但是，需要注意的是，由于的随机游走策略并不是最优的，因此本文重新设计了一种偏置的和自适应的随机游走发生器（biased and self-adaptive random walk generator）。具体方式如下：

1. 对于每一个节点，它的中心性（centrality）越强，则从它开始的随机游走序列越多，即随机游走的迭代次数越多。
2. 定义一个概率值，使得随机游走在每一步都有可能立即停止，从而得到一系列长度不同的随机节点序列。

相关算法伪代码如下：



建模隐式关系：

在前文生成的两个语料库上分别使用模型来学习节点的表示。其目标函数如下：

其中，为语料库中的节点序列，为在该序列中节点的上下文节点的集合，。可以看到，在计算时，公式采用了，会非常花时间。所以，我们需要通过使用负采样的方法来进一步优化目标函数，从而得到最终的优化公式：

其中，代表节点的负采样节点集合。

而为：

其中，代表激活函数，代表节点的嵌入向量，代表节点的上下文（二阶）嵌入向量。

同理，节点类型为的目标函数也是如此：

其中，代表节点的负采样节点集合。

而为：

**联合优化：**

根据显式关系和隐式关系建模得到的目标函数，我们可以得到最终的目标函数公式为：

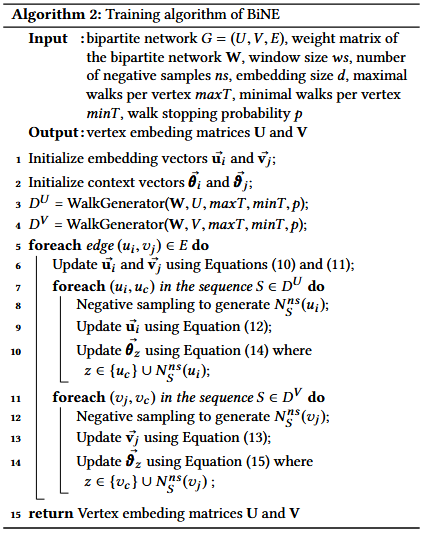
使用Stochastic Gradient Ascent algorithm (SGA)来优化这一函数。但是，由于优化这个函数的不同部分需要使用不同的训练实例，因此我们必须分步进行优化操作。

首先，是显式关系的优化，即中的部分，更新节点表示的公式如下所示：

接着，是隐式关系的优化，公式如下：

其中，用于判断节点是否为节点的上下文节点，同理。最后，上下文向量的更新公式如下：

**BiNE模型的算法伪代码如下：**



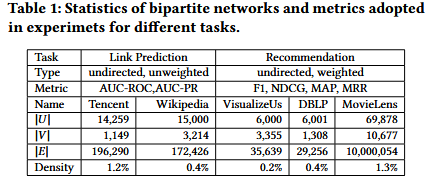
**实验及结果：**

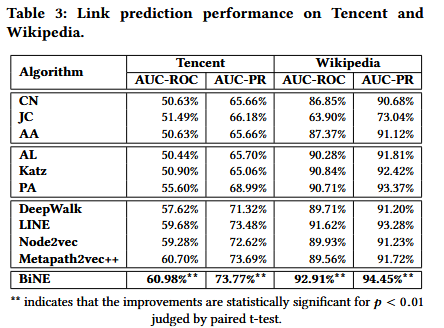
本文进行了两类应用实验：链接预测和推荐，并尝试探究以下四个问题：

1. 和方法与其他新型方法相比，的实际表现到底如何？
2. 明确隐式关系建模是否真的有助于学习节点的表示？
3. 本文设计的有偏随机游走策略是否真的有效？
4. 超参数的设置如何影响模型的性能？（参数敏感度分析）

RQ1：

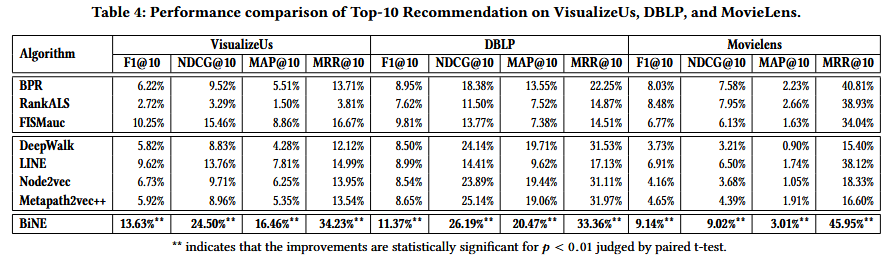
对于链接预测任务，本文使用了来自Wikepedia和Tencent的二部图数据集，分别表示了网页—作者和QQlive上用户—电影的关系信息。数据集情况如下：



而与方法的对比结果如下：

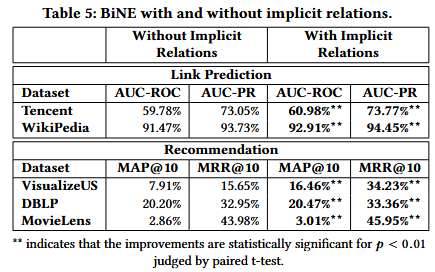
可见在两类数据集的链接预测任务上，BiNE模型都取得了最佳地效果。

而对于推荐任务，实验中使用了DBLP，Movielens和VisualizeUs这三个常见的数据集，实验结果如下：



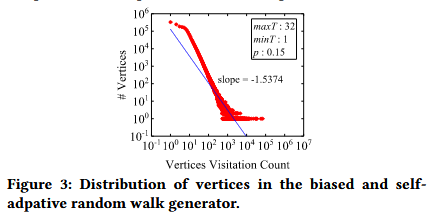
可见在三类数据集的推荐任务上，BiNE模型都取得了最佳地效果。

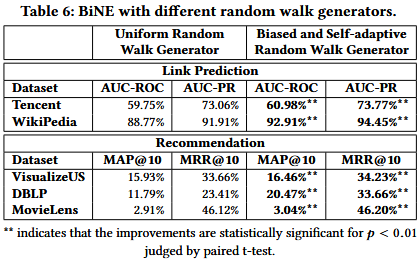
RQ2：



可以看到，隐式关系建模确实能帮助模型提高节点嵌入表示的性能。

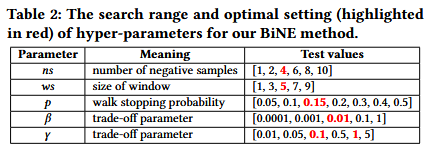
RQ3：

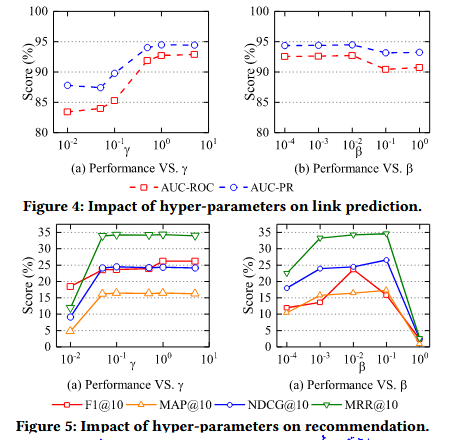




可以看到，本文设计的有偏随机游走策略的确能够提高模型的嵌入学习能力。

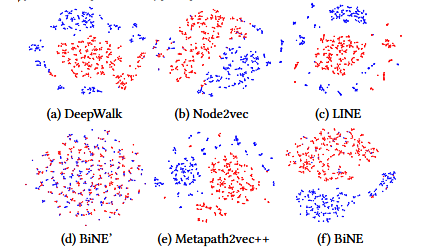
RQ4：





不同超参数的最佳取值以及超参数和取值对模型性能的影响。

不同方法得到的嵌入向量可视化结果：



**总结：**

1. 相比于传统的同质图方法，能够有效地利用二部图（异质图）的特点，生成不同类型节点的embedding，且效果更好；
2. 异质图的SOTA方法认为显式关系和隐式关系的作用相等，并且忽略了图上节点之间的权重信息；而综合考虑了显式关系和隐式关系的作用，并且很好地利用了权重信息。
3. 本文提出的有偏随机游走策略，每个节点采样的游走序列数与节点的重要性相关，并且序列长度不固定，更符合图的原始分布。

**对本文的感悟：**

在我看来，本文最精妙地创新点在于将二部图转化为两个同构图的大胆创新，使得在异构图上的随机游走能够间接等价于在同构图上的随机游走。而这种对于隐式关系的定义无疑是创新且大胆的，这也是我们在研究中值得学习和借鉴的地方。