**Network Embedding as Matrix Factorization: Unifying  
DeepWalk, LINE, PTE, and node2vec**

# ABSTRACT

自从word2vec [28,29]发明以来，Skip-gram模型已经显着地推进了网络嵌入的研究，如最近DeepWalk，LINE，PTE和node2vec方法的出现。在这项工作中，我们表明，所有上述模型的负面抽样可以统一到矩阵分解框架的封闭形式。我们的分析和证明表明：（1）DeepWalk [31]经验地产生网络归一化拉普拉斯矩阵的低秩变换; （2）在理论上，LINE [37]是当顶点的上下文大小设置为1时DeepWalk的一个特例; （3）作为LINE的延伸，PTE [36]可以看作是多网络拉普拉斯算子的联合分解; （4）node2vec [16]分解与二阶随机游走的平稳分布和转移概率张量有关的矩阵。我们进一步提供基于skip-gram的网络嵌入算法与图拉普拉斯算子理论之间的理论联系。最后，我们介绍了NetMF方法1及其计算网络嵌入的近似算法。我们的方法在传统的网络挖掘任务上比DeepWalk和LINE有显着的改进。这为基于跳跃的网络嵌入方法奠定了理论基础，从而更好地理解潜在的网络表示学习。

# 1 INTRODUCTION

传统的网络挖掘和学习范式通常从对结构性质的明确探索开始[13,32]。但是，**介数中心性，三角形数和模块性**等许多这样的性质需要广泛的领域知识和昂贵的计算手工。针对这些问题，以及最近出现的表示学习[2]所提供的机会，**对网络的学习潜在表示（又名网络嵌入）**进行了广泛的研究，**以便自动发现和映射网络的结构特性到一个潜在的空间**。形式上，网络嵌入问题通常形式化如下：给定一个以V为节点集的无向加权图G =（V，E，A），E为边集，A为邻接矩阵，学习函数V→Rd，**将每个顶点映射到一个d维（d«| V |）向量，该向量捕捉其结构特性**。输出表示（The output representations）可以用作各种网络科学任务（如标签分类和社区检测）的挖掘和学习算法的输入。

解决这个问题的尝试可以追溯到谱图理论[11]和社会维度学习[38]。其最近的进展主要受最初提出的用于词嵌入的跳跃模型的影响[28,29]，**其输入是由自然语言的句子组成的文本语料库，并且输出是语料库中的每个词的潜在向量表示。**值得注意的是，受这种设置的启发，DeepWalk [31]将网络随机游走的顶点路径作为句子，并利用跳跃图学习潜在顶点表示来开创网络嵌入。随着DeepWalk的出现，许多网络嵌入模型已经被开发出来，例如LINE [37]，PTE [36]和node2vec [16]。

迄今为止，上述模型已经得到了有效的证明。但是，他们背后的理论机制却不甚明了。我们注意到，用于词语嵌入的带负样本的skip-gram模型已经被证明是**某个词语上下文矩阵的隐含因式分解**[24]，最近有一些理论上从几何角度解释词语嵌入模型[1 ，18]。但是，不清楚单词背景矩阵与网络结构之间的关系。此外，还有一个早期尝试从理论上分析DeepWalk的行为[47]。但是，他们的主要理论结果与原有的DeepWalk论文的设置并不完全一致。另外，尽管DeepWalk，LINE，PTE和node2vec之间存在着非常类似的关系，但是对于它们的底层连接缺乏更深入的了解。

**贡献**

在这项工作中，我们提供了几个skip-gram powered的网络嵌入方法的理论结果。更具体地说，我们首先证明我们提到的模型DeepWalk，LINE，PTE和node2vec**理论上是在执行隐式矩阵分解**。

我们推导出每个模型的矩阵的闭合形式（摘要见表1）。例如，DeepWalk（随机游走图+ skip-gram）实质上是将随机矩阵分解，随机矩阵随着漫游的长度变为无穷大，随机矩阵以概率收敛到闭式矩阵。

其次，从矩阵的封闭形式来看，我们发现有趣的是，LINE可以看作是DeepWalk的一个特例，当上下文的窗口大小T被设置为1时。此外，我们证明PTE作为LINE的延伸，实际上是多网络联合矩阵的隐式因式分解。

第三，我们发现了DeepWalk的隐式矩阵和图拉普拉斯算子之间的理论联系。在此基础上，提出了一种新的算法NetMF逼近DeepWalk隐式矩阵的闭合形式。通过使用SVD对这个矩阵进行明确分解，我们在四个网络（用于DeepWalk和node2vec方法）中的广泛实验证明了NetMF在DeepWalk和LINE上的出色性能（相对提高了50％）。

# 2 THEORETICAL ANALYSIS AND PROOFS

在这一节中，我们将提供详细的理论分析和四种流行网络嵌入方法的证明：LINE，PTE，DeepWalk和node2vec。

## 2.1 LINE and PTE