**图的表示：方法和应用**

目录

[摘要 1](#_Toc66524683)

[1引言 1](#_Toc66524684)

[1.1符号和基本假设 3](#_Toc66524685)

[2嵌入节点 4](#_Toc66524686)

[2.1方法概述:从编码器-解码器的角度 4](#_Toc66524687)

[2.1.1优化注意事项及实施细节 6](#_Toc66524688)

[2.2浅层模型 6](#_Toc66524689)

[2.2.1基于因子分解的方法 7](#_Toc66524690)

[2.2.2随机游走方法 9](#_Toc66524691)

[2.3广义的编码器架构 12](#_Toc66524692)

[2.3.1邻域自编码器方法 13](#_Toc66524693)

[2.3.2邻域聚集和卷积编码器 15](#_Toc66524694)

[2.4实时任务监控 17](#_Toc66524695)

[2.5对多模态图的扩展 17](#_Toc66524696)

[2.5.1处理不同节点和边缘类型 17](#_Toc66524697)

[2.5.2跨层绑定节点嵌入 19](#_Toc66524698)

[2.6嵌入结构角色 20](#_Toc66524699)

[2.7节点嵌入的应用 20](#_Toc66524700)

[3 嵌入子图 22](#_Toc66524701)

[3.1节点嵌入和卷积方法集 22](#_Toc66524702)

[3.1.1 基于总和的方法 22](#_Toc66524703)

[3.1.2图形粗化方法 23](#_Toc66524704)

[3.1.3进一步变化 24](#_Toc66524705)

[3.2图神经网络 24](#_Toc66524706)

[3.3子图嵌入的应用 25](#_Toc66524707)

[4 结论与未来方向 26](#_Toc66524708)

[4.1对未来进步的挑战 26](#_Toc66524709)

[4.2重要开放问题 27](#_Toc66524710)

# 摘要

图上的机器学习是一项重要且普遍存在的任务，其应用范围从药物设计到社交网络中的友情推荐。这个领域的主要挑战是找到一种表示或编码图形结构的方法，以便机器学习模型可以很容易地利用它。传统上，机器学习方法依赖于用户定义的启发式来提取一个图的特征编码结构信息(例如，度统计或核函数)。然而，近年来，使用基于深度学习和非线性降维的技术，自动学习将图结构编码为低维嵌入的方法激增。在此，我们提供了一个概念回顾的关键进展在这个领域的表示图，包括基于矩阵分解的方法，随机行走的算法，和图神经网络。我们回顾了嵌入单个节点的方法以及嵌入整个(子)图的方法。在此过程中，我们开发了一个统一的框架来描述这些最近的方法，并强调了未来工作的一些重要应用和方向。

# 1引言

图是一种普遍存在的数据结构，广泛应用于计算机科学和相关领域。社会网络、分子图结构、生物蛋白质-蛋白质网络、推荐系统——所有这些领域以及更多的领域都可以很容易地建模为图，图捕捉单个单元(即节点)之间的交互作用(即边)。由于图的普遍性，它是无数系统的支柱，允许[2]高效地存储和访问关于交互实体的关系知识。

然而，图形并不是有用的结构化知识仓库:它们还在现代机器学习中起着重要作用。许多机器学习应用程序试图使用图形结构的数据作为特征信息进行预测或发现新的模式。例如,人们希望将一种蛋白质的作用在一个生物交互图,预测一个人的角色在一个协作网络,新朋友推荐给用户的社交网络,或预测新治疗的应用现有的药物分子,其结构可以表示成一个图形。

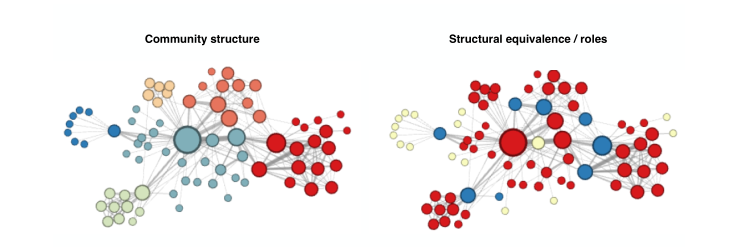


图1:来自《悲惨世界》小说的人物-人物互动图的两种不同观点，如果对应的人物互动，则两个节点连接。左图的上色强调了图中节点在全局位置上的差异:如果节点属于同一个社区，在全局层次上，它们的颜色是相同的。相比之下，右图的颜色表示节点之间的结构等价性，或者表示两个节点在其本地社区中扮演相似的角色(如“桥接节点”用蓝色表示)。这两个图形的颜色是使用node2vec节点嵌入方法[28]的不同设置生成的，见第2节。经许可转载自[28]

图上机器学习的核心问题是找到一种方法，将图形结构的信息合并到机器学习模型中。例如，在社交网络中的连接预测中，可能需要对节点之间的成对属性进行编码，例如关系强度或共同朋友的数量。或节点分类的情况下,可能希望包含信息的全球地位图（全局）中的一个节点或节点的本地图（局部图）附近的结构(图1)。挑战——从一个机器学习的观点是,没有简单的方法来编码这个高维非欧几里得的信息图结构成一个特征向量。

为了从图中提取结构信息，传统的机器方法通常依赖于摘要图统计信息(例如，度数或聚类系数)[6]、核函数[58]或精心设计的特征来测量局部邻域结构[40]。然而，这些方法是有限的，因为这些手工工程的特性是不可改变的。，它们在学习过程中无法适应——设计这些特性可能是一个耗时且昂贵的过程。

最近，出现了一股寻求学习编码图形结构信息的表示方法的热潮。这表示学习方法背后的理念是学习一个嵌入节点的映射,或者整个(子)图,分在一个低维向量空间Rd。我们的目标是优化这种映射,这样在嵌入空间中几何关系反映原始图的结构。在优化了嵌入空间后，学习到的嵌入可以作为下游机器学习任务的特征输入。表示学习方法与以往工作的主要区别在于它们如何处理表示图结构的问题。以前的工作将这个问题作为预处理步骤，使用手工设计的统计数据来提取结构信息。相反，表示学习方法把这个问题当作机器学习任务本身，使用数据驱动的方法来学习嵌入图结构的编码。

在这里，我们概述了图表示学习的最新进展，回顾了表示节点和整个子图的技术。我们的调查试图将多个不同的研究领域融合在一起，最近引起了不同子领域和场所的极大关注在最近几年—例如，节点嵌入方法，它们是数据挖掘社区中流行的研究对象，并且图卷积网络已经在主要的机器学习场所引起了相当大的关注。

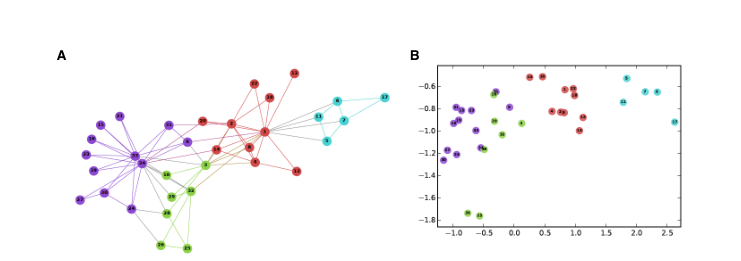


图2:A, Zachary Karate Club社交网络的图结构，如果对应的个体是朋友，则节点连接。节点根据网络中存在的不同社区着色。使用DeepWalk方法(第2.2.2节)[47]从此图生成的节点嵌入的二维可视化。嵌入空间中节点之间的距离反映了原始图中的相似性，并根据不同颜色编码的社区对嵌入节点进行空间聚类。经[47,49]许可转载。

我们关注的是最近在机器学习和数据挖掘领域获得极大关注的一些方法，特别是那些可扩展到海量图(例如，数以百万计的节点)的方法，这些方法受到了深度学习的启发。当然,还有其他行密切相关和相关工作,我们不详细审查——包括潜在的社交网络的空间模型[33],嵌入的方法统计关系学习[43],流形学习算法[38],以及几何深度学习[7]——所有这些涉及表示学习与图结构的数据。我们向读者推荐[33]、[43]、[38]和[7]以获得这些领域的全面概述。

## 1.1符号和基本假设

我们假设我们的表示法学习算法的主要输入是一个无向图和相关的二进制邻接矩阵A [2]. 我们还假设这些方法可以利用节点属性的实值矩阵(例如。表示与节点关联的文本或元数据)。目标是利用A和X中包含的信息将每个节点或子图映射到向量，其中。我们回顾的大多数方法将以一种无监督的方式优化这个映射，只利用矩阵的信息，而不了解特定的下游机器学习任务。然而，我们也将讨论一些监督表示学习的方法，其中模型利用分类或回归标签来优化嵌入。这些分类标签可能与单个节点或整个子图相关联，并且是下游机器学习任务的预测目标(例如，它们可能根据图的表示来标记蛋白质的角色，或分子的治疗特性)。

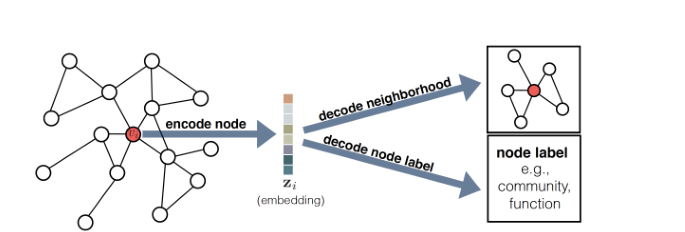


图3:编码器-解码器方法的概述。首先，编码器根据节点在图中的位置、其局部邻域结构和/或其属性，将节点映射到一个嵌入的低维向量。接下来，解码器从低维嵌入中提取用户指定的信息;这可能是关于的局部图邻域的信息(例如，它的邻域的标识)，或者与相关的分类标签(例如，一个分类标签，一个社区标签)。通过联合优化编码器和解码器，系统学会了将图形结构信息压缩到低维嵌入空间中。

# 2嵌入节点

我们从讨论节点嵌入的方法开始，目标是将节点编码为低维向量，总结它们的图位置和它们的局部图邻域的结构。这些低维嵌入可以看作是节点编码或投射到一个潜在空间，在这个潜在空间中的几何关系对应于原始图[33]中的交互作用(例如，边)。图2展示了一个嵌入著名的Zachary Karate Club社交网络[47]的示例，其中二维节点嵌入捕获了社交网络中隐含的社区结构。

## 2.1方法概述:从编码器-解码器的角度

近年来，对节点嵌入的研究激增，导致了符号、动机和概念模型的复杂多样性。因此，在讨论各种技术之前，我们首先建立了一个统一的编解码器框架，明确地构建了这种方法的多样性，并将各种方法置于同等的标注和概念基础上。

在这个框架中,我们组织周围的各种方法两个关键的映射函数:一个编码器,每个节点映射到一个低维向量,或嵌入和adecoder解码结构的图形信息学习嵌入(图3)。encoder-decoder背后的直觉想法如下:如果我们能学会解码高维图像信息(例如全球图中节点的位置或当地街区图编码的低维嵌入的结构,然后,原则上,这些嵌入的下游机器学习任务应包含所有必要的信息。

形式上，编码器是一个函数，



将节点映射到向量嵌入zi∈Rd(其中zicor响应节点vi∈V的嵌入)。解码器是一个函数，它接受一组节点嵌入，并从这些嵌入中解码用户指定的图形统计信息。例如，解码器可以根据节点的嵌入情况预测节点间边的存在[1,36]，或者可以预测某个节点在图中所属的社区[29,35](图3)。原则上，解码器可能有很多;然而，绝大多数作品使用基本的两两解码器，



该方法将节点嵌入对映射到一个实值节点相似性度量，该度量量化了原始图中两个节点的相似性。

当我们对一对嵌入(zi,zj)应用成对解码器时，我们重构了原始图中viand vjin之间的相似性，目标是优化编码器和解码器的映射，使重构中的误差或损失最小化，从而:



其中，sGis是在图g上定义的用户定义的、基于图的节点之间的相似性度量。换句话说，我们希望优化编码器-解码器模型，以便从低维节点嵌入ziand zj中对原始图sG(vi, vj)中的节点相似性进行解码。例如，可以设置sG(vi, vj)， Ai,jand定义节点的相似性为1，如果它们是相邻的，则为0，否则为[1]，或者可以根据在图G上的固定长度随机游走中viand vj共出现的概率来定义sg2[28,47]。在实践中，大多数方法通过最小化一组训练节点对D的经验损失L来实现重构目标(公式3):



式中:R×R→R是用户指定的损失函数，它测量译码(即估计)相似度值DEC(zi,zj)与真实值sG(vi, vj)的差异。一旦我们对编码器-解码器系统进行了优化，我们就可以使用经过训练的编码器来生成节点的嵌入，然后将其用作下游机器学习任务的特征输入。例如，可以将学习到的嵌入式输入logistic回归分类器来预测某个节点属于[47]的社区，或者可以使用嵌入之间的距离来推荐社交网络中的友谊链接[3,28](第2.7节讨论了进一步的应用)。

采用这种编码器-解码器的观点，我们按照以下四种方法组件来组织我们对各种节点嵌入方法的讨论:

1. 图g上定义了一个**两两相似度函数**sG: V×V→R+，该函数度量了图g中节点之间的相似度。

2.一个**编码器函数**，ENC，生成节点嵌入。这个函数包含许多在训练阶段优化的可训练参数。

3.一个**解码器函数**，DEC，它从生成的嵌入重新构造成对的相似值。这个函数通常不包含可训练的参数。

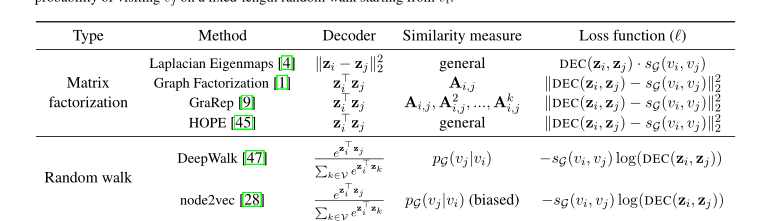
4. 一个**损失函数'**，它决定了为了训练模型如何评估两两重建的质量，即如何将DEC(zi,zj)与真实的sG(vi, vj)值进行比较。

正如我们将要展示的，各种节点嵌入方法之间的主要方法区别在于它们如何定义这四个组件。

### 2.1.1优化注意事项及实施细节

我们回顾所有的方法包括编码器的参数优化算法,ΘENC,通过最小化损失类似于方程(4).3In大多数情况下,随机梯度下降法用于优化,虽然有些算法允许封闭方案通过矩阵分解(例如,[9])。但是，请注意，我们在这里将不关注优化算法，而是强调存在于不同嵌入方法之间的高层次差异，而与优化方法的具体细节无关。

**表1:**总结了一些著名的浅埋算法。注意，基于随机游走的方法的解码器和相似函数是不对称的，相似函数pG(vj|vi)对应从vi开始访问vjon的定长随机游走的概率。



## 2.2浅层模型

大多数节点嵌入算法依赖于我们所说的浅嵌入。对于这些浅层嵌入方法，encoder函数——它将节点映射到向量嵌入——只是一个简单的“嵌入查找”:



在Z∈Rd×V | |是所有节点矩阵包含嵌入向量和vi∈新Z的列向量表示一个炎热的指标对应节点vi。可训练的参数集浅嵌入方法只是ΘENC = {Z},即嵌入矩阵Z是直接优化。

这些方法很大程度上是受经典矩阵分解技术的启发，用于降维[4]和多维尺度[37]。实际上，许多这些方法最初是作为因数分解算法，我们在这里的编码器-解码器框架内重新解释它们。表1总结了在编码器-解码器框架中一些众所周知的浅层嵌入方法。表1强调了如何根据(i)解码器函数，(ii)基于图的相似性度量，(iii)它们的损失函数简明地描述这些方法。下面两节将更详细地描述这些方法，区分基于矩阵因子分解的方法(第2.2.1节)和基于随机漫步的更近期的方法(第2.2.2节)。

### 2.2.1基于因子分解的方法

早期学习节点表示的方法主要集中在矩阵分解方法上，直接受到了维数降维的经典技术的启发[4,37]。拉普拉斯算子eigenmaps。最早和最著名的实例之一是Laplacian eigenmaps (LE)技术[4]，我们可以在编码器-解码器框架内将其视为一种浅层嵌入方法，其中解码器定义为

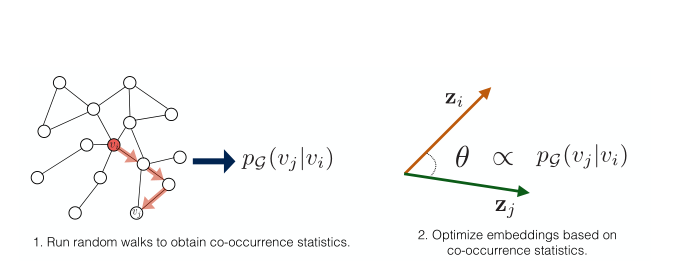


其中损失函数根据图中节点的相似度对节点进行权重:



**内积的方法**。在Laplacian特征映射技术之后，有大量基于成对、内积解码器的嵌入方法:





**图4:**基于随机游走的方法示例大量固定长度的随机漫步从每个节点开始,vi。然后嵌入向量优化的点积,或角,两个嵌入,ziand zj,(大约)访问的概率成正比vjon固定长度的随机漫步从vi。

在这里，两个节点之间关系的强度与它们的内嵌的点积成正比。图因子分解(GF)算法m4[1]， GraRep[9]，和HOPE[45]都属于这个类。特别地，这三种方法都使用了内积解码器，即平均平方误差(MSE)损失，



它们的主要区别在于所使用的节点相似性度量，即它们如何定义sG(vi, vj)。图元分解算法直接基于邻接矩阵(即sG(vi, vj)， Ai,j)[1]定义节点相似性;GraRep考虑邻接矩阵的各种幂次(如sG(vi, vj)， A2 i,j)，以获取高阶节点相似度[9];希望算法支持一般的相似度度量(如基于Jaccard邻域重叠)[45]。这些不同的相似函数在建模“一阶相似度”和建模“高阶相似度”之间进行权衡，其中sg直接度量节点之间的连接(即sG(vi, vj)， Ai,j[1])，其中sg对应更一般的邻域重叠概念(如sG(vi, vj) = A2 i,j[9])。

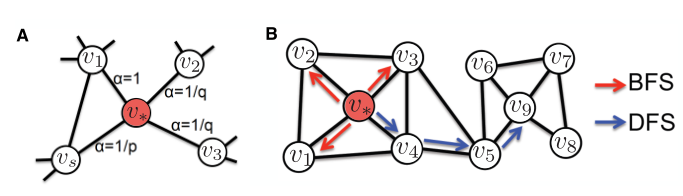
在本节中，我们将这些方法称为矩阵因子分解方法，因为，对所有节点进行平均，它们(大致)优化了以下形式的损失函数:



其中S为包含成对相似测度(即Si,j, sG(vi, vj))的矩阵，Z为节点嵌入矩阵。直观地说，这些方法的目的仅仅是学习每个节点的嵌入，以便学习到的嵌入向量之间的内积近似于节点相似性的一些确定性度量。

### 2.2.2随机游走方法

许多最近成功的方法也属于浅层嵌入方法的类别，学习基于随机游走统计的节点嵌入。他们最主要的创新就是优化节点嵌入,节点有相同的嵌入的如果他们倾向于在短的随机漫步共现图(图4)。因此,而不是使用一个确定性测量节点的相似性,如2.2.1节的方法,这些随机游走方法采用一种灵活的、随机测量节点的相似性,导致性能优越的设置[27]。



**图5:** A，说明node2vec如何使用p和q参数使随机游走产生偏差。假设游走只是从vsto v转换成∗，边缘标签，近似，与下一个时间步走取那条边的概率成正比。基于宽度优先搜索(BFS)和深度优先搜索(DFS)的随机遍历的区别。类似bfs的随机漫步主要局限于探索节点的即时(即单跳)邻域，通常更有效地捕捉结构角色。类似于dfs的walk在远离节点的地方探索，更有效地捕获社区结构。改编自[28]。

**DeepWalk node2ve**c。像上面描述的矩阵分解方法一样，DeepWalk和node2vec依赖于浅层嵌入，并使用基于内积的解码器。然而，这些方法并不是试图解码一个确定性的节点相似性度量，而是优化嵌入来编码随机游动的统计信息。这些方法背后的基本思想是学习嵌入式，以便(大致):



其中pG,T(vj|vi)为从vi开始访问vjon a length-T随机游走的概率，T通常定义在T∈{2，…，10}的范围内。注意，与2.2.1节中的相似性度量不同，pG,T(vj|vi)是随机的和不对称的。

更正式地说，这些方法试图最小化以下交叉熵损失:



其中，训练集D是通过从每个节点开始的抽样随机漫步生成的(也就是说，每个节点viare从分布(vi, vj)的~ pG,T(vj|vj)中采样N对)。然而，天真地评估公式(11)中的损失是非常昂贵的，特别是O(|D||V|)，因为计算公式(10)的分母有时间复杂度O(|V|)。因此，DeepWalk和node2vec使用不同的优化和近似来计算公式(11)中的损失。DeepWalk采用“分级softmax”技术来计算规格化因子，使用二叉树结构来加速计算[47]。相比之下，node2vec使用“负采样”逼近方程(11):node2vec使用一组随机的“负采样”[28]逼近归一化因子，而不是对整个顶点集进行归一化。

除了这些算法差异之外，node2vec和DeepWalk之间的关键区别在于，node2vec允许对随机漫步进行灵活定义，而DeepWalk在图上使用简单的无偏随机漫步。特别地，node2vec引入了两个随机游动超参数，p和q，它们对随机游动产生了偏差(图5.A)。超参数p控制重新访问节点的可能性，而q控制重新访问节点的单跳邻域的可能性。通过引入这些超参数，node2vec能够在更类似于宽度优先或深度优先搜索的遍历之间平滑地进行插值(图5.B)。Grover等人发现，调整这些参数允许模型在强调社区结构的学习嵌入和强调本地结构角色的嵌入之间进行权衡(也参见图1)。

**大规模信息网络嵌入(LINE)**。另一种非常成功的浅层嵌入方法是[54]线法，它不是基于随机游动，而是同时期的，经常与DeepWalk和node2vec进行比较。LINE结合了两个编码器-解码器目标，分别优化“一阶”和“二阶”节点相似性。一阶目标使用一个基于sigmoid函数的解码器，



以及基于邻接的相似度度量(即sG(vi, vj) = Ai,j)。二阶编码器-解码器的目标是相似的，但考虑两跳邻接邻域，并使用与方程(10)相同的编码器。利用kl散度度量[54]得到的损失函数优化了一阶和二阶目标。因此，LINE在概念上与node2vec和DeepWalk相关，因为它使用了概率解码器和丢失，但它明确地分解了一阶和二阶相似性，而不是将它们合并到固定长度的随机游动中。

**HARP:通过图形预处理扩展随机漫步嵌入**。最近，Chen等人[13]引入了一种称为HARP的“元策略”，通过一个图形预处理步骤来改进各种随机游走方法。在这种方法中，使用图粗化过程将G中的相关节点合并为“超级节点”，然后在粗化后的图上运行DeepWalk、node2vec或LINE。在嵌入粗化的G版本之后，学习到的每个超节点的嵌入被用作该超节点组成节点的随机游走嵌入的初始值(在对图的“细粒度”版本的另一轮优化中)。这个一般的过程可以在不同的粗糙级别上以分层的方式重复，并且已经被证明可以持续地提高DeepWalk、node2vec和LINE[13]的性能。

**随机行走想法的额外变体**。也有一些对随机游动思想的进一步扩展。例如,Perozzi et al。[48]扩展DeepWalk学习算法嵌入使用随机漫步“跳过”或“跳”到多个节点在每个步骤,导致相似度度量类似GraRep[9],而Chamberlan 等人。[11]修改node2vec的内积译码器使用双曲线,而非欧几里得距离测量。

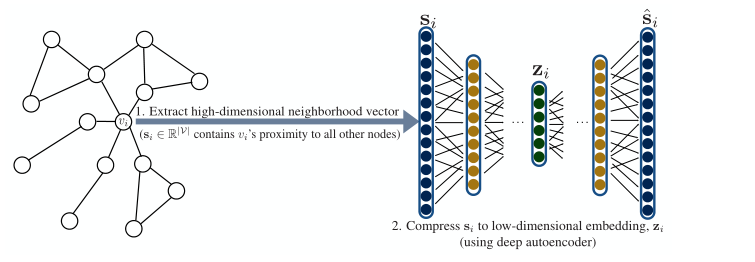
## 2.3广义的编码器架构

到目前为止，我们复习的所有节点嵌入方法都是浅层嵌入方法，其中编码器只是一个简单的嵌入查找(公式5)。然而，这些浅层嵌入方法独立地为每个节点训练唯一的嵌入向量，这导致了一些缺点:

1. 编码器中的节点之间不共享任何参数(即，编码器只是基于任意节点id的嵌入查找)。这在统计上是低效的，因为参数共享可以作为一种强大的正则化形式，而且它在计算上也是低效的，因为它意味着浅层嵌入方法中的参数数量必然会随着O(|V|)的增长而增长。

2. 浅嵌入也不能在编码期间利用节点属性。在许多大型图中，节点具有属性信息(例如社交网络上的用户配置文件)，这些属性信息通常与节点在图中的位置和角色有关。

3.浅埋方法本质上是可转导的[29]，即只能为训练阶段已经存在的节点生成嵌入，不能为之前未见的节点生成嵌入，除非对这些节点进行额外的轮优化来优化嵌入。这对于进化图、不能完全存储在内存中的大量图或需要在训练后泛化成新图的域来说是一个很大的问题。



**图6**:为了生成节点vi的嵌入，邻域自动编码器首先提取一个高维邻域向量si∈R|V|，总结了vi与图中所有其他节点的相似性。然后，si矢量通过一个深度的自动编码器馈入，以降低其维数，产生低维ziembedding

最近，提出了一些方法来解决这些问题的一部分或全部。这些方法仍坚定地在2.1节中概述的encoder-decoder框架,但他们不同于2.2节的浅嵌入方法,他们使用更复杂的编码器,通常基于深层神经网络和更一般的依赖图的结构和属性。

### 2.3.1邻域自编码器方法

Deep Neural Graph representation (DNGR)[10]和Structural Deep Network Embeddings (SDNE)[59]解决了上面提到的第一个问题:与浅层嵌入方法不同，它们直接将图结构合并到使用Deep Neural networks的编码器算法中。这些方法的基本思想是,他们使用autoencoders-a众所周知的方法深度学习[31]——为了压缩信息节点的本地社区(图6)。DNGR SDNE也不同于以前了方法,他们使用一元译码器代替一个成对的。

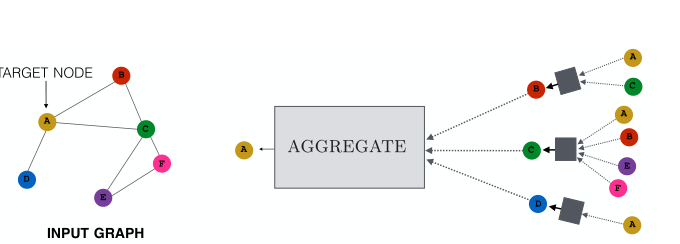
在这些方法中，每个节点vi都与一个邻域向量si∈R|V|相关联，它对应于矩阵S中vi所在的行(回想一下，S包含节点的双相似性，即si,j= sG(vi, vj))。si向量包含了vi与图中所有其他节点的相似性，并作为vi邻域的高维向量表示。DNGR和SDNE的autoencoder目标是使用sivector嵌入节点，这样sivector就可以从这些嵌入重建:



换句话说，这些方法的损失形式如下:



与成对解码器一样，ziembeddings的维数要比|V| (sivectors的维数)小得多，因此我们的目标是将节点的邻域信息压缩成一个低维向量。对于SDNE和DNGR，编码器和解码器函数都由多层堆叠的神经网络层组成:每层编码器降低其输入的维数，每层解码器增加其输入的维数(图6);有关深度自动编码器的概述，请参阅[31])。



**图7:**邻域聚集方法的编码概述。为了生成节点A的嵌入，模型聚合了来自节点A的本地图邻居(即B、C和D)的消息，反过来，来自这些邻居的消息基于从它们各自的邻域聚合的信息，以此类推。这一想法的“深度-2”版本被展示出来(即，信息是从围绕节点A的两跳邻居中聚合的)，但原则上这些方法可以是任意深度的。在最后的“深度”或“层”中，初始消息基于输入节点属性。

SDNE和DNGR的不同之处在于它们用来构造邻域向量siand的相似函数，以及如何优化自动编码器的具体细节。DNGR根据随机游动中同时发生的两个节点的点态互信息定义si根据，类似于DeepWalk和node2vec。SDNE只是简单地将si, Ai，即，设为vi的邻接向量。SDNE还结合了自动编码器目标(方程13)和拉普拉斯特征图目标(方程6)[59]。

注意，方程(13)中的编码器依赖于输入sivector，它包含关于vi的局部图邻域的信息。这种依赖性允许SDNE和DNGR将节点的本地邻居的结构信息作为一种正规化形式直接合并到编码器中，这对于浅层嵌入方法是不可能的(因为它们的编码器只依赖于节点id)。然而，尽管有了这些改进，自动编码器方法仍然受到一些严重的限制。最显著的是，autoencoder的输入维数固定在|和|，这对于有数百万节点的图来说是非常昂贵的，甚至是难以处理的。此外，由于该编码器的结构和尺寸是固定的，所以SDNE和DNGR具有严格的转导性，不能适应图的演变，也不能在图之间推广。

### 2.3.2邻域聚集和卷积编码器

一些最近的节点嵌入方法旨在解决浅层嵌入和自动编码器方法的主要局限性，设计编码器依赖于节点的局部邻域，但不一定依赖于整个图。这些方法背后的直觉是，它们通过聚合来自节点本地邻居的信息来为节点生成嵌入(图7)

与前面讨论的方法不同，这些邻域聚合算法依靠节点特征或属性(表示xi∈Rm)来生成嵌入。例如，一个社交网络可能有文本数据(例如，档案信息)，或者一个蛋白质-蛋白质相互作用网络可能有与每个节点相关的分子标记。邻域聚合方法利用这个属性信息来通知它们的嵌入。在没有给出属性数据的情况下，这些方法可以使用简单的图形统计作为属性(如节点度数)[29]，或者为每个节点分配一个单热点指标向量作为属性[36,53]。这些方法通常称为卷积方法，因为它们将节点表示为其周围邻居的函数，其方式类似于计算机视觉[35].5中的中心环绕卷积内核的接收域。

在编码阶段，邻域聚合方法以迭代或递归的方式为节点构建表示(参见算法1中的伪代码)。首先，初始化节点嵌入等于输入节点属性。然后在编码器算法的每次迭代中，节点使用一个对向量集进行操作的聚合函数来聚合它们相邻节点的嵌入。每次聚合后，给每个节点分配一个新的嵌入，等于其聚合的邻域向量与上一次迭代中嵌入的邻域向量相结合。最后，这个组合嵌入通过一个密集的神经网络层，并重复这个过程。随着流程的迭代，节点嵌入包含从图中越来越远的地方聚合而来的信息。然而，嵌入的维数在迭代过程中仍然受到限制，因此编码器被迫将所有的邻域信息压缩成一个低维向量。迭代K次后，过程终止，输出最终的嵌入向量作为节点表示。

有许多最近的方法遵循算法1中概述的基本过程，包括graph convolutional networks (GCN) [35, 36, 53, 56]， column networks[50]，和GraphSAGE算法[29]。可训练的参数算法1 a组聚合函数和一组权重矩阵{周,∀k∈(1,k)}指定如何从一个节点的本地社区聚合信息,与浅嵌入方法(2.2节),这些参数是跨节点共享。相同的聚合函数和权重矩阵用于为所有节点生成嵌入，只有输入节点属性和邻域结构会根据嵌入的节点而改变。这种参数共享提高了效率(即，参数的尺寸与图的大小无关)，提供了正则化，并允许使用这种方法为训练[29]期间没有观察到的节点生成嵌入。

GraphSAGE、列网络和各种GCN方法都遵循算法1，但主要区别在于如何执行聚合(第4行)和向量组合(第5行)。GraphSAGE在第5行中使用了连接并允许使用一般的聚合函数;作者使用element-wise平均数、最大汇聚神经网络和LSTMs[32]作为聚合器进行实验，他们发现更复杂的聚合器，尤其是最大汇聚神经网络，获得了显著的收益。GCNs和列网络在第5行使用加权和，在第4行使用(加权)元素平均数

列网络还在第7行之前添加了一个额外的“插值”项，即设置



其中，幅值为插值权值，计算为hk−1v和hk−1n (v)的非线性函数。这个插值项允许模型在过程迭代时保留局部信息(例如，随着k的增加和模型的集成)。

原则上，GraphSAGE、列网络和GCN编码器可以与前面讨论的任何解码器和损耗函数相结合，整个系统可以使用SGD进行优化。例如，Hamilton等[29]使用与node2vec相同的解码器和丢失函数，而Kipf等[36]使用与图分解方法相似的解码器和丢失函数。

在节点分类[29,35]和链路预测[56,36,53]的基准测试中，邻域聚集编码器跟随算法1被发现与它们的浅层嵌入对手相比提供一致的增益。在较高的层次上，这些方法解决了浅埋入的四个主要限制，在2.3节的开头提到:它们将图形结构合并到编码器中;它们利用节点属性;它们的参数维数可以在|V|中实现子线性;他们还可以为训练中没有出现的节点生成嵌入。

## 2.4实时任务监控

基本encoder-decoder框架描述到目前为止是默认情况下无人监督的,也就是说,模型优化,或训练,在设置节点对重建成对相似性值sG (vi, vj),它只取决于图g .然而,许多社区聚合方法在部分节点嵌入algorithms-especially 2.3.2-can也包含特定于任务的监督(29岁,35岁,53岁,60岁)。特别是，为了学习嵌入，通常的方法是将节点分类任务的监督纳入其中。6为了简单起见，我们讨论了节点具有关联的二进制分类标签的情况，但是我们描述的方法很容易扩展到更复杂的分类设置。

假设我们有一个二分类标签yi∈Z与每个节点关联。学会节点映射到他们的标签,我们可以养活我们嵌入向量,子,通过物流或乙状结肠,函数ˆ易=σ(z >我θ),θ是可训练的参数向量。然后我们可以计算这些预测的类概率和真实标签之间的交叉熵损失:



根据式(16)计算的梯度可以通过编码器反向传播，优化其参数。这种针对任务的监督可以完全替代解码器计算的重构损失(即公式3)[29,35]，也可以与解码器损失[60]一起纳入。

## 2.5对多模态图的扩展

虽然我们关注的是简单的、无向图，但许多真实世界的图具有复杂的多模态或多层结构(例如，异构节点和边缘类型)，而且许多工作已经引入了应对这种异构性的策略。

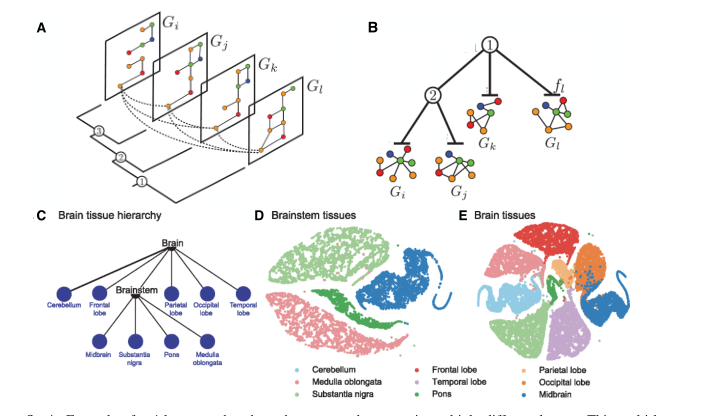
### 2.5.1处理不同节点和边缘类型

许多图包含不同类型的节点和边。例如，推荐系统图由两个不同的层组成——用户和内容——而许多生物网络有不同的层，它们之间有不同的交互作用(例如，疾病、基因和药物)。

处理这个问题的一般策略是(i)对不同类型[12]的节点使用不同的编码器，(ii)使用特定类型的参数对解码器进行扩展[43,53]。例如，在具有不同边类型的图形中，标准的内积边解码器（即z>izj≈Ai，j）可以用双线性形式代替[12，43，53]：



其中，可以对特定的边类型进行索引，而可以对特定于边的类型进行学习。公式(17)中的矩阵，一个可正则化的矩阵(例如，约束为对角)[53]，这在有大量的边类型时特别有用，比如在嵌入知识图的情况下。的确，知识图完成性方面的文献——其目标是预测知识图中缺失的关系——包含许多相关的技术，用于解码大量的边类型(即关系)[43]。7最近，Dong等人[19]也提出了一种从异构图中采样随机游动的策略，其中随机游动仅限于特定类型节点之间的转移。这种方法允许将2.2.2节中的许多方法应用于异类图，并且补充了包含特定类型的编码器和解码器的思想。



**图8**:一个4层图的例子，其中相同的节点出现在多个不同的层中。通过要求同一节点在不同层中的嵌入彼此相似，可以利用这种多层结构来规范不同层的学习。多层图可以呈现层次结构，其中层次结构中的非根层包含子层中边的并集。，一个来自整个人类大脑的生物交互图包含了额叶和颞叶相互作用的结合。这种结构可以通过在层次结构的各个层次上学习嵌入来利用，并且只在父子关系的层之间应用正则化。C-E，多层图嵌入不同脑组织蛋白质相互作用图实例应用C显示不同组织区域的层次结构，D和E显示脑干和全脑层生成的蛋白嵌入。利用多层欧姆网方法生成嵌入件，并利用t-SNE将嵌入件投影到二维空间。改编自[61]。

### 2.5.2跨层绑定节点嵌入

在某些情况下，图有多个包含相同节点副本的“层”(图8.A)。例如，在来源于不同组织(如大脑或肝脏组织)的蛋白质-蛋白质相互作用网络中，一些蛋白质存在于多个组织中。在这些情况下，跨层共享信息是有益的，这样，嵌入到一层的节点可以通过嵌入到其他层来得知。Zitnik等人[61]为这个问题提供了一种解决方案，称为欧姆网，它结合了node2vec和一个正则化惩罚，将嵌入层连接起来。具体来说，假设我们有一个节点vi，它属于两个不同的层g1和G2，我们可以将该节点的标准嵌入损失扩大如下:



式中，L为该节点通常的嵌入损失(如式8或式11)，其中，拟合项为正则化强度，zG1 i和zG2 i分别为vi在两层中的嵌入量。

Zitnik等人通过利用图层之间的层次进一步扩展了这个想法(图8.B)。例如，在来自各种组织的蛋白质-蛋白质相互作用图中，有些层对应于整个大区域的相互作用(例如，发生在任何脑组织中的相互作用)，而其他相互作用图则更为精细(例如，仅发生在额叶中的相互作用)。为了利用这个结构，可以在层次结构的各个层次学习嵌入，并且可以在层次结构中具有父-子关系的层之间递归应用等式(18)中的正则化。

## 2.6嵌入结构角色

到目前为止，我们复习过的所有方法都对节点嵌入进行了优化，使图中附近的节点具有类似的嵌入。然而，在许多任务中，更重要的是学习与节点的结构角色相对应的表示，独立于它们的全局图位置(例如，在通信或运输网络中)[30]。node2vec方法引入2.2.2节中提供了一个解决这个问题的,格罗弗等人发现偏置随机漫步,可以让他们的模型更好的捕捉结构角色(图5)。然而,最近,里贝罗et al。[51]和Donnat et al。[20]开发了节点嵌入方法是专门设计来捕获结构角色。

Ribeiro等人提出了struc2vec，它涉及生成一系列加权辅助图G0 k, k ={1,2，…}，其中辅助图G0 kcapture节点k-hop邻域之间的结构相似性。特别地，让Rk(vi)表示距离vi正好k跳的节点的度的有序序列，在辅助图G0 kare中，边权值wk(vi, vj)递归定义为



其中w0(vi, vj) = 0, d(Rk(vi)， Rk(vj))度量有序度序列Rk(vi)和Rk(vj)之间的“距离”(例如，通过动态时间扭曲[51]计算)。计算完这些加权辅助图后，struc2vec对它们进行有偏随机遍历，并将这些遍历作为node2vec优化算法的输入。

## 2.7节点嵌入的应用

节点嵌入最常见的用例是用于可视化、集群、节点分类和链接预测的，这些用例中的每一个都与许多应用领域相关，范围从计算社会科学到计算生物学。

**可视化和模式发现**。在2D界面中可视化图形的问题由来已久，在数据挖掘、社会科学和生物学[17]中都有应用。节点嵌入为图形可视化提供了一个强大的新范例:因为节点被映射到实值向量，研究人员可以轻松地利用现有的、通用的技术来可视化高维数据集[57,55]。例如，节点嵌入可以与t-SNE[57]或主成分分析(PCA)等众所周知的技术相结合，从而生成图形的2D可视化[47,54]，这对于发现社区和其他隐藏结构非常有用(图2和图8)。

**集群和社区检测**。与可视化类似，节点嵌入是聚类相关节点的强大工具，这项任务有无数的应用，从计算生物学(如发现相关药物)到市场营销(如发现相关产品)[23]。同样，因为每个节点都与实值向量嵌入相关联，所以可以将任何通用的聚类算法应用于已学习节点嵌入的集合(例如k-means或DB-scan[22])。这为传统的社区检测技术提供了一种开放的、强大的替代方法，而且还提供了新的方法机会，因为节点嵌入可以捕获不同节点所扮演的功能或结构角色，而不仅仅是社区结构。

**节点分类和半监督学习**。节点分类可能是用于评估节点嵌入的最常见的基准任务。在大多数情况下,节点分类的任务是semisupervised学习的一种形式,标签在哪里只有一小部分的节点,目标是将完整的图仅基于这个小初始种子集。semi-supervised节点分类的常见应用程序包括分类根据其生物功能的蛋白质[28]和分类文件, 视频、网页或个人进入不同类别/社区[28,35,47,54]，Hamilton等[29]最近引入了归纳节点分类的任务，其目标是对训练中没有看到的节点进行分类，例如在演化信息图时对新文档进行分类，或者将其推广到不可见的蛋白质-蛋白质相互作用网络。

**链接预测**。节点嵌入作为链接预测的特性也是非常有用的，其目标是预测缺失的边，或者预测未来[3]中可能形成的边。链接预测是推荐系统的核心，节点嵌入的常见应用体现了这种深度连接，如预测社交网络[54]中缺失的友情链接，用户对电影的亲和力[56]等。链路预测在计算生物学中也有重要的应用。许多生物相互作用图(例如蛋白质与其他蛋白质之间，或药物与疾病之间)是不完整的，因为它们依赖于从昂贵的实验室实验中获得的数据。预测这些噪声图中的链接是自动扩展生物数据集和为湿实验室实验[41]推荐新方向的重要方法。更普遍的是，链接预测与统计关系学习[24]密切相关，其中一个常见的任务是预测知识图[43]中实体之间缺失的关系。

# 3 嵌入子图

现在我们转向对(子)图进行表示学习的任务，目标是将一组节点和边编码为低维向量嵌入。更正式,学习的目标是一个连续的向量表示,z∈Rd,诱导子图G (S)的完整的图G,在年代⊆诉注意,这些方法可以嵌入两个子图(S⊂V)以及整个图(S = V)嵌入,z,可以用来预测整个子图;例如，可以嵌入对应于不同分子的图来预测其治疗特性[21]。

子图的表示学习与图核的设计密切相关，图核定义了子图[58]之间的距离度量。也就是说，我们省略了对图核的详细讨论，图核本身就是一个大而丰富的研究领域，读者可以参考[58]来获得详细的讨论。我们回顾的方法与传统的图核文献的主要不同之处是，我们寻求从数据中学习有用的表示，而不是通过核函数预先指定特征表示。

本节中的许多方法都基于在第2节中介绍的用于嵌入单个节点的技术。然而，与节点嵌入设置不同，大多数子图嵌入方法都是完全监督的，用于子图分类，目标是预测与特定子图关联的标签。因此，在本节中，我们将重点讨论生成zS嵌入的各种不同方法，并假设这些嵌入是通过交叉熵损失函数(类似于式(16))提供的。

## 3.1节点嵌入和卷积方法集

有几种子图嵌入技术可以看作卷积节点嵌入算法的直接扩展(在2.3.2节中描述)。这些方法背后的基本直觉是，它们将子图等同于节点嵌入集。他们使用卷积邻域聚合思想(即算法1)为节点生成嵌入，然后使用附加模块对子图对应的节点嵌入集进行聚合。本节中不同方法之间的主要区别是它们如何聚合对应于子图的节点嵌入集。

### 3.1.1 基于总和的方法

例如Duvenaud等人引入的“卷积分子指纹”[21]通过对子图中嵌入的所有单个节点求和来表示分子图表示的子图:



嵌入的,{Zi,∀vi∈S},使用生成算法1的一个变种。

DAI等人。[16]使用一个类似的sum-based方法但注意概念连接meanfield推理:如果图中的节点被视为潜在的变量在一个图形模型中,算法1可以被视为一种平均场推理的消息传递操作已经替换为可微的神经网络选择。基于这种联系，Dai等人[16]也提出了一种基于循环信念传播[42]的改进编码器。使用来自算法1的占位符和表示法，这种替代方法背后的基本思想是构造中间嵌入，即对应于边， 

然后将这些边嵌入聚合成节点嵌入:



一旦计算出了嵌入，Dai等人[16]使用简单的元素和来组合子图的节点嵌入，如公式(21)所示。

### 3.1.2图形粗化方法

Defferrard等人[18]和Bruna等人[8]也使用卷积方法，但他们没有对整个图的节点嵌入进行累加，而是堆叠卷积和“图粗化”层(类似于2.2.2节中的HARP方法)。在图粗化层中，节点被聚集在一起(使用任何图聚类方法)，并且聚集的节点嵌入使用元素的最大池进行组合。聚类之后，新的粗化图再次通过卷积编码器反馈，并重复这个过程。

与2.3.2中讨论的卷积方法不同，Defferrard等人[18]和Bruna等人[8]也非常重视基于图傅里叶变换[15]设计卷积编码器。然而，因为图的傅里叶变换需要识别和操纵图的拉普拉斯特征向量，这些方法的原始版本必须是O(|V|3)。对这些光谱方法的最先进的近似(例如，使用Chebyshev多项式)在概念上类似于算法1，只有一些小的变化，我们建议读者参阅Bronstein等人[7]来全面讨论这些技术。

### 3.1.3进一步变化

卷积思想的其他变体由Neipert等人[44]和Kearnes等人[34]提出。两种方法都提倡聚合子图对应的节点嵌入集的备选方法:Kearnes等人使用“模糊”直方图而不是总和来聚合节点集，他们也使用类似于[16]的边缘嵌入层。Neipart等人在节点上定义了一个顺序。使用特定问题的顺序或者使用现成的顶点着色算法——并且使用这种顺序，他们将所有节点的嵌入连接起来，并通过标准的卷积神经网络结构将这个连接起来的向量输入进来。

## 3.2图神经网络

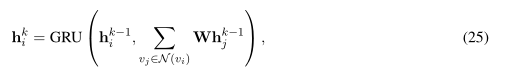
除了上面讨论的基于卷积的嵌入方法外，在“graph neural networks”(GNNs)[52]上还有一个相关的、按时间顺序优先的工作。在概念上，GNN思想与算法1密切相关。然而，GNNs背后的直觉是，可以将图看作是为节点之间的“消息传递”算法指定脚手架，而不是从邻居处聚合信息。

在原始的GNN框架[52]中，每个节点viis用一个随机嵌入的h0 i进行初始化，在GNN算法的每次迭代中，节点按照



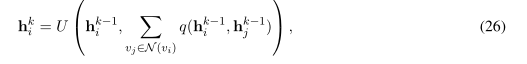
其中，h是形式为h的任意可微函数:Rd×Rm×Rm→Rd。式(24)以递归的方式反复应用，直到嵌入收敛，必须特别注意确保h是一个收缩映射。一旦嵌入聚合后K迭代,最终的输出映射进行计算是zvi = g(Hk),g是一个任意的可微函数的形式g: Rd→Rd Scarselli et al。[52]讨论各种参数化h和g基于多层感知器(mlp),尽管他们是有限的,需要迭代的收敛性和消息传递限制必须收缩映射f。

Li等人[39]对GNN框架进行了扩展和修改，使用门关递归单元和[14]时间反向传播，这样就不需要运行方程(24)来收敛。通过调整GNN框架来使用现代循环单元，还允许Li等人利用节点属性进行初始化，并使用子图中间嵌入的输出。特别地，Li等人的门控图神经网络使用节点属性(即h0 i= xi)初始化h0 ivector，并有形式的更新方程



其中W∈Rd×dis是一个可训练的权重矩阵，GRU为Cho等[14]引入的门控递归单元。

最后，Gilmer等人[25]讨论了gnn的另一种抽象，考虑了表单的模型



其中q: Rd×Rd→Rd0是可微分的函数，计算来自邻居的“消息”，U: Rd×Rd0→Rdis是可微分的“更新”函数。这个框架称为消息传递神经网络(MPNNs)，它概括了Li等人的门控图神经网络以及许多前面提到的卷积方法。Gilmer等人从基于分子图结构预测分子性质的角度，讨论了MPNNs的一些变体和扩展(例如，合并边缘特征)。

所有这些图神经网络方法原则上都可以用于节点级嵌入任务，尽管它们更常用于子图级嵌入。要计算子图嵌入，可以采用3.1节中描述的任何一种聚合过程，但Scarselli等人[52]也建议，可以通过引入一个“虚拟”超级节点来完成聚合，该超级节点连接到目标子图中的所有节点。

## 3.3子图嵌入的应用

子图嵌入的主要用例是子图分类，它在许多领域都有重要的应用。最突出的应用领域是对对应于不同分子的图的性质进行分类[16,21,44,34]。子图嵌入可用于分类或预测分子图的各种属性，包括预测潜在太阳能电池材料[16]的功效，或预测候选药物[34]的治疗效果。更普遍的是，子图嵌入被用于对图像进行分类(将图像转换为图形表示后)[8]，预测计算机程序是否满足某些形式性质[39]，以及执行逻辑推理任务[39]

# 4 结论与未来方向

图形机器学习的表示学习方法为传统的特征工程提供了一个强有力的替代方案。近年来，这些方法一直在推动诸如节点分类和链接预测等任务的最新发展。然而，还有很多工作要做，不仅要提高这些方法的性能，而且可能更重要的是，要发展一致的理论框架，以供未来的创新之用。

## 4.1对未来进步的挑战

在这篇综述中，我们试图统一之前的一些工作，但是这个领域作为一个整体仍然缺乏一个一致的理论框架——或者说是一组框架——来精确地描述图形表示学习的目标。目前，大多数工作的隐含目标是生成在特定分类或链接预测基准测试集上表现良好的表示(可能还会生成质量上令人满意的可视化)。然而，完全不同的基准和概念模型的泛滥给未来的发展带来了真正的风险，而在机器学习和数据挖掘领域中，节点和图嵌入技术的流行只会加剧这个问题。前进作为一个领域,需要新的理论工作,更精确地描述了我们希望学到的图结构表示编码,我们预计模型对这一信息进行编码,如何和什么约束(如果有的话)应该强加给这些学到的潜在空间。

更发达的理论基础不仅对该领域的研究人员有利。，通过通知一致的和有意义的基准任务——这些基础还将允许应用领域——专家们更有效地选择和区分不同的方法。当前的方法通常在各种不同的基准上进行评估，这些基准强调各种不同的图形属性(例如，社区结构、节点之间的关系强度或结构角色)。然而，许多真实的应用程序更加集中，并且没有必要使用对各种各样的任务通常有用的表示。作为一个领域，我们需要明确什么时候应该使用什么方法，并且规定这样的用例需要更精确的理论理解我们学到的表示是编码什么。

## 4.2重要开放问题

除了上述的一般性挑战之外，在图形表示学习领域还有许多具体的开放问题有待解决。

**可伸缩性**。虽然我们回顾的大部分工作在理论上都是高度可扩展的(例如，O(|E|)训练时间)，但在缩放节点和图嵌入方法以实现真正的海量数据集(例如，数十亿节点和边)方面仍有大量工作要做。例如，大多数方法依赖于训练和存储每个单独节点的唯一嵌入。此外，大多数评估设置都假设用于训练和测试的所有节点的属性，嵌入和边缘列表都可以放入主内存中，这一假设与大多数应用程序领域的实际情况不符，因为在这些领域中，图是庞大的，不断发展的， 并且通常以分布式方式存储。开发真正可扩展到实际生产设置的表示学习框架是必要的，以防止扩大学术研究社区和这些方法的应用程序消费者之间的脱节。

**解码高阶主题**。虽然近年来很多工作都致力于改进和改进用于生成节点嵌入的编码器算法，但大多数方法仍然依赖于基本的成对解码器，这种解码器预测节点之间的成对关系，并忽略涉及两个以上节点的高阶图结构。众所周知，高阶结构基序对复杂网络[5]的结构和功能至关重要，开发能够对复杂基序进行解码的译码算法是未来工作的一个重要方向。

**动态建模，时间图**。许多应用领域涉及高度动态的图形，其中时间信息是关键的。、即时消息网络或金融交易图表。然而，我们缺乏嵌入方法来应对时序图所带来的独特挑战，例如合并边缘的时序信息。时间图正成为[46]越来越重要的研究对象，扩展图嵌入技术对其进行操作，将开辟一个广泛的应用领域。

**对大量候选子图集的推理**。当前的子图嵌入方法的一个主要技术限制是需要在学习过程之前预先指定目标子图。然而，许多应用程序寻求发现具有某些属性的子图，并且这些应用程序需要能够对可能的候选子图的组合大空间进行推理的模型。例如，人们可能想要在基因调控网络中发现中心子图，或者在社交网络中发现邪恶的子社区。我们需要改进的子图嵌入方法，能够有效地对大量候选子图集进行推理，因为这种改进对于扩展子图嵌入的有用性，使之超出基本子图分类的任务是至关重要的。

**改善可解释性**。表示学习很有吸引力，因为它减轻了手工设计特性的负担，但它也付出了众所周知的可解释性代价。我们知道，基于嵌入的方法可以提供最先进的性能，但是这些算法的基本限制和潜在的偏差相对来说是未知的。为了向前发展，必须注意开发新的技术，以提高已学习表征的可解释性，超越可视化和基准评估。考虑到这些方法的复杂性和代表性能力，研究人员必须时刻保持警惕，确保他们的方法真正学会了表示相关的图表信息，而不仅仅是利用基准的统计趋势。