概要

从电信网络到量子化学，图形结构化数据在自然科学和社会科学中无处不在。如果我们希望系统可以从此类数据中学习，推理和推广，则在深度学习体系中建立关系归纳偏差至关重要。近年来，涌现了大量关于图表示学习的研究，其中包括用于深度图性嵌入技术，卷积神经网络对图形结构数据的概括以及受信念传播启发的神经信息传递方法。图表示学习的这些进步催生了许多领域的最新成果，包括化学合成，3D视觉，推荐系统，人机问答和社交网络分析。

本书旨在对一种关于图表示学习进行概述。我们首先讨论图形表示学习的目标，以及图理论和网络分析的主要方法论基础。在此之后，我们介绍并回顾了用于学习节点嵌入的方法，包括基于随机游走的方法和对知识图的应用。然后，我们对取得高度成功的图神经网络（GNN）形式进行了技术综合和介绍，而该形式已成为使用图形数据进行深度学习的主要且快速增长的典范。本书最后总结了深度生成模型的最新进展情况，这个模型是新生的，但是是增长迅速的图形表示学习的子领域。

前言

在过去七年中，图表示学习领域以一种时而不可思议，时而笨拙缓慢的步伐向前发展。我第一次接触这个领域是在2013年读本科的时候，在那期间，大量的研究员开始对关于嵌入图形结构数据的深度学习方法进行研究。自2013年以来，从标准图神经网络典范的发展到关于图结构化数据的深度生成模型的初期工作，图表示学习领域见证了一种真正的令人印象深刻的崛起和扩张。这个领域已经从从事相对利基研究的一小部分研究人员转变为深度学习增长最快的子领域之一。

然而，随着领域发展，我们对于图表示学习的底层方法和理论的理解也随之向后伸展。现在，我们可以将流行的“节点嵌入”方法看作是关于降维的经典著作的很好理解的扩展。现在，我们从图谱理论，调和分析，变分推论和图同构理论的丰富历史工作线中，对图神经网络如何演化（有些独立）进行了理解和赞赏。这本书是我的以一种实践方式关于综合分析和总结这些方法论的尝试。我希望将读者引入到当前实践的领域，与此同时也能将实践与机器学习及之外的更广泛的历史研究相联系。

# 第一章

## 简介

图是无处不在的数据结构，是描述复杂系统的通用语言。在大量常规的视图中，图形是一个伴随有若干个两两交互的，简单的对象的集合。比方说，将社交网络编码为图表，我们也许会使用节点表示个体，使用边来表示两个个体之间的关系。在生物学领域，我们使用图中节点表示蛋白质，使用边来表示两个生物个体之间的相互作用，比如蛋白质之间的动力学相互作用。

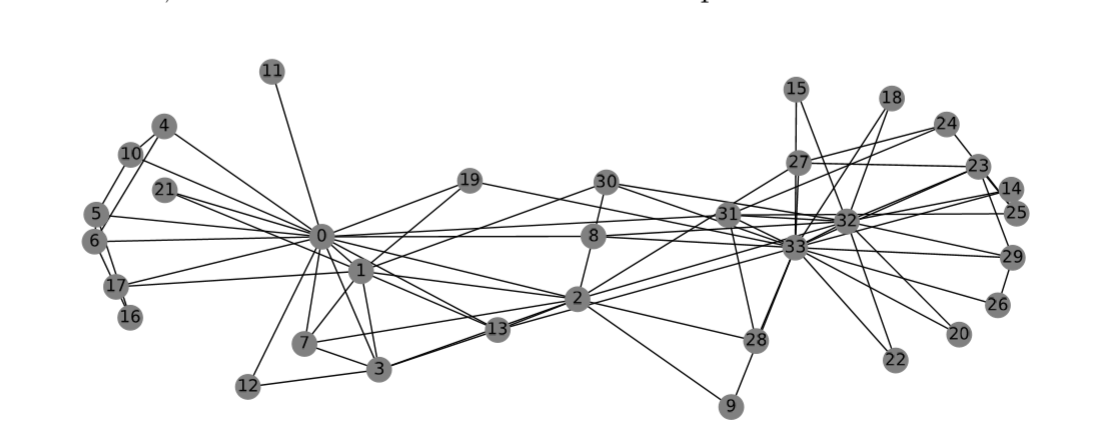


图1.1著名的Zachary空手道俱乐部网络代表了1970年至1972年间由Wayne W. Zachary研究的一个空手道俱乐部成员之间的友谊关系。如果两个人在俱乐部外交往，则用边将它们连接起来。 在Zachary的研究过程中，俱乐部分为以节点0和33为中心的两个派系，并且Zachary能够根据图结构正确预测节点是哪个派系[Zachary，1977]

图形式论的力量不仅在于它关注节点之间的关系（而不是单个节点的属性），还在于它的普遍性。同样的图形式论能被用来表示社交网络，下面举几个例子，例如药品和蛋白质之间相互作用，分子中原子的相互作用关系，通信网络中不同终端间的联系。

然而，图不仅能提供一个好的理论框架，还能做很多事情。他们提供了数学基础，这使得我们能从真实的复杂系统中继续从事分析、理解和学习。在过去25年中，对于研究者可用的图形结构数据在数量和质量上都在急剧增长。随着大规模社交网络平台的出现，对交互组，食物网，分子图结构的数据库以及数十亿个互连的基于网络的设备进行建模的大规模科学计划，不乏有意义的图形数据供研究人员进行分析。挑战在于释放这些数据的潜力。

这本书是关于我们如何能使用机器学习去处理这种挑战。当然，机器学习不是分析图形数据唯一的方法。但是，鉴于我们试图分析的图形数据集的规模和复杂性不断增加，可以清楚的一点是机器学习将会在提升我们建模、分析和理解图形数据的能力上发挥重要作用。

### 1.1图是什么？

在讨论关于图的机器学习之前，有必要再给出“图数据”更正式的描述。严格地讲，图被定义为一系列地节点和在这些节点间一系列的边。我们把从节点到节点作为一条边()∈。在大部分情况下，我们仅仅需要关注简单图，这些简单图绝大部分情况下，在每两个节点间只有一条边，节点与自己间没有边，且所有边都是无方向的，等等，。

一种表示图的快捷方式是通过一个邻接矩阵。为了用邻接矩阵表示图，我们给节点排序以至于每个节点都能处在邻接矩阵特定行和列的位置上。然后，我们可以将边的存在表示为该矩阵中的条目：当()∈时，，否则。如果图仅包含无向边，则是一个对称矩阵，但是如果图是一个有向的，那么就不对称。有一些图还包含权重边，那么在邻接矩阵中的条目可能是任意的真实值，而不0或1。例如，在蛋白质与蛋白质相互作用中，一个有权重的边可以表示两种蛋白质之间关联的强度。

#### 1.1.1多关系图

除了无向边，有向边和加权边的区别之外，我们还将考虑具有不同边类型的图。例如，在表示药物间的相互作用的图形中，我们可能希望不同的边对应于同时服用一对药物时可能发生的不同副作用。在这些情况下，我们可以扩展边符号使得能够包括边或者关系类型τ等等，，并且可以为每种边类型定义一个邻接矩阵。我们称这种图为多关系图，并且整个图可以用邻接张量来概括，其中是关系集。多关系图的两个重要子集为异构图和多重图。

**异构图** 在异构图中，节点也代表不同的类型，这意味着我们可以将节点集合划分为不相交的集合，其中。异构图中的边通常根据节点类型满足约束，最常见的约束是某些边只连接某些类型的节点，即。例如，在异质生物医学图中，可能一类节点代表蛋白质，一类代表药物，一类代表疾病。代表“治疗”的边只会出现在药物节点和疾病节点之间。类似地，代表“多药副作用”的边只会出现在两个药物节点之间。多部图是最广为人知的一种特殊的异构图，其中边只连接不同类型的节点，即。

**多重图** 在多重图中，我们假设该图可以分解为一组k层。假定每个节点都属于每个层，并且每个层对应一个唯一关系，表示该层的层内边类型。我们还假设存在层间边类型，不同层间通过同一节点进行跨层连接。通过示例可以更好地理解多重图形。例如，在多重运输网络中，每个节点可能代表一个城市，而每一层可能代表不同的运输方式（例如，航运或火车）。层内边将代表通过不同交通方式连接的城市，而层间边则代表在特定城市内转换交通方式的可能性。

#### 1.1.2 特征信息

最后，在许多情况下，我们还具有与图相关的属性或特征信息（例如，在社交网络中与用户关联的个人资料图片）。大多数情况下，这些是节点级别属性，我们使用实值矩阵表示，其中我们假定节点的顺序与邻接矩阵中的顺序是一致的。在异构图中，我们通常假定每种不同类型的节点都有其自己独特的属性类型。在极少数情况下，我们还将考虑除离散边类型外还具有实值边特征的图，在某些情况下，我们甚至将实值特征与整个图相关联。

图还是网络？在本书中我们使用术语“图”，但是你将看到在许多其他资料中使用术语“网络”来描述相同类型的数据。在某些地方，我们会同时使用这两个术语（例如，用于社交或生物网络）。那么哪个术语是正确的？在许多方面，这种术语差异是一种历史和文化上的差异：“图”一词在机器学习社区中似乎更为流行，但是“网络”在历史上一直在数据挖掘和网络科学界流行。本书中我们都使用了这两个术语，但我们也区分了这些术语的用法。我们使用术语“图”来描述本书所关注的抽象数据结构，但是我们也会经常使用术语“网络”来描述该数据结构的特定的，在现实中的实例化（例如，社交网络）。这种术语上的区别与它们当前流行的用法相吻合。网络分析通常与现实世界数据的属性有关，而图论与数学图形抽象的理论属性有关。

### 1.2 机器学习在图上的应用

机器学习本质上是一个问题驱动的学科。我们寻求构建可以从数据中学习的模型以解决特定任务，并且机器学习模型通常根据其寻求解决的任务类型进行分类：它是根据输入数据来预测输出结果的受监督的任务吗？它是目标是推断数据中的模式（例如点簇）的无监督的任务吗？

使用图进行机器学习没有什么不同，但是对于图来说，有监督和无监督的通常类别不一定是最有用的信息。在本节中，我们简要概述了图的数据上最重要且经过充分研究的机器学习任务。就像我们将看到的那样，“监督”问题在图数据中很普遍，但是图上的机器学习问题和传统机器学习类别之间的界限是模糊的。

#### 1.2.1 节点分类

假设我们有一个拥有数百万用户的大型社交网络数据集，但是我们知道这些用户中有很大一部分实际上是机器人。识别这些机器人很重要，原因如下：公司可能不想给机器人做广告，或者机器人实际上也可能违反了社交网络的服务条款。但是，手动检查每个用户是否是机器人的成本非常昂贵，因此，理想情况是，我们想要建立一个可以通过少量标签示例辨别出是机器人（或者不是）。

这是一个节点分类的经典示例，当我们仅在节点训练集中获得真实标签时，我们的目标是预测与所有节点相关联的标签（可以是类型，类别或属性）。节点分类可能是图数据上最流行的机器学习任务，尤其是近年来。在社交网络以外的节点分类示例中，包括对交互作用中对蛋白质功能进行分类[Hamilton等，2017b]，并基于超链接或引文图对文档主题进行分类[Kipf and Welling，2016a]。通常，我们假设在一个简单图中，我们仅对其中很小一部分节点带有标签信息（例如，从一小组手动标签示例中对社交网络中的机器人进行分类）。但是，也有节点分类的情况，这涉及许多标记的节点和/或需要在不连续的图中进行概括（例如，在不同物种的相互作用中对蛋白质的功能进行分类）。

乍一看，节点分类似乎是标准监督分类的直接变体，但实际上存在重要区别。最重要的区别是图中的节点不是独立和均匀分布的。通常，当我们建立监督机器学习模型时，我们假设每个数据点在统计上都独立于所有其他数据点。否则，我们可能需要对所有输入点之间的依赖关系进行建模。我们还假设数据点是相同分布的；否则，我们没法将模型推广到新的数据点。然后，节点分类彻底打破了这一假设。我们不是建模一组独立且分布均匀的数据点，而是改为建模一组互连的节点。

实际上，绝大部分最成功的节点分类方法背后的关键点是显式地利用节点之间的连接。一种特别流行的思想是利用同构性，即节点趋向于与图中的相邻节点共享属性[McPherson等，2001]。例如，人们倾向于与拥有相同兴趣或人口特征的人做朋友。基于同构的概念，我们可以建立机器学习模型，尝试为图中的相邻节点分配相似的标签[Zhou et al。，2004]。除了同构以外，还存在诸如结构等价的概念[Donnat et al。，2018]，即具有相似局部邻域结构的节点将具有相似标签和异质性，这意味着节点将优先连接到具有不同标签的节点。当我们建立节点分类模型时，我们要利用这些概念并对节点之间的关系进行建模，而不是简单地将节点视为独立的数据点。

监督或半监督？由于节点分类的非典型性，研究人员经常将其称为半监督型[Yang等，2016]。之所以使用此术语，是因为当我们训练节点分类模型时，我们通常可以访问完整的图，包括所有未标记的（例如，测试）节点。我们唯一缺少的是测试节点的标签。但是，我们仍然可以使用有关测试节点的信息（例如，了解图中的邻域）在训练过程中改善模型。这与通常的监督设置不同，后者在训练过程中完全看不到未标记的数据点。

用于在训练期间将带标签的数据和未带标签的数据进行合并的模型称为半监督学习，因此可以理解，该术语经常用于引用节点分类任务。不过，必须注意的是，半监督学习的标准公式仍然需要节点独立且均匀分布这一假设，这不适用于节点分类。图的机器学习任务很难满足我们的标准类别！

#### 1.2.2 关系预测

节点分类对于基于节点与图中其他节点的关系来推断有关节点的信息很有用。但是，如果我们缺少此关系信息，情况又如何呢？如果我们只知道给定细胞中存在的一些蛋白质相互作用，但我们想对我们缺少的相互作用做出一个很好的猜测，该怎么办？我们可以使用机器学习来推断图中节点之间的边吗？

根据特定的应用领域，此任务可以有很多别称，如链接预测，图完成和关系推断。在这里，我们将其简称为关系预测。与节点分类一起，它是带有图数据的应用教普遍的机器学习任务之一，并且在现实世界中有无数的应用：向社交平台中的用户推荐内容[Ying et al., 2018a]，预测药物的副作用[Zitnik et al., 2018]，或在关系数据库中推断新关系因子[Bordes et al., 2013]，所有这些任务都可以视为关系预测的特殊案例。

关系预测的标准设置给我们提供了一组节点和这些节点之间的不完整的边集合。我们的目标是使用此部分信息来推出缺失的边集合\。此任务的复杂性很大程度上取决于我们正在查看的图数据类型。例如，在简单图中，比如仅仅对“友谊”关系进行编码的社交网络，存在基于两个节点共享多少邻接节点可以实现出色性能的简单启发式方法[and Zhou, 2011]。另一方面，在更复杂的多关系图的数据集中，例如编码数百种不同生物学相互作用的生物医学知识图，关系预测可能需要复杂的推理和推理策略[Nickel et al., 2016]。像节点分类一样，关系预测模糊了传统机器学习类别的界限（通常被称为有监督的和无监督的），并且它需要特定图域的归纳偏差。另外，像节点分类一样，关系预测有很多变体，包括在单个固定图上进行预测的设置[and Zhou, 2011]，以及必须在多个不连续图上关系预测的设置[Teru et al., 2020]。

#### 1.2.3 聚类与社区检测

节点分类和关系预测都需要推断出关于图的数据的丢失信息，并且在许多方面，这两个任务是监督学习的图的相似体。另一方面，社区检测是无监督聚类的图的相似体。

假设我们可以访问谷歌学术中的所有引用信息，并且如果两个研究人员共同撰写了一篇论文，那么我们可以制作一个协作图，将两个研究人员联系起来。如果我们要研究这个网络，我们是否会期望找到一个密集的“毛线球”，让每个人都同样有可能与其他人合作？该图更有可能分离为不同的节点簇，并按研究领域，机构或其他人口统计学因素进行分组。换句话说，我们希望该网络（就像许多现实世界中的网络一样）表现出社区结构，其中节点更有可能与属于同一社区的节点形成边。

这是社区检测任务的基本直觉。社区检测的挑战在于仅在给出输入图的情况下，推断潜在的社区结构。社区检测的许多实际应用包括发现遗传交互网络中的功能模块[Agrawal et al., 2018]和发现金融交易网络中的欺诈性行为[Pandit et al., 2007]。

#### 1.2.4 图分类、回归和聚类

在有关图的机器学习应用程序的最后一类涉及整个图的分类、回归或聚类问题。例如，给定一个表示分子结构的图，我们可能想要建立一个回归模型来预测该分子的毒性或溶解性[Gilmer et al., 2017]。或者，我们可能希望通过分析基于语法和数据流的图形表示来建立分类模型，以检测计算机程序是否为恶意软件。在这些图分类或回归应用中，我们试图学习图数据，但不是对一个简单图的各个组成部分（即节点或边）进行预测，我们是给多个不同图的数据集，我们的目标是针对每个图形具体做出独立的预测。在图聚类的相关任务中，目标是学习无监督的成对的图之间的相似性度量。

在图的所有机器学习任务中，图的回归和分类问题也许是标准监督学习的最直接的类似物。每个图都是一个独立且均匀分布的与标签相关联的数据点，目标是使用一组带标签的训练点来学习从数据点（即图形）到标签的映射。以类似的方式，图聚类是对数据的无监督的聚类问题的直接扩展。但是，这些图 级任务的挑战是，如何定义有用功能使得每个数据点间关系结构都考虑进去

# 第二章

## 背景及传统方法

在介绍图表示学习和图的深度学习的概念之前，有必要提供一些方法论背景和上下文介绍。在现代深度学习方法问世之前，图使用过哪些类型的机器学习方法？在本章中，我们将对图的传统学习方法进行非常简短且有针对性的浏览，在此过程中为能更彻底处理方法论方法提供指示和参考。本章还将介绍图分析中的关键概念，这些概念将为以后的章节奠定基础。

我们的学习旅行将大致与图表上的各种学习任务保持一致。我们将从基本图形统计，内核方法及其在节点和图形分类任务中的使用开始讨论。接下来，我们将介绍并讨论用于测量节点邻域之间重叠的各种方法，这些方法构成了用于关系预测的强大启发式方法的基础。最后，我们将通过使用图的拉普拉斯对光谱聚类进行简要介绍来结束本背景部分。光谱聚类是对图进行聚类或社区检测方面研究最深入的算法之一，我们关于技术的讨论将会引入一些关键的数据概念，而这些概念将会在全书中反复出现。

### 2.1 图的统计和内核方法

使用图数据进行传统分类的方法，遵循在深度学习出现之前流行的标准机器学习范式。 我们首先基于启发式函数或领域知识提取一些统计信息或特征，然后将这些特征用作标准机器学习分类器的输入（例如逻辑回归）。 在本节中，我们将首先介绍一些重要的节点级特征和统计信息，然后我们将讨论如何将这些节点级统计信息推广到图级统计信息并扩展到设计图的内核方法。 我们的目标是引入各种启发式统计数据和图形属性，这些属性通常用作应用于图的传统机器学习管道中的功能。

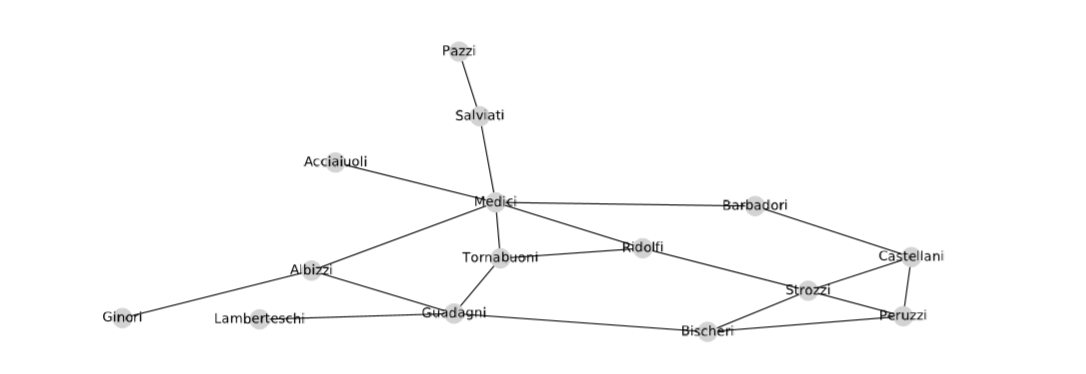


图2.1十五世纪佛罗伦萨不同知名家庭之间婚姻的可视化图

#### 2.1.1 节点级统计和功能

继Jackson [2010]之后，我们将通过一个简单（但很有名）的社交网络（15世纪佛罗伦萨婚姻网络）来开始我们对节点级统计和功能的讨论（图2.1）。这个社会网络是众所周知的，这是因为Padgett和Ansell [1993]的工作，他们利用这个网络来说明了统治佛罗伦萨政治的美第奇家族（描绘在中心附近）的权力上升。政治婚姻是在美第奇时代巩固权力的重要途径，因此这种婚姻联系网络在很大程度上决定了这段时期的政治结构。

为了我们的目的，我们将从机器学习的角度考虑这个网络和Medici的兴起，并提出一个问题：机器学习模型使用了哪些功能或统计数据来预测Medici的兴起？换句话说，Medici节点的哪些属性或统计将其与图的其余部分区分开？而且，更笼统地说，我们能使用哪些有用的属性和统计信息来表示此图中的节点？

原则上，我们下面讨论的属性和统计数据可用作节点分类模型中的特征（例如，用作逻辑回归模型的输入）。当然，我们将无法在像佛罗伦萨婚姻网络一样小的图上实际地训练机器学习模型。但是，考虑可用于区分这种现实网络中的节点的功能种类仍然具有说明性，而且我们讨论的属性通常可用于各种节点分类任务。

**节点度** 要检查的最明显，最直接的节点特征是度，通常将度表示为节点u∈V，简单地统计连接此节点的边数：

(2.1)

注意，在有向图和加权图的情况下，可以通过对等式（2.1）中的行或列中对应出度或入读求和来区分不同的度概念。 通常，节点的度是要考虑的基本统计信息，并且通常是应用于节点级任务的传统机器学习模型中最有用的功能之一。

在我们的佛罗伦萨婚姻说明图中，我们可以看到，度确实是区分美第奇家族的一个好功能，因为它们在图中具有最高的度数。但是，他们的学位与Strozzi和Guadagni两个最接近的家族比例仅为3:2。是否可能有其他或更多区分性的特征可以帮助将Medici家族与其他图表区分开？

**节点中心度** 节点度只是衡量一个节点有多少个邻节点，但这并不一定足以衡量图中一个节点的重要性。在许多情况下（例如我们的佛罗伦萨婚姻示例图），我们可以从其他更有效的节点重要性度量中受益。为了获得更强大的重要性度量，我们可以考虑各种所谓的节点中心性度量，这些度量可以在各种节点分类任务中形成有用的功能。

一个普遍的和重要的中心度衡量被称为所谓的特征向量中心度。 然而度只是衡量每个节点有多少个邻接点，而特征向量中心性同时还考虑到节点的邻节点有多重要。 特别是，我们通过递归关系定义节点的特征向量中心度，其中节点的中心度与其邻节点的平均中心度成正比：

(2.2)

其中λ是一个常数。将e作为节点中心点的向量，我们用向量符号重写该方程,我们可以看到这种重复定义了邻接矩阵的标准特征向量方程：

(2.3)

换句话说，中心性度量能满足公式2.2中的递归对应于邻接矩阵的特征向量。 假设我们需要正中心值，我们可以应用Perron-Frobenius定理1进一步确定中心值e的向量是由对应于A的最大特征值的特征向量给定的。

特征向量中心度的一种观点是，它对图中的无限长的随机游走中访问一个节点的可能性进行排名。该视图可以通过考虑使用幂迭代来获得特征向量中心值来说明。 也就是说，由于λ是A的前导特征向量，因此我们可以使用幂迭代来计算e:

(2.4)

如果我们以向量进行本次幂迭代，那么我们可以看到，在第一次迭代之后，将包含所有节点的度数。通常，在迭代t≥1时，将包含到达每个节点的长度为t的路径数。因此，通过无限期地迭代此过程，我们得到一个分数，该分数与在无限长的路径上访问节点的次数成正比。节点的重要性，随机游动和图邻接矩阵的光谱之间的这种联系将在本书的后续各节和各章中经常提到。

回到佛罗伦萨婚姻网络的例子，如果我们在此图上计算特征向量中心值，则我们再次看到美第奇家族的影响最大，与第二高值为0.36相比，它的值达到0.43。当然，我们还可以使用其他中心值度量来表示该图中的节点，这些节点相对于美第奇家族的影响力更为明显。这些包括相互之间的中心度（betweenness centrality）（它衡量一个节点位于两个其他节点之间的最短路径上的概率）以及邻近性中心度（closeness centrality）（它衡量一个节点到其他所有节点之间的平均最短路径长度）。如果节点到图中其他节点的最短距离都很小，那么它的接近中心性就很高。相比中介中心性，接近中心性更接近几何上的中心位置。这些衡量方法及更多其他方法的详情请见纽曼[2018]。

**聚类系数** 衡量的重要性，例如度和中心值，对于区分著名的美第奇家族与其他佛罗伦萨婚姻网络显然很有用。但是，对于区分图形中其他节点有用的功能又如何呢？例如，图中的Peruzzi和Guadagni节点具有非常相似的度数（3对4）和相似的特征向量中心度（0.28对0.29）。 但是，从图2.1中的图表来看，这两个家族之间存在明显的差异。 佩鲁兹（Peruzzi）家族处于相对紧密的家族集群之中，而瓜达尼（Guadagni）家族则扮演着更像“明星”的角色。

1 Perron-Frobenius定理是线性代数的基本结果，由Oskar Perron和Georg Frobenius独立证明[Meyer，2000]。 完整定理有很多含义，但是出于我们的目的，定理中的关键断言是任何不可约方阵都具有的唯一的最大实特征值，这是唯一可以选择其对应特征向量具有严格正分量的特征值。

2请注意，为简单起见，我们忽略了幂迭代计算中的归一化，因为这不会改变主要结果。

可以使用聚类系数的变化来衡量这种重要的结构区别，聚类系数用于衡量节点的局部邻域中闭合三角形的比例。聚类系数局部变量比较流行的计算公式如下[Watts and Strogatz，1998]：

（2.5）

此等式中的分子计算节点的邻居之间的边数（此处，我们使用表示节点邻域）。分母计算出的邻居中有多少对节点。

聚类系数的名称源于以下事实：它可以测量节点邻域的**聚类紧密程度**。聚类系数为1表示的所有邻居也是彼此的邻居。在我们的佛罗伦萨婚姻图中，我们可以看到一些节点高度聚类（例如，Peruzzi节点的聚类系数为0.66），而其他节点（例如Guadagni节点）的聚类系数为0。就中心度而言，存在许多变化聚类系数（例如，考虑有向图）的数量，Newman [2018]也对其进行了详细审查。整个社会和生物学科学中，现实世界网络的一个有趣且重要的特性是，它们的聚类系数往往比通过随机采样的边会产生的聚类系数高得多（Watts和Strogatz，1998年）。

**闭合三角形，本我图和图案**

查看聚类系数的另一种方法（而不是衡量局部聚类）是，它计算每个节点的局部邻域内的闭合三角形的数量。用更精确的术语来说，聚类系数与节点的本我图中（即包含该节点，其邻居以及该节点中节点之间的所有边的子图）中的实际三角形数与三角形的可能总数之比有关，即：包含该节点，其相邻节点以及其相邻节点之间的所有边的子图

这种想法可以推广到对节点的本我图中的任意图案或小图形进行计数的概念。也就是说，我们不仅可以考虑三角形，还可以考虑更复杂的结构，例如特定长度的循环，并且可以通过计数这些不同的图案在其本我图中出现的频率来表示节点。实际上，通过以这种方式检查节点的本我图，我们可以将计算节点级统计信息和特征的任务从本质上转变为图形级任务。因此，我们现在将注意力转向这个图形级问题。

#### 2.1.1 图级特征和图内核

到目前为止，我们已经讨论了节点级的各种统计信息和属性，这些统计信息和属性可用作节点级分类任务的功能。但是，如果我们的目标是进行图级分类怎么办？例如，假设我们得到了代表分子的图形的数据集，而我们的目标是根据它们的图形结构对这些分子的溶解度进行分类。我们将如何做？在本节中，我们将简要介绍为此类任务提取图形级特征的调查方法。

许多调查方法都属于图内核方法的一般分类，图内核方法是设计可用于机器学习模型的图或隐式内核函数的特征的方法。我们将仅涉及在这个大区域内的一小部分方法，并且我们将专注于提取显式特征表示的方法，而不是定义图之间的隐式内核（即相似性度量）的方法。

**包节点**

定义图形级功能的最简单方法是仅聚合节点级统计信息。例如，可以计算直方图，或者基于图中节点的度，中心值和聚类系数来计算其他汇总统计信息。然后，可以将这种汇总信息用作图形级表示。这种方法的缺点是它完全基于局部节点级别的信息，并且可能会忽略图中的重要全局属性。

**Weisfieler-Lehman内核**

一种改进基本包节点的方法是使用**迭代邻域聚合策略**。这些方法的思想是提取节点级特征，这些节点级特征不仅包含其局部本我图，而且还包含更多信息，然后将这些更丰富的特征聚合到图级表示中。这些策略中最重要和最著名的也许是WeisfielerLehman（WL）算法和内核[Shervashidze等人，2011； Weisfeiler和Lehman，1968]。 WL算法的基本思想如下：

1.首先，我们为每个节点分配一个初始标签。 在大多数图中，该标签只是度数，即。

2.接下来，我们通过对节点附近的当前标签的多集合进行散列来迭代地为每个节点分配一个新标签：，其中双括号用于表示多集，并且函数将每个唯一的多集合映射到唯一的新标签。

3.在运行了次迭代（即步骤2）之后，我们现在为每个节点添加了一个标签，该标签总结了其**阶邻域**的结构。然后，我们可以计算这些标签上的直方图或其他汇总统计信息，作为图形的特征表示。换句话说，WL内核是通过测量两个图的结果标签集之间的差异来计算的。

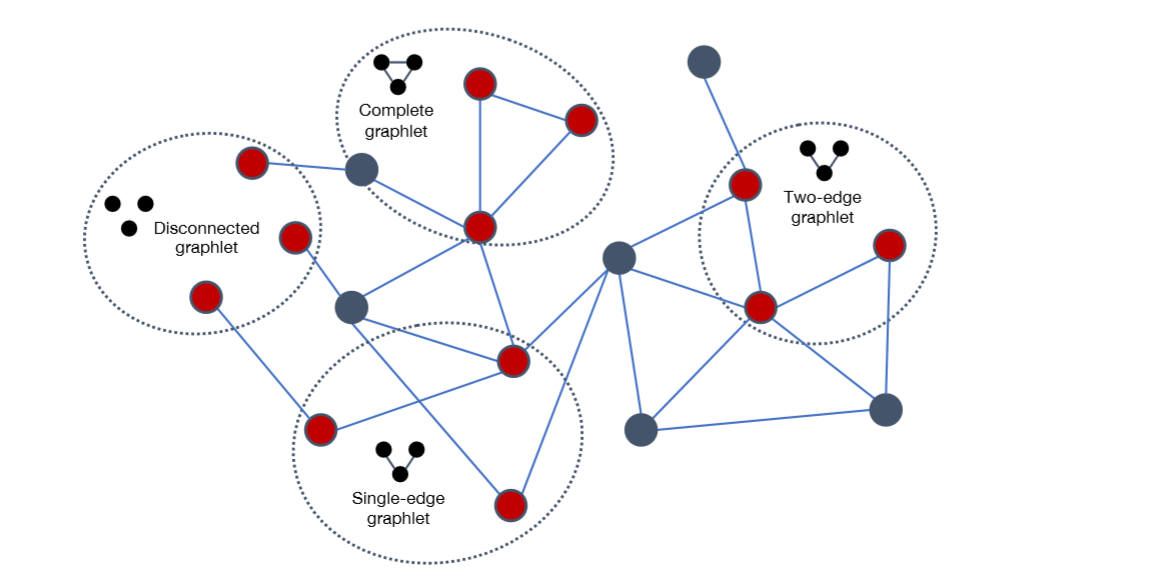


图2.2一个简单图中可能出现的四个不同的size-3 graphlet

WL内核是流行的，经过深入研究的，并且具有重要的理论特性。例如，一种流行的近似图同构的方法是检查WL算法经过K回合后，两个图是否具有相同的标签集，并且已知该方法可以解决大量图的同构问题[Shervashidze等人，2011]。

**graphlet及基于路径的方法**

最后，就像我们在讨论节点级特征时一样，一种在图上定义特征的有效且强大的方法是简单地计算不同graphlet结构出现的概率。在形式上，graphlet内核涉及枚举特定大小的所有可能的图形结构，并计算它们在整个图形中出现的次数。这种方法所面临的挑战是，尽管已经提出了许多近似方法，但对这些图进行计数是一个组合困难的问题[Shervashidze and Borgwardt，2009]。

枚举所有可能的graphlet的替代方法是使用基于路径的方法。在这些方法中，不是枚举graphlet数，而是简单地检查图中出现的各种路径。例如，Kashima等人提出的随机游走内核。[2003]涉及对图进行随机游走，然后计算不同度数序列的发生，而Borgwardt和Kriegel [2005]的最短路径内核也有类似的想法，但仅使用节点之间的最短路径（而不是随机游走）。正如我们将在本书的第3章中看到的那样，这种基于游走和路径表征图形的想法是一种强大的方法，因为它可以提取丰富的结构信息，同时又避免了图形数据的许多组合的陷阱。

### 2.2邻域重叠检测

在上一节中，我们介绍了关于节点或图的各种提取特征或统计信息的方法。这些节点和图的统计数据对于许多分类任务很有用。但是，它们的局限性在于它们不能量化节点之间的关系。例如上一节讨论的统计数据对于关系预测的任务不是很有用，我们的目标是预测两个节点之间是否存在边（图2.3）。

在本节中，我们将考虑成对节点之间关于邻域重叠的各种统计度量，从而量化了成对节点之间关系的紧密程度。例如，最简单的邻域重叠度量是计算两个节点共享的邻居节点数：

(2.7)

我们用来表示量化节点和之间关系的值,且令表示相似矩阵，它汇总了所有成对节点统计信息。

即使本节中讨论的任何统计量都没有涉及“机器学习”，但是对于关系预测，它们仍然是非常有用且功能强大的基准。给定邻域重叠统计量，常见的策略是假设边()存在的概率与成比例：

(2.8)

因此，为了能使用邻域重叠度量来处理关系预测任务，一个简单的做法是设置一个阈值以确定何时预测边存在的可能性。需要注意的是在关系预测设置中 通常我们假设只知道边集合的子集--真实边。我们希望的是在计算边的训练数据集上,节点与节点的相似性度量能准确预测关于边的测试数据集的存在性（图2.3）。

#### 2.2.1 局部重叠度量

局部重叠统计的是两个节点共享的邻接点的数量,即。例如，Sorensen指数定义了一个矩阵，它表示整个图中节点与节点邻域重叠情况，矩阵元素由以下公式给出：

(2.9)

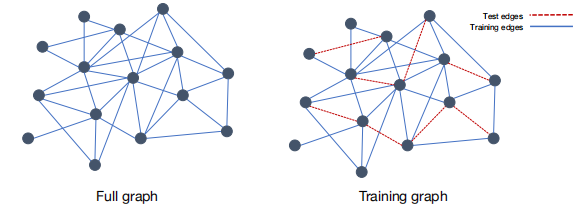


图2.3：关于完整图和用于训练的子采样图的图示说明。当训练一个模型或计算重叠统计数据时，训练图中的虚线边会被移除。基于预测这些保留的边的测试数据集的能力以评估模型。

它通过对节点度的总和来规范公共邻居节点的数量。某种形式的规范化通常非常重要；否则，对于那些度较多的节点预测边数据将会由较大的偏差。其他类似的方法包括Salton指数，该指数通过u和v的乘积进行规范化，即:

(2.10)

以及Jaccard重叠：

(2.11)

通常，这些措施试图量化节点邻域之间的重叠，同时最小化那些由节点度引起的任何偏差。在文献[Lüuand Zhou，2011]中还有很多这种方法动态说明。

除了简单地计算共享邻居节点的数量外，这些度量方法试图以某种方式来说明共享邻居节点的重要性。资源分配（RA）索引计算共享邻居节点数的倒数,

(2.12)

然而Adamic-Adar（AA）索引使用了度的对数的倒数进行类似的计算：

(2.13)

这两种度量方法都给那些在共享邻居节点中拥有较低度数的节点以更高的权重凭直觉，共享的低度邻居节点比共享的高度邻居节点能提供更多信息。

#### 2.2.2 全局重叠度

局部重叠度量对于链路预测是非常有效的启发式方法，即使与先进的深度学习方法相比，也常常有不错的性能。但是，局部方法的限制在于他们只考虑局部节点邻域。例如，两个节点在邻域内可能不存在局部重叠，但仍然属于图中的同一社区。全局重叠统计试图将这种关系考虑进来。

**卡茨指数**

Katz index是最基本的全局重叠统计数据，为了计算Katz索引，我们只需统计一对节点之间所有长度的路径数：

(2.14)

其中β∈R+是用户定义的参数，用于控制对短路径与长路径赋予多少权重。 一个小的β<1会降低长路径的重要性。

**矩阵的几何级数** Katzindex是矩阵的几何级数的一个示例，其变体在图分析和图表示学习中经常出现。下面的定理给出了矩阵的基本几何级数的解：

**定理1**.令为实值方阵，令表示的最大特征值。

当且仅当并且是非奇异矩阵时。

证明:令那么我们有

和

如果我们有所以:

基于定理1，我们可以看到给出了Katz index的解通过

(2.15)

其中是节点-节点相似性值的完整矩阵。

Leicht，Holme和Newman（LHN）相似

Katz index的一个问题是它严重依赖于节点的度数。当与较低度数节点相比时，较高度数节点在公式（2.14）中通常会给出更高的整体相似性评分，因为较高度数节点通常会涉及更多路径。为了减少这种情况，Leicht提出了一种改进方案，即通过考虑实际两个节点之间的实际观察路径数和预期路径数的比率：

(2.16)

根据我们对随机模型下期望的路径数期望（），对两个节点之间的路径数进行规范化。

为了计算期望，我们依靠所谓的配置模型，假设我们绘制具有相同度数集的随机图作为我们给定的图。在此假设下，我们可以分析计算

(2.17)

我们使用的地方表示图中的边总数。公式(2.17)指出，在随机配置模型下，可能性 边缘的角度与两个节点度的乘积成正比。

可以注意到，有边离开，并且这些边中的每一个有可能以结尾。对于我们可以类似地计算

(2.18)

这是因为长度为2的路径可以通过任何中间顶点，并且这种路径的可能性与可能性成正比,给定一个离开到达给定 乘以边离开到达的概率给定 (我们减去一个，因为我们已经用完了的边之一作为的传入边)

不幸的是，当我们超越路径时，随机配置模型下期望节点路径计数的解析计算就变得棘手长度3。因此，获得期望以获得更长的路径长度（即），Leicht等。 [2006]依靠最大特征值可以是用于估算路径数量的增长。特别是如果我们定义作为计算长度之间的长度为的路径的向量节点和所有其他节点，那么我们就拥有

(2.19)

因为最终将收敛到图的主导特征向量。这个表示两个节点之间的路径数在处增长在每次迭代中，我们记得是的最大特征值。对于大的近似值以及对 = 1的精确解，我们得到：

(2.20)

最后，将它们放在一起，我们可以获得规范化的 Katz指数，我们称为LNH指数（基于作者的姓名缩写 谁提出了算法）：最后，将它们放在一起，我们可以获得规范化的 Katz指数，我们称为LNH指数（基于作者的姓名缩写 谁提出了算法）：

(2.21)

是以一致的方式索引为的恒等矩阵与Katz索引不同，LNH索引占预期路径数节点之间只有在两个节点上比我们预期更多的路径。使用定理1求解矩阵序列（忽略对角线术语后）可以写成如下公式：

(2.22)  
其中是在对角线上具有节点度的矩阵。  
**随机游走法**

另一组全局相似性度量考虑随机游走而不是图上路径的确切计数。例如，我们可以直接应用 PageRank方法的一种变体[Page等，1999] 4-被称为 个性化PageRank算法我们定义了随机矩阵并计算：

(2.23)

在这个等式中，是节点的一个单指标向量，而给出从节点开始的随机游走访问节点的平稳概率。在此，项确定随机游走在以下位置重新开始的概率每个时间步的节点。没有这种重启概率，随机游走概率将简单地收敛到特征向量的归一化变量中心性。但是，有了这种重启概率，我们反而获得了一个量度特定于节点的重要性，因为随机游走是连续不断的被“传送”回该节点。给出了这种复发的解决方案通过

(2.24)

我们可以定义一个节点-节点随机游走相似性度量为

(2.25)

即,一对节点之间的相似性与我们的可能性成正比 以从另一个节点开始的随机游走到达每个节点。

### 2.3图拉普拉斯和频谱方法

已经讨论了使用图数据进行分类的传统方法以及关系预测的传统方法,我们现在将问题转向学习在图上聚类节点。这部分将开始低维度节点嵌入的学习任务。我们首先定义一些可用于表示图形的重要矩阵，并简要介绍频谱图理论的基础。

#### 2.3.1图拉普拉斯算子

邻接矩阵可以表示图而不会丢失任何信息。 但是，存在具有有用的数学属性的图形的替代矩阵表示。 这些矩阵表示形式称为拉普拉斯算子，由邻接矩阵的各种变换形成。

**未规范化的拉普拉斯算子**

最基本的拉普拉斯矩阵是未规范化的拉普拉斯矩阵，定义为如下：

(2.26)

其中是邻接矩阵，是度矩阵。拉普拉斯简单图的矩阵具有许多重要属性：

1.它是对称的并且是半正定的。

2.以下向量恒等式为

(2.27)

(2.28)

3.L有非负特征值：

拉普拉斯算子和连接的组件拉普拉斯算子总结了图的许多重要属性。例如，我们有遵循定理：

定理2。拉普拉斯算子的特征值0的几何多重性对应于图中连接分量的数量。 证明。可以注意到，对于任何特征值0的特征向量，我们都有

(2.29)

通过特征值-特征向量方程的定义。而且，结果 公式(2.29)表示

(2.30)

那么上面的等式意味着，其中反过来，对于在其中的所有节点，都是相同的常数。相同的连接组件。因此，如果图形完全连接，则特征值0的特征向量将是所有图中的节点，这将是特征值0的唯一特征向量，因为在这种情况下，方程（2.29）只有一个唯一的解决方案。

相反，如果图由多个连接的组件组成 那么我们将得到方程2.29在每个 拉普拉斯算子中与每个连接的组件相对应的块。那是，如果图由个连通的分量组成，则图中的节点存在顺序，这样拉普拉斯矩阵可以被写成

 (2.31)

其中该矩阵中的每个块都是有效的图拉普拉斯算子原始图的完全连接的子图。由于它们是有效的 完全连通图的拉普拉斯算子，对于每个块方程（2.29）成立，并且每个子拉普拉斯主义者都有一个 特征值0乘以1且特征向量均为1（已定义 仅在该组件中的节点上）。而且，由于是一个块对角矩阵，其光谱由所有个块，即的特征值是的特征值的并集矩阵和的特征向量是的特征向量的并集，所有矩阵的0值填充在其他块的位置。 因此，我们可以看到每个块为特征值贡献一个特征向量 0，该特征向量是该连接节点的指标向量零件。  
**规范化的拉普拉斯**

除了未归一化的拉普拉斯算子外，还有两种流行的归一化 拉普拉斯算子的变体。对称归一化拉普拉斯算子定义为

(2.32)

而随机游走拉普拉斯算子被定义为

(2.33)

这两个矩阵都具有与拉普拉斯算子相似的性质，但是由于归一化，它们的代数性质相差很小的常数。例如，定理2完全适用于。对于，定理2成立，但具有缩放的0特征值的特征向量。正如我们将看到的本书中，拉普拉斯算子的这些不同变体对于分析和学习任务。

#### 2.3.2图切割和聚类

在定理2中，我们看到对应于0的特征值的特征向量拉普拉斯算子可用于根据连接的节点将节点分配给群集它们所属的组件。但是，这种方法仅允许我们对已经在断开连接的组件中的节点进行集群，这很简单。在这个部分，我们将这一想法更进一步，表明拉普拉斯算子可以用于在完全连接的图中给出节点的最佳群集。

**图切割**  
为了激发拉普拉斯光谱聚类方法，我们首先必须定义最佳聚类的含义。为此，我们定义了 在图上切。令表示图中节点的子集，令表示该集合的补数，即。给定一个我们将图划分为个不重叠的子集该分区的切割值为

(2.34)  
换句话说，切割仅是越过边界的边数的计数 节点分区之间。现在，一个定义最佳聚类的选项 到个群集中的节点的选择是选择一个分区，以最小化 削减价值。有有效的算法可以解决此任务，但存在一个已知问题 使用这种方法的原因是它倾向于简单地使包含单个 节点[Stoer and Wagner，1997]。 因此，我们通常不力求最小化切割，而不仅仅是简单地减少切割。 削减，同时还要求所有分区都相当大。之一 强制执行此操作的常用方法是最小化“比率削减”：

 (2.35)  
这不利于选择较小群集大小的解决方案。另一个受欢迎 解决方案是最小化标准化切割（NCut）：

(2.36)  
其中。强制所有集群具有相似的 入射到其节点的边数。

**用Laplacian光谱逼近RatioCut**

现在，我们将得出一种寻找最小化集群分配的方法使用Laplacian光谱的RatioCut。 （可以使用类似的方法尽量减少NCut值。）为简单起见，我们将考虑这种情况其中 = 2，并且我们将节点分为两个集群。我们的目标是解决以下优化问题

(2.37)

为了更方便地重写此问题，我们定义以下向量

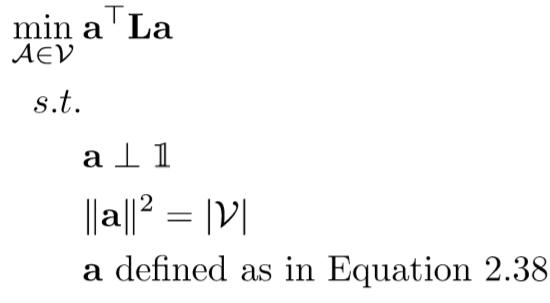
(2.38)

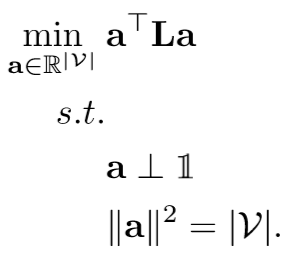
将此向量与我们的拉普拉斯图的属性结合起来，我们可以看到

因此，我们可以看到a允许我们用拉普拉斯算子来写比例削减 （不超过一个常数）。此外，a还有两个其他重要属性：

(2.45)

(2.46)  
其中1是所有向量的向量。 将所有这些放在一起，我们可以重写比率削减最小化问题 在公式（2.37）中为

  
但是不幸的是，这是一个NP难题，因为的定义如公式2.38所示，要求我们对离散量进行优化组。显而易见的放松是消除了这种离散性条件，并将最小化简化为超过实值向量：



根据Rayleigh-Ritz定理，此优化问题的解决方案是由的第二个最小特征向量给定（因为最小特征向量为等于1）。因此，我们可以通过将设置为来近似化RatioCut的最小化是拉普拉斯算子的第二小特征向量。要转实值向量变成一组离散的集群分配，我们可以简单地将节点分配给根据的符号进行聚类，即

(2.49)

总之，拉普拉斯算子的第二小特征向量是连续的 离散向量的近似，给出最佳的聚类分配（相对于RatioCut）。可以显示出近似的结果以近似NCut值，但是它依赖于第二个最小的特征向量归一化的拉普拉斯LRW [Von Luxburg，2007]。

#### 2.3.3广义谱聚类

在上一节中，我们看到了拉普拉斯算子的频谱，使我们能够将图有意义地划分为两个簇。 特别是，我们看到第二小的特征向量可用于将节点划分为不同的簇。 通过检查拉普拉斯算子的K个最小特征向量，可以将此一般化思想扩展到任意数量的K个簇。 此一般方法的步骤如下：

1.找到的个最小特征向量（不包括最小特征向量）：。

2.形成矩阵，其中步骤1的特征向量为 列。

3.用矩阵中的对应行表示每个节点，即

4.对嵌入进行均值聚类。

与上一节中对 = 2情况的讨论一样，这种方法 可以适应使用归一化的Laplacian，并且近似结果 对于 = 2，也可以推广到 > 2的情况[Von Luxburg，2007]。 频谱聚类的一般原理是有力的。我们可以使用图拉普拉斯算子的频谱来表示图中的节点 表示可以作为原则上的最佳近似 图聚类。光谱之间也存在紧密的理论联系 图上的聚类和随机游走，以及图信号处理领域Ortega等人。 [2018]。以后我们将讨论许多这样的联系章节。

### 2.4走向学习的表示

在前面的部分中，我们看到了许多传统的学习方法 在图上。我们讨论了图统计和内核如何提取特征 分类任务的信息。我们看到了邻域重叠统计如何为关系预测提供强大的启发式方法。而且，我们提供了 频谱聚类的概念的简要介绍，它使我们能够以有原则的方式将节点聚类为社区。但是，方法 本章讨论的内容，尤其是节点和图级统计信息 由于它们需要仔细的手工统计数据而受到限制 和措施。这些手工设计的功能不灵活-即它们无法通过学习过程进行调整-设计这些功能可以 耗时且昂贵的过程。本书的以下各章 介绍学习图的替代方法：图表示 学习。我们将寻求学习而不是提取手工设计的功能 表示有关图形的结构信息的表示形式。

**Part1 节点嵌入**

**第三章 邻里重建方法**

这本书的这一部分与学习节点嵌入的方法有关。这些方法的目标是将节点编码为低维向量，总结其图的位置及其局部图邻域的结构。换句话说，我们希望将节点投影到一个潜在空间中，这个潜在空间中的几何关系对应于原始图或网络中的关系(如边)[等人，2002](图3.1)。

在本章中，我们将提供简单和加权图的节点嵌入方法的概述。第4章将提供对多关系图的类似嵌入方法的概述。

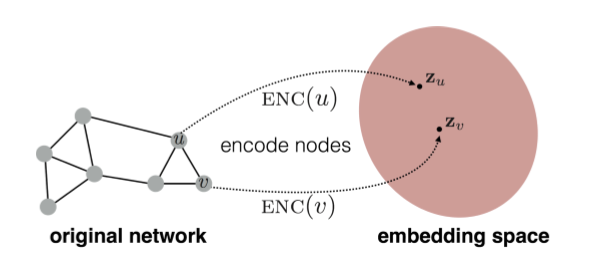


图3.1：节点嵌入问题的图示。我们的目标是学习一种将节点映射到低维嵌入空间的编码器()。对这些嵌入进行了优化，使嵌入空间中的距离反映原始图中节点的相对位置。

**3.1编解码器视角**

我们基于编码和解码图的框架来组织节点嵌入的讨论。这种观察图形表示学习的方式将在全书中反复出现，并且基于汉密尔顿等人的观点，我们基于这种观点的节点嵌入的方法的介绍非常的紧密 [2017a]。

在编解码器框架中，我们认为图形表示学习问题涉及两个关键操作。首先，编码器模型将图中的每个节点映射为低维向量或嵌入。接下来，解码器模型采用低维节点嵌入并使用它们重构原始图中每个节点的邻域信息。图3.2总结了这一思想。

**3.1.1编码器**

形式上，编码器是将节点映射到矢量（其中对应于节点的嵌入）的函数，在最简单的情况下，编码器具有以下的签名：

：，（3.1）

表示编码器将节点作为输入以生成节点嵌入。在大多数有关节点嵌入的工作中，编码器依赖于我们所说的浅层嵌入方法，其中，该编码器功能只是基于节点的嵌入查找。换句话说，我们有

，（3.2）

其中是一个矩阵，其中包含所有节点的嵌入矢量，表示节点对应的行。

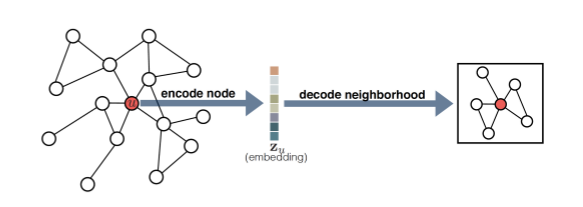
浅层嵌入方法将是本章的重点。然而，我们注意到编码器也可以推广到浅层嵌入方法之外。例如，编码器可以使用节点特征或每个节点周围的局部图结构作为输入来生成嵌入。这些广义编码器架构（通常称为图形神经网络()）将是本书第二部分的主要重点。

图3.2:编码器-解码器方法的概述。编码器将节点映射到一个低维嵌入。译码器利用重构的局部邻域信息。

**3.1.2解码器**

解码器的作用是从编码器生成的节点嵌入中重构特定的图形统计信息。例如，给定一个节点嵌入一个节点的，解码器可能试图预测的邻接集合或它在图邻接矩阵中的行。

虽然可能有很多解码器，但标准做法是定义成对解码器，其签名如下:

，（3.3）

成对解码器可以解释为预测节点对之间的关系或相似性。例如，一个简单的成对解码器可以预测图中的两个节点是否相邻。

将成对解码器应用于一对嵌入)可以重建节点和之间的关系。目标是优化编码器和解码器以最小化重建损失，从而

)≈[. （3.4）

这里，我们假设是节点间基于图的相似性度量。例如，预测两个节点是否是邻居的简单重构目标对应的是和。然而，我们也可以用更一般的方式定义，例如利用2.2节中讨论的任意成对邻域重叠统计量。

**3.1.3优化编解码器模型**

为了实现重构目标(式3.4)，标准做法是在一组训练节点对上最小化经验重构损失:

，（3.5）

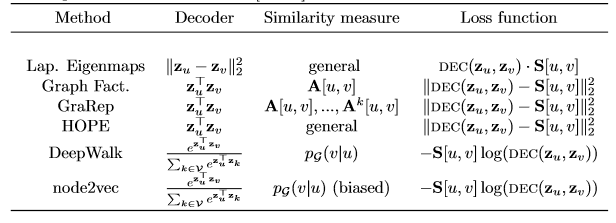
式中是度量译码(即估计)相似度值与真实值之间差异的损失函数。根据解码器和相似函数的定义，损失函数可能是一个均方误差，甚至是分类损失，如交叉熵。因此,总体目标是训练编码器和译码器,以便在训练集上有效的重建成对的节点关系。大多数方法使用随机梯度下降来最小化公式3.5中的损耗[,1951],但在某些情况下,可以使用更专业的优化方法(例如,基于矩阵分解)。

**3.1.4编解码器方法概述**

表3.1从编码器-解码器的角度总结了几种已知的节点嵌入方法，所有这些方法均使用浅层编码方法。编码器-解码器框架的主要优点是，它可以使人们基于(i)解码器功能，(ii)基于图形的相似性度量，(iii)损失函数来简洁地定义和比较不同的嵌入方法。

在接下来的章节中，我们将更详细地描述表3.1中代表性的节点嵌入方法。我们将首先讨论由矩阵分解方法激发的节点嵌入方法(第3.2节)推动并与频谱聚类有紧密理论联系的节点嵌入方法(请参阅第1章)。接下来，我们将讨论基于随机游走的最新方法(第3.3节)。这些随机游走方法最初是受自然语言处理的启发而产生的，但是，正如我们将要讨论的那样，它们也与频谱图理论有着密切的理论联系。

表3.1一些著名的浅层嵌入算法的摘要。请注意，基于随机游走方法的解码器和相似函数是不对称的，相似函数对应于从开始的定长随机游走访问的概率。改编自等人[2017]。



**3.2基于分解的方法**

从矩阵分解的角度来看编解码器思想的一种方法。实际上，从节点的嵌入中解码局部邻域结构的挑战与重构图邻接矩阵中的条目密切相关。更一般地说，我们可以将此任务视为使用矩阵分解来学习节点-节点相似度矩阵的低维近似，其中推广了邻接矩阵并捕获了一些用户定义的节点-节点相似度概念(如3.1.2节所讨论的)[]。

**拉普拉斯特征图** 特征图()技术是最早也是最具影响力的基于因子分解的方法之一，它建立在第2章[]中讨论的光谱聚类思想之上。在这种方法中，我们根据节点嵌入之间的距离来定义解码器:

然后，损失函数根据图中的节点权重对她们进行加权：

， （3.6）

这种方法背后的直觉是，当非常相似的节点有距离很远的嵌入时，公式(3.6)会对模型进行惩罚。如果构造使其满足矩阵的性质，则式(3.6)中使损失最小化的节点嵌入与我们在2.3节中讨论的光谱聚类的解相同。特别地，如果我们假设嵌入是维的，那么最小化方程(3.6)的最优解由拉普拉斯方程的个最小特征向量给出(不包括所有特征向量)。

**内积方法** 根据拉普拉斯特征图技术，最近的工作通常采用基于内积的解码器，定义如下：

, （3.7）

在这里，我们假设两个节点之间的相似度（例如，它们的局部邻域之间的重叠）与它们的嵌入点积成比例。

这种类型的节点嵌入算法的示例包括图因子分解（）方法1[等，2013]， [等，2015]和 [等，2016]。这三种方法都将内积解码器（公式3.7）与以下均方方差结合在一起：

（3.8）

它们主要区别在于它们如何定义，即它们使用的节点-节点相似性或邻域重叠的概念。方法直接使用邻接矩阵并设置，，而和方法则采用更通用的策略。尤其是，根据邻接矩阵幂定义，而算法则支持一般的邻域重叠度量（例如，第2.2节中的任何邻域重叠度量）。

这些方法称为矩阵分解方法，因为可以使用分解算法（例如奇异值分解（））将其损失函数降至最低。 实际上，通过将节点嵌入堆叠到矩阵中，这些方法的重建目标可以写为

（3.9）

直观上讲，这些方法的目标是为每个节点学习嵌入，以使学习的嵌入矢量之间的内积近似于节点相似性的确定性度量。

**3.3随机游走嵌入**

上一节中讨论的内积方法都使用确定性的节点相似性度量。 它们通常将定义为邻接矩阵的多项式函数，并且优化节点嵌入，以便。 在这些成功的基础上，近年来，成功的方法不断涌现，这些方法使内部产品方法适用于使用邻域重叠的随机度量。 这些方法的关键创新在于优化了节点嵌入，因此，如果两个节点倾向于在图上进行短暂的随机游走，则两个节点具有相似的嵌入。

**和** 与上述矩阵分解方法相似，和使用浅层嵌入方法和内部乘积解码器。 这些方法的主要区别在于它们如何定义节点相似性和邻域重构的概念。 这些方法不是直接重建邻接矩阵或的某些确定性函数，而是优化嵌入以对随机游走的统计信息进行编码。 从数学上讲，目标是学习嵌入，从而使以下内容（大致）成立：

（3.10）

其中，是从开始在长度为的随机游历中访问的概率，通常将定义为{2，...，10}。 再次，等式（3.10）与基于因子分解的方法（例如，等式3.8）之间的关键区别在于，等式（3.10）中的相似性度量既是随机的又是非对称的。

为了训练随机游动嵌入，通常的策略是使用公式（3.10）中的解码器，并最小化以下交叉熵损失：

（3.11）

在这里。我们用表示随机游动的训练集，它是从每个节点开始抽样随机游动产生的。例如，我们可以假设每个节点有对共发生节点，从分布 ~，中采样。

但是，不幸的是，天真地评估方程式（3.11）中的损耗在计算上可能会很昂贵。 实际上，仅在方程式（3.10）中评估分母具有时间复杂度，这使得评估损失函数的总体时间复杂度为。 有多种策略可以克服此计算难题，这是原始算法和算法之间的本质区别之一。 使用分层近似方程式（3.10），其中涉及利用二叉树结构来加快计算速度[等人，2014]。 另一方面采用噪声对比方法来近似方程式（3.11），其中归一化因子使用负样本以以下方式近似[，2016]：

（3.12）

在这里，我们使用表示逻辑函数，使用表示节点集合上的分布，并且我们假设> 0是超参数。 在实践中，通常被定义为均匀分布，并且期望值是使用蒙特卡洛采样法近似的。

通过允许更加灵活地定义随机游走，方法也与较早的算法有所不同。 特别是，尽管仅采用统一的随机游动来定义，，而方法引入了超参数，这些参数允许随机游动概率在更类似于广度搜索或深度优先的游动之间平滑地插值 在图上搜索。

**大规模信息网络嵌入（）** 等人除了和外。 [2015]的算法通常在随机游走方法的背景下进行讨论。 方法未明确利用随机游走，但与和共享概念动机。 中的基本思想是将两个编码器/解码器目标组合在一起。 第一个目标旨在对一阶邻接信息进行编码，并使用以下解码器：

（3.13）

具有基于邻接关系的相似性度量（即）。 第二个目标与随机游走方法更相似。 它是与公式（3.10）相同的解码器，但使用KL散度对其进行了训练，以对两跳邻接信息（即中的信息）进行编码。 因此，在概念上与和有关。 它使用概率解码器和概率损失函数（基于散度）。 但是，它没有对随机游走进行采样，而是显式地重建了第一和第二阶邻域信息。

**随机游走想法的其他变体** 随机游走方法的一个好处是可以通过对随机游走进行偏置或修改来扩展和修改它。 例如，等。 [2016]考虑随机游走“跳过”节点，这会产生类似于（在3.2节中讨论）和等人的相似性度量。 [2017]根据节点之间的结构关系（而不是邻域信息）定义随机游走，它会生成节点嵌入，这些节点嵌入对图形中的结构角色进行编码。

**3.3.1随机游动方法和矩阵分解**

可以证明，随机游走方法实际上与矩阵分解方法密切相关[。，2018]。 假设我们定义以下节点-节点相似度值矩阵：

（3.14）

其中是常数，。 在这种情况下，邱等人。 [2018]显示，学习的嵌入满足：

（3.15）

有趣的是，我们还可以将等式（3.14）的内部分解为：

（3.16）

其中是对称归一化拉普拉斯算子的本征分解。 这表明学习的嵌入实际上与本书第一部分中讨论的频谱聚类嵌入紧密相关。 关键区别在于，嵌入通过来控制不同特征值的影响，即随机游走的长度。 邱等。 [2018]得出了与矩阵分解相关的相似连接，并讨论了受此连接启发的其他相关的基于分解的方法。

**3.4浅层嵌入的局限性**

本章的重点（以及本书的更一般的部分）一直在浅层嵌入方法上。 在这些方法中，将节点映射到嵌入的编码器模型只是一个嵌入查找（公式3.2），它为图中的每个节点训练唯一的嵌入。 在过去的十年中，这种方法取得了许多成功，在下一章中，我们将讨论如何将这种浅层方法推广到多关系图。 但是，还必须注意，浅层嵌入方法具有一些重要的缺点：

1.第一个问题是，浅层嵌入方法在编码器的节点之间不共享任何参数，因为编码器直接为每个节点优化了唯一的嵌入向量。 缺乏参数共享在统计和计算上都是无效的。 从统计的角度来看，参数共享可以提高学习效率，并且还可以作为有力的正则化形式。 从计算的角度来看，缺少参数共享意味着浅层嵌入方法中的参数数量必定会随着的增长而增加，这在大规模图中是很难处理的。

2.浅层嵌入方法的第二个关键问题是它们不利用编码器中的节点特征。 许多图形数据集都具有丰富的特征信息，这些特征信息可能在编码过程中提供信息。

3.最后，也许是最重要的是，浅层嵌入方法本质上是转导性的[.，2017]。 这些方法只能为训练阶段存在的节点生成嵌入。 除非执行其他优化来学习这些节点的嵌入，否则不可能在训练阶段之后为新节点生成嵌入。 此限制阻止了浅层嵌入方法用于归纳应用程序，该方法涉及在训练后推广到看不见的节点。

为了减轻这些限制，可以使用更复杂的编码器来替换浅层编码器，而这些编码器通常更依赖于图形的结构和属性。 在本书的第二部分中，我们将讨论定义此类编码器的最流行的范例，即图神经网络（）。

**第四章多关系数据和知识图**

在第3章中，我们讨论了用于学习节点的低维嵌入的方法。 我们专注于所谓的浅层嵌入方法，其中我们为每个节点学习了唯一的嵌入。 在本章中，我们将继续专注于浅层嵌入方法，并将介绍用于处理多关系图的技术。

知识图完成我们在本章中回顾的大多数方法最初都是为知识图完成而设计的。 在知识图完成中，我们得到一个多重关系图，其中边定义为元组，指示在两个之间存在特定关系 节点。 这样的多关系图通常称为知识图，因为我们可以将元组解释为指定在两个节点和之间存在特定的“事实”。例如，在生物医学知识中图中我们可能有一个边缘类型，并且边缘可能表明与结点相关的药物可以治疗与结点相关的疾病。通常，知识图完成的目标是预测结点中缺失的边 图，即关系预测，但也有使用多关系图的节点分类任务的示例[等，2017]。

在本章中，我们将简要概述多关系图的嵌入方法，但是必须注意，对知识图完成的全面处理不在本章范围之内。 并非所有的知识图完成方法都依赖于嵌入，因此在此我们不会涵盖嵌入方法的所有变体。 我们将感兴趣的读者推荐给等。 [2016]对该领域进行补充审查。

**4.1重建多关系数据**

与第3章中讨论的简单图一样，我们可以将嵌入多关系图视为重构任务。 给定两个节点的嵌入和，我们的目标是重建这些节点之间的关系。 与上一章的设置相比，复杂性在于我们现在必须处理多种不同类型的边缘。

为了解决这种复杂性，我们增加了解码器以使其具有多重关系。 现在，我们不仅将一对节点嵌入作为输入，还定义了解码器接受一对节点嵌入以及一个关系类型，即。 我们可以将解码器的输出即解释为图中边缘存在的可能性。

举一个具体的例子，一种学习多关系嵌入的最简单，最早的方法（通常称为）将解码器定义为[等，2011]：

（4.1）

其中是关系的可学习矩阵。使用此解码器可以使事情变得简单，我们可以使用基本重建损失来训练嵌入矩阵和关系矩阵，：

（4.2） （4.3）

其中是多关系图的邻接张量。如果我们要优化方程式（4.2），实际上我们将执行一种张量分解。因此，将张量分解的想法概括了第3章中讨论的矩阵分解方法。

损失函数，解码器和相似性函数在第3章中，我们讨论了节点嵌入方法的多样性如何主要源于使用不同的解码器（），相似性度量（）和损失函数（） 。解码器给出一对节点嵌入之间的分数；相似度函数定义我们试图解码的节点-节点相似度；损失函数告诉我们如何评估解码器的输出与地面真实度相似性度量之间的差异。在多关系设置中，我们还将看到解码器和损失函数的多样性。然而，几乎所有的多关系嵌入方法都只是直接基于邻接张量来定义相似性度量。换句话说，本章中的所有方法都假定我们试图从低维嵌入中重建直接（多关系）邻居。这是由于难以在多关系图中定义高阶邻域关系，以及大多数多关系嵌入方法是专门为关系预测而设计的事实。

**4.2损失函数**

如上所述，用于多关系节点嵌入方法的两个关键要素是解码器和损失函数。 我们首先简要讨论用于此任务的标准损耗函数，然后再将注意力转移到文献中已提出的众多解码器上。作为我们考虑的损失函数的动力，值得考虑的是我们在公式（4.2）中引入的简单重建损失的弊端。 这种损失有两个主要问题。 第一个问题是计算成本非常高。 公式（4.2）中的嵌套和要求进行运算，并且此计算时间在很多大图中都是禁止的。 此外，由于许多多重关系图都是稀疏的，即—理想情况下，我们希望损失函数为。 第二个问题更加微妙。 我们的目标是从低维节点嵌入中解码邻接张量。 我们知道（在大多数情况下）该张量将仅包含二进制值，但是等式（4.2）中的均方误差不适合这种二进制比较。 实际上，均方误差是回归的自然损失，而我们的目标更接近边缘分类。

**负熵的交叉熵**

一种既有效又适合我们的任务的流行损失函数是负采样的交叉熵损失。 我们将这种损失定义为：

（4.4）

其中表示对数函数，表示节点集合（可能取决于）上的“负采样”分布，并且> 0是超参数。 这与我们在（公式3.12）中看到的损耗基本相同，但是在这里我们考虑的是通用的多关系解码器。

我们称其为交叉熵损失，因为它是从标准二进制交叉熵损失中得出的。 由于我们将解码器的输出馈送到逻辑函数，因此我们可以在[0,1]中获得归一化的分数，可以将其解释为概率。 期限

(4.5)

然后等于对图上实际存在的边预测的对数似然。 另一方面，术语

(4.6)

然后等于期望的对数似然性，我们可以正确预测图中不存在的边的“假”。 实际上，使用蒙特卡洛近似法对期望值进行评估，而这种损失的最流行形式是

(4.7)

其中是从采样的一组节点（通常很小）。

**有关负采样的注意事项**

生成负采样的方式可能会对所学习嵌入的质量产生重大影响。 定义分布的最常用方法是简单地在图形的所有节点上使用均匀分布。 这是一个简单的策略，但是这也意味着我们在交叉熵计算中会得到“假阴性”。 换句话说，我们可能会意外地对图中实际存在的“负”元组进行采样。 一些作品通过过滤这种假阴性来解决这个问题。

否定采样的其他变体尝试生成更多的“困难”否定采样。 例如，某些关系只能存在于某些类型的节点之间。 （在知识图中表示一个人的节点不太可能涉及涉及关系的边）。 因此，一种策略是仅对满足此类类型约束条件的否定示例进行抽样。等。 [2019]甚至提出了一种通过学习对抗模型来选择具有挑战性的负样本的方法。

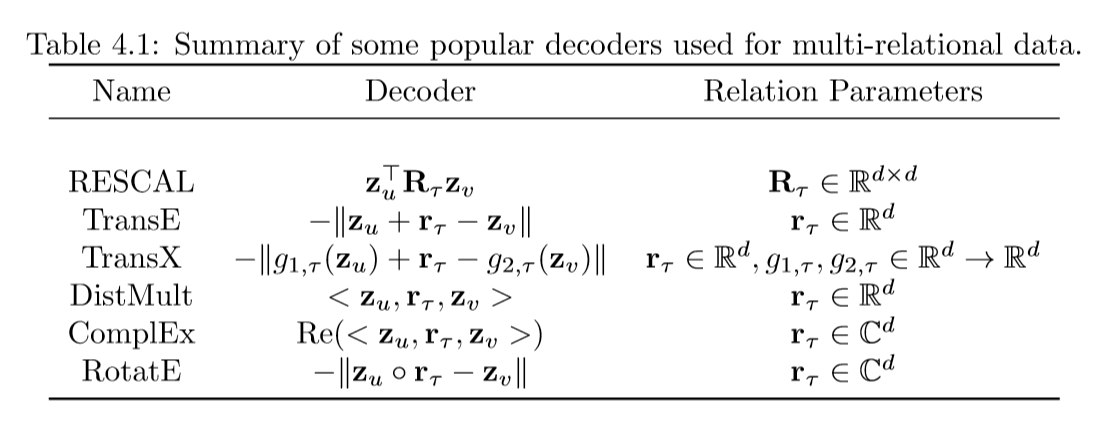
还要注意，在不失一般性的前提下，我们假设负采样发生在边缘元组的第二个节点上。 也就是说，我们假设通过用负样本替换元组中的尾节点来绘制负样本。 总是对尾节点进行采样会简化符号，但是会导致边关系很重要的多关系图中的偏差。 实际上，最好为该关系的头节点（即）和尾节点（即）绘制负样本。

**最大保证金损失**

用于多关系节点嵌入的另一个流行的损失函数是余量损失：

(4.8)

在这种损失中，我们再次将真实对的解码得分与否定样本进行比较，这种策略通常称为对比估计。 但是，在方程（4.8）中，我们没有将其视为二进制分类任务，而只是比较了解码器的直接输出。 如果“真”对的得分大于“负”对的得分，那么我们损失很小。 项称为余量，如果所有示例的分数差异至少大到0，则损失等于0。 这种损失也称为铰链损失。



**4.3多关系解码器**

上一节介绍了用于学习多学习节点嵌入的两个最受欢迎的损失函数。 这些损失可以与各种不同的解码器功能结合在一起，现在我们将注意力转向这些解码器的定义。 到目前为止，我们仅讨论了第4.1节中介绍的一种可能的多关系解码器，即所谓的解码器：

(4.9)

在解码器中，我们将可训练矩阵与每个关系相关联。 但是，这种方法的局限性（以及它不经常使用的原因）是其表示关系的计算和统计成本很高。 中每种关系类型都有参数，这意味着与实体相比，关系需要更多数量级的参数来表示。

更流行的现代解码器旨在仅使用参数来表示每个关系。 尽管我们的调查远未详尽，但我们将在此处讨论多关系解码器的几种流行变体。 表4.1总结了本章介绍的解码器。

翻译解码器

一类流行的解码器将关系表示为嵌入空间中的转换。 这种方法是由等人发起的。 [2013]的模型，将解码器定义为

(4.10)

在这些方法中，我们使用维嵌入来表示每个关系。 在根据关系嵌入平移头节点之后，边缘的可能性与头节点的嵌入与尾节点的嵌入之间的距离成比例。 是提出的最早的多关系解码器之一，并且在许多应用中仍然是牢固的基准。

的一个局限性是它的简单性，然而，许多著作也提出了对该翻译思想的扩展。 我们将这些模型统称为模型，它们具有以下形式：

(4.11)

其中是依赖于关系的可训练变换。 例如，等。 [2014]的模型将解码器定义为:

(4.12)

方法在执行翻译之前，将实体嵌入投影到可学习的关系特定的超平面上（由法向矢量定义）。 等人提出了模型的其他变体。 [2016]和等。 [2015]。

多线性点积

第二流行的工作不是通过翻译嵌入来定义解码器，而是通过从简单图形中概括点积解码器来开发多关系解码器。 在这种方法（通常称为等人的和首次提出的方法）中，我们将解码器定义为:

(4.13) (4.14)

因此，这种方法直接定义了点积，可以在三个向量上进行定义。

**复合解码器**

等式（4.13）中的DistMult解码器的局限性在于它只能编码对称关系。 换句话说，对于等式（4.13）中定义的多线性点积解码器，我们有:

这是一个严重的限制，因为多关系图中的许多关系类型都是有向的且不对称的。 为了解决这个问题，等人。 [2016]提出通过采用复值嵌入来增强编码器。 他们将定义为：

（4.15） （4.16）

其中是复值嵌入，表示复数向量的实分量。 由于我们采用了尾部嵌入的复共轭，因此这种解码方法可以适应非对称关系。

一种称为的相关方法将解码器定义为复杂平面中的旋转，如下所示[等，2019]：

（4.17）

其中表示哈达玛积。 在公式4.17中，我们再次假设所有嵌入都是复数值，并且我们另外限制了的项，使得| | = 1，∈{1，...，}。 这种限制意味着关系嵌入的每个维度都可以表示为，因此对应于复平面中的旋转。

**4.3.1代表性能力**

表征各种多关系解码器的一种方法是根据它们表示关系上不同逻辑模式的能力。

对称和反对称例如，许多关系是对称的，这意味着：

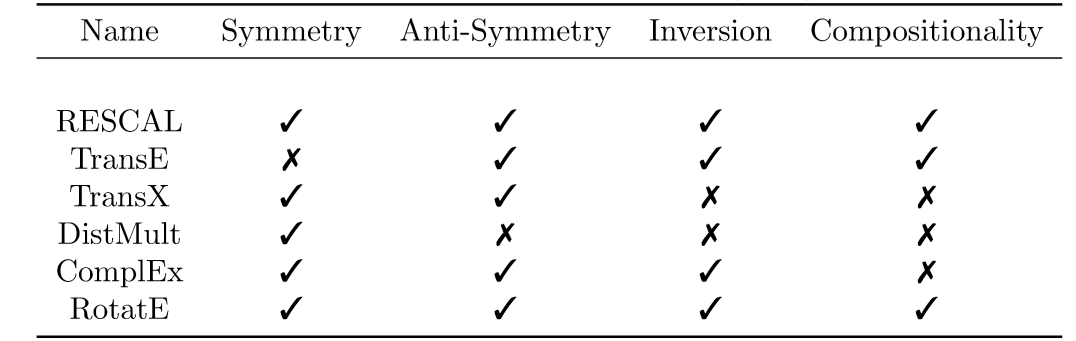
（4.18）

在其他情况下，我们明确具有满足以下条件的反对称关系：

（4.19）

一个重要的问题是，不同的解码器是否能够对对称和反对称关系进行建模。 例如，DistMult只能表示对称关系，因为

表4.2：一些流行的多关系解码器对关系模式进行编码的能力的摘要。 改编自等。 [2019]。



根据该方法的定义。 另一方面，TransE模型只能表示反对称关系，因为

反转与对称有关的是反转的概念，其中一个关系表示存在另一个方向相反的关系：

（4.20）

尽管DistMult同样无法对这种模式进行建模，但大多数解码器都能够表示逆关系。

最后，我们可以考虑解码器是否可以编码以下形式的关系表示之间的组成：

(4.21)

例如，在中，我们可以通过定义来解决这个问题。 我们可以通过定义类似地在中对组成进行建模。

通常，考虑这些类型的关系模式对于比较不同的多关系解码器的表示能力很有用。 实际上，我们可能不会期望这些模式完全正确，但是可能会有许多关系在某种程度上表现出这些模式（例如，对称关系> 90％的时间）。 表4.2总结了我们讨论的各种解码器对这些关系模式进行编码的能力。

**第五章** **图神经网络模型**

这本书的第一部分讨论了学习图中节点的低维嵌入的方法。我们讨论的节点嵌入方法使用浅层嵌入方法来生成节点的表示法，为每个节点优化了唯一的嵌入向量。在本章中，我们将重点放在更复杂的编码器模型上,并介绍图神经网络(GNN)的形式，它是定义图数据上的深度神经网络的一般框架。关键思想是，我们想要生成实际上依赖于图形结构的节点的表示，以及我们可能拥有的任何特征信息。

在开发复杂的图形结构数据编码器的主要挑战是，我们通常的深度学习工具箱不适用。例如，卷积神经网络(CNNs)只在网格结构的输入(如图像)上定义良好，而递归神经网络(RNNs)只在序列(如文本)上定义良好。要在一般图上定义深度神经网络，我们需要定义一种新的深度学习架构。

|  |
| --- |
| 置换不变性和等变性  在图上定义深度神经网络的一个合理的想法是简单地使用邻接矩阵作为深度神经网络的输入。例如，为了生成整个图的嵌入，我们可以简单地将邻接矩阵扁平化，然后将结果反馈给一个多层感知器(MLP):  (5.1)  其中表示邻接矩阵的一行，我们用⊕表示向量级联。  这种方法的问题是，它依赖于我们在邻接矩阵中使用的节点的任意顺序。换句话说，这样的模型不是置换不变的，设计图上神经网络的一个关键要求是它们应该置换不变(或等变)。在数学术语中，任何以邻接矩阵作为输入的函数理论上都应满足以下两个特性之一:  (置换不变性)  (置换等变性)  其中是一个置换矩阵。置换不变性意味着函数不依赖邻接矩阵中行/列的任意排序，而置换等变性意味着当我们对邻接矩阵进行置换时，的输出以一致的方式进行置换。(第一部分中讨论的浅编码器是置换等变函数的一个例子。)确保不变或等方差是我们在学习图时的一个关键挑战，我们将在后面的章节经常重新讨论置换等方差和不变性的问题。 |

**5.1神经信息传递**

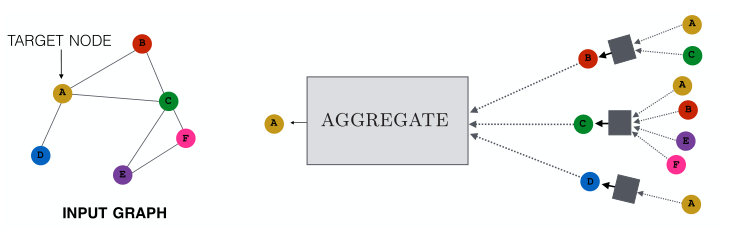
基本的图形神经网络(GNN)模型可以用多种方式激励。同样的基本GNN模型被导出为对非欧几里得数据的卷积推广[Bruna et al.， 2014]，作为信念传播的可区分变体[Dai et al.， 2016]，并类推到经典的图同构检验[Hamilton et al.， 2017b]。不管动机如何，GNN的定义特征是它使用一种神经信息传递形式，其中向量信息在节点之间交换，并使用神经网络进行更新[Gilmer et al., 2017]。

在本章的其余部分，我们将详细介绍这个神经消息传递框架的基础。我们将关注消息传递框架本身，并将训练和优化GNN模型的讨论推迟到第6章。本章的大部分内容将描述我们如何获取输入图以及一组节点特征，并使用这些信息生成节点嵌入。我们还将讨论如何使用GNN框架生成子图和整个图的嵌入。

**5.1.1消息传递框架概述**

在GNN中的每个消息传递迭代期间，根据从u的图邻域聚集的信息，更新对应于每个节点的隐藏嵌入 (图5.1)。该消息传递更新可以表示如下:





**图5.1单个节点如何聚合来自其本地邻居的消息的概述。该模型聚集来自A的局部图邻居(即B、C和D)的消息，并且反过来，来自这些邻居的消息基于从它们各自的邻居聚集的信息，等等。这个可视化展示了消息传递模型的两层版本。请注意，GNN的计算图通过在目标节点周围展开邻域形成了一个树形结构。**

其中UPDATE和AGGREGATE是任意可微函数(即神经网络)，而是从的图邻域聚合的“消息”。我们使用上标来区分消息传递的不同迭代中的嵌入和函数。(消息传递的不同迭代有时也称为GNN的不同“层”。)

在GNN的每次迭代，聚集函数将的图邻域中的节点的嵌入集作为输入，并基于该聚集的邻域信息生成消息。然后更新函数update将消息与节点的先前嵌入相结合，以生成更新的嵌入。时的初始嵌入被设置为所有节点的输入特征，即。在运行GNN消息传递的次迭代之后，我们可以使用最终层的输出来定义每个节点的嵌入，即，

 (5.6)

请注意，由于AGGREGATE函数将集合作为输入，因此以这种方式定义的GNNs在设计上是置换等变的。

|  |
| --- |
| 节点特征  请注意，与本书第一部分讨论的浅层嵌入方法不同，GNN框架要求我们将节点特性作为模型的输入。在许多图中，我们将有丰富的节点特征来使用(例如，生物网络中的基因表达特征或社交网络中的文本特征)。但是，在没有节点功能可用的情况下，仍然有几个选项。一种选择是使用节点统计信息(如第2.1节中介绍的)来定义特征。另一种常用的方法是使用身份特征，其中我们将每个节点与唯一标识该节点的一个热指示器特征相关联。  但是，使用身份特征使模型具有转导性，并且无法推广到看不见的节点。 |

**5.1.2动机和直觉**

GNN消息传递框架背后的基本直觉很简单:在每次迭代中，每个节点都会聚集来自其本地邻域的信息，并且随着迭代的进行，每个节点嵌入都会包含来自图的更远区域的越来越多的信息。精确地说:在第一次迭代之后，每个节点嵌入包含来自其1-hop邻域的信息，即每个节点嵌入包含关于其直接图邻域的特征的信息，其可由图中长度为1的路径到达；在第二次迭代之后,每个节点嵌入包含来自其2-hop邻域的信息；通常，在k次迭代之后，每个节点嵌入都包含关于其-hop邻域的信息。

但是这些节点嵌入实际上编码了什么样的“信息”？通常，这些信息有两种形式。一方面是关于图形的结构信息。例如，在GNN消息传递的次迭代之后，节点的嵌入可能编码关于的-hop邻域中所有节点的度的信息。这些结构信息对许多任务都很有用。例如，当分析分子图时，我们可以使用程度信息来推断原子类型和不同的结构基序，如苯环。

除了结构信息，通过GNN节点嵌入捕获的另一种关键信息是基于特征的。在传递 GNN 消息的 k 次迭代之后，每个节点的嵌入也将关于它们的 k-hop 邻域中的所有特性的信息编码。神经网络的这种局部特征聚集行为类似于卷积神经网络(CNNs)中卷积核的行为。尽管CNN会根据图像中空间定义的色块聚合特征信息，但在GNN会基于局部图邻域聚合信息。我们将在第7章中更详细地探讨神经网络和卷积之间的联系。

**5.1.3 基本的GNN**

到目前为止，我们已经以一种相对抽象的方式讨论了GNN框架，即使用新和聚合函数进行的一系列消息传递迭代(公式5.4)。为了将公式(5.4)中定义的抽象GNN框架转换为我们可以实现的东西，我们必须给出这些更新和聚合函数的具体实例。我们从最基本的GNN框架开始，它是Scarselli等人[2009]提出的原始GNN模型的简化。

基本GNN消息传递定义为

 (5.7)

其中，是可训练的参数矩阵，而表示元素非线性(如tanh或RcLU)。为了简化符号，偏差项通常被省略，但是包含偏差项对于实现较强的性能很重要。在这个等式中——以及本书的其余部分——我们使用上标来区分GNN不同层中的参数、嵌入和维数。

在基本GNN框架中传递的信息类似于标准多层感知器(MLP)或埃尔曼型递归神经网络，即埃尔曼·RNN[埃尔曼，1990]，因为它依赖于线性运算，随后是单一元素的非线性。我们首先总结周围节点传来的消息；然后，我们使用线性组合将邻域信息与节点先前的嵌入相结合；最后，我们应用了一个基本的非线性。

我们可以通过更新和聚合函数等效地定义基本GNN:

 (5.8)

 (5.9)

召回函数如下：

 (5.10)

作为一种简写，表示从u的图邻域中聚集的消息。还要注意的是，为了简洁起见，我们在上面的等式中省略了表示迭代的上标。

(一般来说，参数，和可以通过迭代在GNN消息中共享，或者为每一层单独训练。)

|  |
| --- |
| 节点与图层方程  在上面对基本GNN模型的描述中，我们定义了节点层的核心消息传递操作。我们将在本章的大部分内容和整本书中使用这个约定。然而，重要的是要注意，许多神经网络也可以使用图层方程简洁地定义。在基本GNN的情况下，我们可以将模型的图层定义编写如下:  (5.11)  其中，表示的是GNN中层的节点表示矩阵(每个节点对应于矩阵中的一行)，是图邻接矩阵，为了简单起见，我们省略了偏差项。虽然这种图层表示并不容易适用于所有的GNN模型——比如我们下面讨论的基于注意力的模型——但它通常更简洁，并且还强调了使用少量稀疏矩阵运算可以有效地实现多少个神经网络。 |

**5.1.4带有自循环的消息传递**

作为神经消息传递方法的简化，通常在输入图中添加自循环，并省略显式更新步骤。在这种方法中，我们将消息传递简单地定义为

 (5.12)

现在集合被代替，即节点的邻居和节点本身。这种方法的好处是，我们不再需要定义显式的更新函数，因为更新是通过聚合方法隐式定义的。通过这种方式简化消息传递通常可以缓解过拟合，但也严重限制了GNN的表达能力，因为来自节点邻居的信息无法与来自节点本身的信息进行区分。

在基本GNN的情况下，添加self循环就相当于在和矩阵之间共享参数，给出如下图级更新:

 (5.13)

在下面的章节中，我们将把它称为自循环GNN方法。

**5.2广义邻域聚合**

公式(5.7)中概述的基本GNN模型可以实现较强的性能，它的理论能力是众所周知的(见第7章)。但是，就像一个简单的MLP或Elman RNN一样，基本的GNN可以在许多方面进行改进和推广。在这里，我们讨论如何对聚合操作符进行推广和改进，下面的部分(第5.3节)将对更新操作进行类似的讨论。

**5.2.1邻域归一化**

最基本的邻域聚合运算(等式5.8)只是取邻域嵌入的总和。这种方法的一个问题是它可能不稳定并且对节点度高度敏感。例如，假设节点u的邻居是节点u0的100倍(即，高得多的程度)，那么我们将合理地预期 (对于任何合理的向量范数)。这种幅度上的巨大差异会导致数值不稳定以及优化困难。

这个问题的一个解决方案是简单地基于所涉及的节点的程度来标准化聚合操作。最简单的方法是取平均值，而不是求和:

 (5.14)

但是研究人员也发现了其他标准化因素的成功，例如Kipf和Welling [2016a]采用的以下对称标准化:

 (5.15)

例如，在引用图(Kipf和Welling [2016a]分析的数据类型)中，来自高度节点的信息(即被引用多次的论文)对于推断图中的社区成员可能不是很有用，因为这些论文可以在不同的领域中被引用数千次。对称归一化也可以基于谱图理论来激活。具体来说，将对称归一化聚合(等式5.15)与基本GNN更新函数(等式5.9)相结合，得到谱图卷积的一阶近似，我们在第7章中对此进行了扩展。

图卷积网络(GCNs)

最流行的基线图神经网络模型之一——图卷积网络(GCN)——采用对称归一化聚合和自循环更新方法。因此，GCN模型将消息传递函数定义为

 (5.16)

这种方法首先由Kipf和Welling提出[2016a]，并被证明是最常用和最有效的基线GNN架构之一。

|  |
| --- |
| 归一化还是不归一化？  使用GNN时，适当的归一化对于实现稳定和强大的性能至关重要。但是，需要注意的是，归一化也会导致信息丢失。例如，在归一化之后，可能很难(甚至不可能)使用所学的嵌入来区分不同程度的节点，并且各种其他结构图特征可能被标准化所掩盖。事实上，使用等式(5.14)中的归一化聚合运算符的基本GNN比等式(5.8)中的基本总和聚合运算的功能更弱(参见第7章)。因此，归一化的使用是一个特定于应用程序的问题。通常，在节点特征信息远比结构信息更有用的任务中，或者在可能导致优化期间不稳定的节点度范围非常宽的任务中，归一化是最有帮助的。 |

**5.2.2设置聚合器**

邻域归一化是提高GNN性能的有用工具，但我们还能做更多来改进AGGREGATE操作吗?  除了对邻居嵌入进行求和之外，也许还有其他更复杂的方法吗？

邻域聚合操作本质上是一个集合函数。给我们一组领域嵌入并且必须将这个集合映射到单个向量。事实上，是一个集合这一事实非常重要:节点的领域没有自然的排序，因此我们定义的任何聚合函数都必须是置换不变的。

设置池

定义聚合函数的一个原则是基于排列不变神经网络的理论。例如，Zaheer等[2017]表明，以下形式的聚合函数是通用集函数逼近器:

 (5.17)

其中通常我们使用来表示由一些可训练参数参数化的任意深度多层感知器。Zaheer等人[2017]的理论结果表明，将一组嵌入映射到单个嵌入的任何置换不变函数都可以通过遵循等式(5.17)的模型近似到任意精度。

注意到在Zaheer等人[2017]中提出的理论在应用第一个之后采用了嵌入的总和(如等式5.17所示)。但是，也可以用一个归约函数来代替这个和，如Qi等人[2017]中的按元素划分的最大值或最小值，并且将基于等式(5.17)的模型与第5.2.1节中讨论的归一化方法结合起来也是常见的，如在GraphSAGE-pool方法中[Hamilton等人，2017b]。

基于等式(5.17)的集合汇集方法通常会导致性能的小幅提高，尽管它们也会增加过度拟合的风险，具体取决于所使用的的深度。如果使用集合池，通常使用只有一个隐藏层的多层模型，因为这些模型足以满足理论要求，但不会被过度参数化，以免出现严重的过度拟合。

Janossy池

在第5.1.3节中讨论的更基本的聚合体系结构的基础上，设置领域聚合的池方法基本上只是增加了额外的层。这个想法很简单，它会增加GNN的理论容量。然而，还有另一种被称为Janossy池的方法，它比简单地取相邻节点嵌入的和或平均值更有效[墨菲等人，2018]。

回忆一下，邻域聚合的挑战在于我们必须使用置换不变函数，因为节点的邻域没有自然排序。在集合池方法中(公式5.17)，我们通过依赖总和、均值或元素的最大值来将嵌入集减少到单个向量来实现这种置换不变性。通过将这种减少与前馈神经网络(即MLPs)相结合，使模型性能更好。Janossy池采用了一种完全不同的方法:我们不使用置换不变约简(例如，求和或均值)，而是使用置换敏感函数，对许多可能的置换求平均。

让表示一个置换函数映射集合到一个特定的序列。换句话说，生成相邻嵌入的无序集，并根据任意的顺序将这些嵌入放置在一个序列中。然后Janossy池方法执行邻域聚合：

 (5.18)

其中表示一组排列，是排列敏感函数，例如，对序列进行操作的神经网络。在实践中，通常被定义为一个LSTM函数[Hochleiter和Schmidhuber，1997]，因为已知LSTMs是一个强大的序列神经网络结构。

如果等式（5.18）中的置换集合Π等于所有可能的置换，则等式（5.18）中的聚合器也是集合的通用函数近似器，如等式（5.17）。 但是，对所有可能的排列求和通常很困难。 因此，在实践中，Janossy池采用以下两种方法之一：

1.在聚合器的每次应用期间，对可能排列的随机子集进行采样，并且只对该随机子集求和。

2.采用邻域集中节点的规范排序；例如，根据节点的程度以降序对节点进行排序，同时随机断开连接。

墨菲等人[2018]包括对这两种方法的详细讨论和经验比较，以及其他近似技术(例如，截断序列长度)，他们的结果表明，Janossy风格的池可以改进许多综合评估设置中的集合汇集。

**5.2.3 Neighborhood Attention**

除了更一般形式的集合聚合，在GNNs中改进聚合层的一个主流策略是应用attention[Bahdanau等人，2015]。基本思想是为每个相邻节点分配一个关注权重或重要程度，用于在聚合步骤中衡量该相邻节点的影响力。第一个应用这种attention风格的GNN模型是Velickovi等人[2018]的图形注意力网络(GAT)，它使用attention权重来定义邻域的加权和:

 (5.19)

其中，当我们在节点聚合信息时，表示对相邻节点的attention权重。在最初的GAT论文中，权重定义为

 (5.20)

以及使用MLPs对attention层次的变化，例如，

 (5.21)

其中MLP被限制为标量输出。

此外，虽然在GNN文献中并不常见，但也可以按照流行的transformer架构的风格添加多个attention节点[Vaswani et al.， 2017]。在此方法中，人们使用独立参数化的attention节点计算出K个不同的attention权值(例如)。使用不同attention权重聚合的消息然后在聚合步骤中进行转换和组合，通常使用线性投影和级联操作，例如:

 (5.23)

 (5.24)

其中，K个attention head的attention权值都可以用上述任何一种attention机制计算得到。

|  |
| --- |
| **Graph attention and transformers**  具有多头attention的GNN模型(等式5.23)与Transformer架构密切相关[Vaswani等人，2017]。Transformers是自然语言处理(NLP)和计算机视觉的流行架构，就NLP而言，它们一直是最先进的大型NLP系统背后的重要驱动力，例如BERT [Devlin等人，2018]和XLNet [Yang等人，2019]。Transformers背后的基本思想是完全基于attention操作来定义神经网络层。在Transformers中的每一层，通过使用多个attentions head来计算输入中所有位置对之间的注意权重，为输入数据中的每个位置(例如，句子中的每个单词)生成新的隐藏表示，然后基于这些attention权重(以类似于等式5.23的方式)用加权和来聚集这些attention权重。事实上，如果我们假设GNN接收到一个完全连通的图形作为输入，那么基本的transformer层就相当于使用多头 attention的GNN层(即等式5.23)。  GNNs和Transformers之间的这种联系已经在许多工程中得到利用。例如，设计GNNs的一个实现策略是简单地从一个tansformer模型开始，然后在attention层应用一个二进制邻接掩码，以确保信息仅在图中实际连接的节点之间聚合。这种风格的GNN实现可以受益于现有的变压器架构的众多精心设计的库。然而，这种方法的缺点是，tansformer必须计算输入中所有位置/节点之间的成对attention，这导致聚集图中所有节点的消息的时间复杂度为，而对于更标准的GNN实现，时间复杂度为 。 |

增加attention是增加GNN模型表示能力的一个有用策略，特别是在你事先知道一些相邻节点可能比其他相邻节点提供更多信息的情况下。例如，考虑基于引用网络将论文分类为主题类别的情况。通常有一些论文跨越了主题界限，或者在不同领域被高度引用。理想情况下，基于注意力的GNN将学会忽略神经信息传递中的这些文件，因为当试图识别特定节点的主题类别时，这种混杂的领域可能不会提供信息。在第七章中，我们将从信号处理的角度讨论attention如何影响神经网络的感应偏置。

**5.3 Generalized Update Methods**

GNN模型中的聚集算子在提出新的架构和变体方面通常受到研究者的最大关注。尤其是在引入了广义邻域聚合概念的GraphSAGE框架之后更是如此[Hamilton等人，2007b]。 然而，GNN消息传递涉及两个关键步骤：聚合和更新，在许多方面，UPDATE算子在定义GNN模型的功率和诱导偏差方面起着同样重要的作用。

到目前为止，我们已经看到了基本的GNN方法-其中更新操作涉及节点当前嵌入与其邻居消息的线性组合-以及自环方法，它只是在执行邻域聚合之前向图中添加一个自环。 在本节中，我们将注意力转向UPDATE操作符的更多样化的概括。

|  |
| --- |
| 过度平滑和邻域影响  GNNs的一个常见问题——广义更新方法有助于解决这个问题——被称为过度平滑。过度平滑的基本思想是，在GNN消息传递的几次迭代之后，图中所有节点的表示可以变得彼此非常相似。这种趋势在基本GNN模型和采用自循环更新方法的模型中尤其常见。过度平滑是有问题的，因为它不可能建立更深的GNN模型——它利用了图中的长期相关性——因为这些深GNN模型倾向于产生过度平滑的嵌入。  GNNs中的过平滑问题可以通过定义每个节点的输入特征对图中所有其他节点的最终层嵌入的影响来形式化，即，。特别地，对于任何一对节点和，我们可以通过检查相应雅可比矩阵的大小来量化节点对GNN中节点的影响[Xu等人，2018]:  (5.25)  在上式中代表所有的向量。值使用雅可比矩阵中条目的总和作为节点的初始嵌入对GNN中节点的最终嵌入的影响程度的度量。  鉴于影响力的这一定义，徐等[2018]证明如下:  **定理3** 对于使用自循环更新方法和聚合函数形式的任何GNN模型  (5.26)  其中是一个任意可微的归一化函数，我们有  (5.27)  其中表示在从节点开始的长度为K的随机漫步中访问节点的概率。  这个定理是徐等人[2018]中**定理**1的直接结果。它指出，当我们使用K层GCN型模型时，节点和节点的影响与从节点开始的K步随机游走中到达节点的概率成比例。然而，这种情况的一个重要结果是，当，每个节点的影响接近图中随机游走的平稳分布，这意味着局部邻域信息丢失。此外，在许多真实世界的图中——这些图包含高次节.点并且类似于所谓的“扩展”图——从任何节点开始的随机行走只需要步就能收敛到几乎均匀的分布[Hoory等人，2006]。  **定理**3直接适用于使用自循环更新方法的模型，但是结果也可以在基本GNN更新(即等式5.9)的无耳意义上扩展，只要在每层k处相邻。因此，当使用简单的GNN模型时——尤其是那些使用自循环更新方法的模型——构建更深的模型实际上会损害性能。随着更多的层被添加，我们丢失了关于局部邻域结构的信息，我们学习的嵌入变得过度平滑，接近几乎均匀的分布。 |

**5.3.1连接和跳过连接**

如上所述，过度平滑是神经网络的核心问题。当特定于节点的信息在GNN消息传递的几次迭代后变得“过时”或“丢失”时，就会发生过度平滑。直观地说，在消息传递期间从节点邻居聚集的信息开始主导更新的节点表示的情况下，我们可以预期出现过度模拟。在这些情况下，更新的节点表示(即，向量)将过于强烈地依赖于从领域聚集的输入消息(即，向量)，而牺牲了来自先前层的节点表示(即，向量)。缓解这个问题的一个自然方法是使用向量连接或跳过连接，这种方法试图在更新步骤中直接保存前几轮消息传递中的信息。

这些连接和跳过连接方法可以与大多数其他GNN更新方法结合使用。因此，为了一般性起见，我们将使用来表示我们正在构建的基础更新函数(例如，我们可以假设由等式5.9给出)，并且我们将在这个基础函数之上定义各种跳过连接更新。

最简单的跳过连接更新之一是在消息传递过程中使用级联来保留更多的节点级信息:

 (5.28)

在这里，我们简单地将基本更新函数的输出与节点的上一层表示连接起来。同样，这里的关键直觉是，我们鼓励模型在消息传递过程中解开信息——从每个节点(即)的当前表示中分离来自相邻节点(即)的信息。

基于级联的跳过连接是在GraphSAGE框架中提出的，它是第一批强调这种对更新函数的修改可能带来的好处的工作之一[Hamilton等人，2017a]。然而，除了级联，我们还可以采用其他形式的跳跃连接，例如Pham等人[2017]提出的线性插值方法:

 (5.29)

其中门控向量，和表示元素乘法。在这种方法中，最终更新的表示是先前表示和基于邻域信息更新的表示之间的线性插值。门控参数可以通过多种方式与模型联合学习。例如，Pham等人[2017]生成作为单独的单层GNN的输出，其将当前隐藏层表示作为特征。然而，也可以采用其他更简单的方法，例如简单地直接学习每个消息传递层的参数，或者使用当前节点表示上的MLP来生成这些选通参数。

一般来说，这些连接和剩余连接是简单的策略，有助于减轻神经网络中的过平滑问题，同时也提高了优化的数值稳定性。实际上，类似于卷积神经网络中剩余连接的效用[何等人，2016]，将这些方法应用于卷积神经网络可以促进更深层次模型的训练。在实践中，这些技术往往对具有中等深度的神经网络(例如，2-5层)的节点分类任务最有用，并且它们在表现出同质性的任务上表现出色，即，每个节点的预测与其局部邻域的特征密切相关。

**5.3.2门控更新**

在前一节中，我们讨论了跳跃连接和剩余连接方法，它们与计算机视觉中用来构建更深层次的有线电视网络体系结构的技术有很强的相似性。在类似的工作中，研究人员也从用于提高递归神经网络的稳定性和学习能力的门控方法中获得了灵感。具体来说，查看GNN消息传递算法的一种方式是聚合函数从领域那里接收观察结果，然后使用该观察结果来更新每个节点的隐藏状态。在这种观点下，我们可以根据观察结果直接应用用于更新RNN建筑隐藏状态的方法。例如，最早的架构之一[李等人，2015]将更新功能定义为：

 (5.30)

其中GRU表示门控循环单位(GRU)细胞的更新方程[Cho等人，2014]。其他方法采用了基于LSTM体系结构的更新[Selsam等人，2019]。

一般来说，任何为网络节点定义的更新函数都可以在网络节点的上下文中使用。我们简单地用节点的隐藏状态替换RNN更新函数的隐藏状态参数(通常表示为)，并用从本地邻域聚集的消息替换观察向量(通常表示为)。重要的是，这种RNN式更新的参数总是在节点间共享的(即，我们使用相同的LSTM或GRU单元来更新每个节点)。在实践中，研究人员通常在GNN的消息传递层共享更新函数的参数。

这些门控更新在促进深度GNN架构(例如，超过10层)和防止过度平滑方面非常有效。通常，它们最适用于预测任务需要对图的全局结构进行复杂推理的应用，例如程序分析应用[李等人，2015]或组合优化应用[Selsam等人，2019]。

**5.3.3跳跃知识连接**

在前面的部分中，我们已经隐式地假设我们正在使用GNN的最后一层的输出。换句话说，我们一直假设我们在下游任务中使用的节点表示等于GNN中的最终层节点嵌入:

 (5.31)

这一假设是由许多GNN方法提出的，这一策略的局限性促使人们需要进行剩余更新和门控更新来限制过度平滑。

然而，提高最终节点表示质量的补充策略是在消息传递的每一层简单地利用表示，而不是只使用最终层输出。在这种方法中，我们将最终节点表示定义为

 (5.32)

其中是任意可微函数。这种策略被称为添加跳跃知识(JK)连接，由徐等人[2018]首次提出并分析。在许多应用中，函数可以简单地定义为身份函数，这意味着我们只是连接每一层的节点嵌入，但徐等人[2018]也探索了其他选项，如最大池方法和关注层。这种方法通常会在各种各样的任务中带来一致的改进，并且通常是一种有用的策略。

**5.4边缘特征和多关系神经网络**

到目前为止，我们对神经网络和神经消息传递的讨论隐含地假设我们有简单的图。然而，在许多应用中，所讨论的图是多关系的或者异构的(例如，知识图)。在本节中，我们将回顾一些主流的策略，这些策略是为了适应这些数据而开发的。

**5.4.1关系图神经网络**

为解决这个问题而提出的第一种方法通常被称为关系图卷积网络(RGCN)方法[Schlichtkrull等人，2017]。在这种方法中，我们通过为每种关系类型指定一个单独的转换矩阵来扩充聚合函数以适应多种关系类型:

 (5.33)

其中是一个归一化函数，它可以依赖于节点的邻域以及被聚集的邻域。Schlichtkrull等人[2017]讨论了几种标准化策略，以定义类似于第5.2.1节中讨论的。总的来说，RGCN中的多关系聚合类似于带有规范化的基本GNN方法，但是我们单独聚合不同边缘类型的信息。

参数共享

原始RGCN方法的一个缺点是参数的数量急剧增加，因为现在每种关系类型都有一个可训练的矩阵。在某些应用中，例如在具有许多不同关系的知识图上的应用，这种参数的增加会导致过度拟合和学习缓慢。Schlichtkrull等人[2017]提出了一个通过与基矩阵共享参数来解决这个问题的方案，其中

 (5.34)

在这种基矩阵方法中，所有的关系矩阵被定义为基矩阵。对于每个关系，唯一关系特定的参数是个组合权重。在这种基础共享方法中，我们可以将整个聚合函数重写为

 (5.35)

式中是将基础矩阵叠加而成的张量， 是包含关系的基组合权重的向量,表示沿模式的张量积。因此，参数共享RGCN方法的另一种观点是，我们正在学习每个关系的嵌入以及所有关系共享的张量。

延伸和变化

RGCN体系结构可以在许多方面进行扩展，一般来说，我们指的是将每个关系定义为独立聚合矩阵的方法，即关系图神经网络。例如，Zitnik等人(2018年)采用了这种方法的一种变体(没有参数共享)来建模与药物、疾病和蛋白质相关的多关系数据集，Marcheggiani和Titov(2017年)采用了类似的策略来分析语言依赖图。其他工作已经成功地将RGCN风格的聚合与注意力结合起来[Teru等人，2020]。

**5.4.2 attention和特征连接**

在关系GNN方法中，我们为每个关系定义一个单独的聚合参数，这种方法适用于多关系图和具有离散边特征的情况。为了适应我们有更一般形式的边缘特征的情况，我们可以在注意中利用这些特征，或者通过在消息传递期间将这些信息与邻居嵌入连接起来。例如，给定任何基础聚合方法聚合基础利用边缘特征的一个简单策略是定义一个新的聚合函数，如下所示

 (5.36)

其中表示边的任意向量值特征。这种方法简单而通用，并且最近成功地将基于注意力的方法作为基础聚合函数[Sinha等人，2019]。

**5.5图形池**

神经消息传递方法产生了一组节点嵌入，但是如果我们想在图的层次上进行预测呢？换句话说，我们一直假设目标是学习节点表示，但是如果我们学习整个图的嵌入呢？这个任务通常被称为图形池，因为我们的目标是将节点嵌入集合在一起，以便学习整个图形的嵌入。

设定池化方法

类似于AGGREGATE操作，图形池的任务可以被看作是集合学习的问题。我们想设计一个池函数，它映射一组节点嵌入到表示整个图形的嵌入。事实上，在第5.2.2节中讨论的学习邻居嵌入集的任何方法也可以在图的层次上用于池化。

在实践中，有两种方法最常用于通过集合池学习图级嵌入。第一种方法是简单地计算节点嵌入的总和(或平均值):

 (5.37)

其中是一些归一化函数(例如，恒等式函数)。虽然非常简单，但是基于节点嵌入的总和或平均值的池化对于涉及小图的应用程序来说通常是足够的。

第二种流行的基于集合的方法使用LSTMs和注意力的组合来汇集节点嵌入，其方式受到Vinyals等人[2015]工作的启发。在这种池方法中，我们迭代一系列基于注意力的聚合，这些聚合由下面的方程组定义，对测试:

 (5.38)

 (5.39)

 (5.40)

 (5.41)

在上面的等式中，向量表示每次迭代时的attention查询向量。在等式(5.39)中，使用关注函数 (例如，点积)，查询向量被用于计算每个节点上的关注分数，然后该关注分数在等式(5.40)中被归一化。最后，在等式(5.41)中，基于关注权重计算节点嵌入的加权和，并且该加权和用于使用LSTM更新来更新查询向量(等式5.38)。通常，和向量用全零值初始化，并且在迭代等式(5.38)-(5.41)进行次迭代之后，整个图的嵌入被计算为

 (5.42)

这种方法代表了一种复杂的基于关注的集合汇集体系结构，并且它已经成为许多图级分类任务中流行的汇集方法。

图形粗化方法

集合池方法的一个限制是它们没有利用图的结构。虽然将图形汇集任务简单地视为一个集合学习问题是合理的，但是在汇集阶段利用图形拓扑也有好处。实现这一点的一种流行策略是执行图形聚类或粗化，作为汇集节点表示的一种手段。

在这些方法中，我们假设我们有一些聚类功能

 (5.43)

它将图中的所有节点映射到簇上的一个赋值。我们假设这个函数输出一个赋值矩阵，其中表示节点和簇之间的关联强度。函数的一个简单示例是第1章中描述的谱聚类方法，其中聚类分配基于图邻接矩阵的谱分解。在更复杂的定义中，人们实际上可以使用另一个GNN来预测簇分配[应等人，2018b]。

不管用于生成聚类分配矩阵的方法是什么，图形粗化方法的关键思想是我们使用这个矩阵来粗化图形。特别地，我们使用分配矩阵来计算新的粗化邻接矩阵

 (5.44)

一组新的节点特征

 (5.45)

因此，这个新的邻接矩阵现在表示图中簇之间的关联强度(即边)，并且新的特征矩阵表示分配给每个簇的所有节点的聚集嵌入。然后，我们可以在这个粗化的图形上运行GNN，并重复整个粗化过程若干次迭代，其中图形的大小在每一步都减小。该图的最终表示然后通过集合汇集足够粗化的图中的节点的嵌入来计算。

这种基于粗化的方法受到了卷积神经网络(CNNs)中使用的pooling方法的启发，它依赖于一种直觉，即我们可以构建对输入图的不同粒度进行操作的层次神经网络。在实践中，这些粗化的方法可以带来强大的性能，但它们也可能不稳定，难以训练。例如，为了使整个学习过程端到端可微，聚类函数fc必须是可微的，这就排除了大多数现有的聚类算法，如光谱聚类。Cangea等人[2018]对该领域的不同方法进行了深刻的比较，强调了学习稀疏集群分配的好处以及在实施这些粗化方法时面临的一些挑战。

**5.6广义消息传递**

到目前为止，本章的介绍主要集中在最流行的GNN消息传递样式上，它主要在节点级别上运行。然而，GNN消息传递方法也可以推广到在消息传递的每个阶段利用边缘和图级信息。

例如，在Battaglia等人[2018]提出的更通用的方法中，我们根据以下等式定义消息传递的每次迭代:

 (5.46)

 (5.47)

 (5.48)

 (5.49)

这种广义消息传递框架的重要创新在于，在消息传递过程中，我们为图中的每条边生成隐藏的嵌入，以及对应于整个图的嵌入。这使得消息传递模型能够轻松地集成边缘和图形级特性，并且最近的研究结果也显示了这种广义消息传递方法在逻辑表达能力方面比标准GNN具有优势[Barcelo et al.， 2020]。生成边缘的嵌入以及整个消息传递过程中的图形处理也使根据图形或边缘级分类任务定义损失函数变得很简单。对于这种通用消息传递框架中的消息传递操作，我们首先基于事件节点的嵌入来更新边缘嵌入（等式5.46）。接下来，我们通过汇总所有入射边缘的边缘嵌入来更新节点嵌入（等式5.47和5.48）图嵌入用于节点和边表示的更新方程中，图级嵌入本身通过在每次迭代结束时聚合所有节点和边嵌入来更新（等式5.49）。在这样一个通用的消息传递框架中，所有单独的更新和聚合操作都可以使用本章中讨论的技术来实现(例如，使用池方法来计算图级更新)。

**第6章**

**图神经网络在实践中的应用**

在第五章中，我们介绍了一些图形神经网络(GNN)的结构。 然而，我们没有讨论这些体系结构是如何优化的，以及什么样的损失函数和正则化 通常使用ON。 在本章中，我们将把注意力转向GNNs的一些实际方面。 我们将讨论一些有代表性的应用程序，以及GNN一般是如何操作的 在实践中，包括讨论无监督的训练前方法，可以特别有效。 我们还将介绍常用的技术，用于正规化和提高效率 GNN。

应用和损失函数

在当前的绝大多数应用中，GNN用于三个任务之一：节点分类、图分类或关系预测。 正如第一章所讨论的，这些任务反映了 大量的现实应用，如预测用户是否是社交网络中的机器人（节点分类），基于分子图结构的属性预测(图类 在线平台中的内容推荐（关系预测）。 在本节中，我们简要描述了这些任务如何转化为GNNs的具体损失函数，我们还简要介绍了GNNs的具体损失函数 讨论如何以无监督的方式预先训练GNN，以提高这些下游任务的性能。

在下面的讨论中，我们将使用zu∈RD表示GNN的最后一层的节点嵌入输出，我们将使用zG RD表示池functi的图级嵌入输出 上。 原则上，第5章中讨论的任何GNNAp-proaches都可以用于生成这些嵌入。 通常，我们将在zu和zGem上定义损失函数-

床上用品，我们将假设损失的梯度是反向传播的

通过使用随机梯度下降或其变体之一的GNN参数[Rumelhart等人，1986年]。

**6.1.1 用于节点分类的GNN**

节点分类是GNN最流行的基准任务之一。 例如，在2017年至2019年期间-当GNN方法开始在机器学习中获得突出地位-参考 在GNN上，H主要由Cora、Citeseer和Pubmed引文网络基准主导，由Kipf和Welling[2016a]推广]。 这些基线涉及对其类别或专题进行分类 科学论文基于它们在引文网络中的位置，具有基于语言的节点特征（例如，单词向量），并且每个类仅给出极少数的正面示例(USU 少于10%的节点)。

将GNN应用于这种节点分类任务的标准方法是以完全监督的方式训练GNN，其中我们使用Softmax经典函数和负log-likeli定义损失 新闻:

 （6.1）

在这里，我们假设Yu∈Zc是一个单热向量，表示训练节点u∈Vtrain的类别；例如，在引文网络设置中，Yu将表示论文u的主题。 我们用softmax (zu，yu)表示节点属于yu类的预测概率，通过Softmax函数计算：

 （6.2）

其中 ，i=1，...，c是可训练的参数。 有监督节点损失的其他变化，但基于方程（6.1）中的损失以监督的方式训练GNN是最常见的OP之一 GNNs的时间化策略。

**监督，半监督，转导和归纳注释**

正如第1章所讨论的那样，节点分类设置通常被称为监督和半监督。 在应用这些术语时，一个重要的因素是是否和如何不同 节点在训练GNN期间使用。 一般来说，我们可以区分三种类型的节点：

1. 有训练节点集合，。 这些节点包含在GNN消息传递操作中，也用于计算损失，如公式(6.1)。

2除了训练节点，我们还可以有转导测试节点。这些节点没有标记，也不用于损失计算，但是这些节点和它们的事件边缘仍然参与GNN消息传递操作。也就是说，在GNN消息传递操作过程中，GNN会为u∈中的节点生成隐藏表示。 但是最后一层嵌入的zu将不会用于这些节点的损耗函数计算。

3.最后，我们还将有归纳测试节点。在训练期间，这些节点既不用于损失计算，也不用于GNN消息传递操作，这意味着在训练GNN时，这些节点——以及它们的所有节点——完全没有被观察到。

术语半监督适用于在转换测试节点上测试GNN的情况，因为在这种情况下，GNN在训练期间观察测试节点(而不是它们的标签)。术语归纳节点分类用于区分测试节点及其所有事件边缘在训练期间完全未被观察到的设置。归纳节点分类的一个例子是在引文网络的一个子图上训练一个GNN，然后在一个完全不相交的子图上测试它。

**6.1.2 用于图形分类的GNN**

与节点分类类似，图级分类上的应用程序被弹出作为基准任务。 从历史上看，核方法在图形分类中很受欢迎，结果就是一些最流行的图形分类早期基准是从内核文献中改编而来的，例如涉及基于基于图形的REPRE-Sentat的酶特性分类的任务离子[Morris等人，2019年]。 在这些任务中，通常使用类似于方程（6.1）的Softmax分类损失，其关键区别在于

损失是用图级嵌入在一组标记训练图上计算T = {G1, ..., Gn}。 近年来，GNNs也取得了成功

在涉及图形数据的回归任务中-特别是涉及从基于图形的分子表示中预测分子性质（例如溶解度）的任务。 在这些情况下，标准是采用下列形式的平方误差损失：

 （6.3）

其中MLP是一个具有单变量输出的密集连接神经网络，yGi∈R是训练图Gi的目标值。

**6.1.3 用于关系预测的GNNs**

虽然分类任务是迄今为止最流行的GNNs应用，但GNNs也被用于关系预测任务，如推荐系统[Ying等人，2018a]和知识图Co 完成[Schlichtkrull等人，2017年]。在这些应用中，标准实践是使用第3章和第4章中引入的成对节点嵌入损失函数。原则上，GNN 可以与这些章节中讨论的任何成对损失函数相结合，用GNN的输出代替浅层嵌入。

**6.1.4 预训练GNNs**

预训练技术已成为深度学习的标准实践[Good-Fellow等人，2016年]。 在GNN的情况下，人们可以想象使用一个邻域重新训练GNN 第三章的构建损失可能是提高下游分类任务性能的一种有用的策略。 例如，可以预先训练GNN来重建图中缺失的边 在对节点分类损失进行微调之前。

然而，有趣的是，这种方法在GNNs的con-文本中几乎没有取得成功。事实上，Veliˇckovi´c等人 [2019]甚至发现，与一个预先训练的邻域重建损失相比，随机初始化的GNN同样强大。解释这一发现的一个假设是GNN消息传递已经有效地编码了邻域信息这个事实。由于消息传递的结构，图中的邻域节点往往在GNN中具有相似的嵌入，因此强制邻域重构结构损失可以简单地被冗余。

尽管在社区重建损失的预训练方面取得了这种负面结果，但使用其他预训练策略也取得了积极的结果。例如，Velickovic等人 [2019年]提出的Deep Graph Infomax (DGI)，DGI涉及最大化节点嵌入Zu和图片嵌入Zc之间的相互信息。从形式上讲，这种方法优化了以下损失：

 （6.4）

在这里，表示基于图G的GNN生成的节点u的嵌入，而表示基于cor-rupted版本图生成的节点u的嵌入，表示 。我们使用D来表示判别器函数，它是一个神经网络，它被训练用来预测节点嵌入是来自真实的图G还是损坏的版本 。通常，通过以某种随机方式修改节点特征、邻接矩阵或两者（例如，特征矩阵的洗牌)来破坏图形）。这种损失背后的直觉是GNN模型必须学会生成能够区分真实图及其损坏对应物的节点嵌入。可以看出，这种优化与最大化节点嵌入Zu和图级嵌入ZG之间的相互信息密切相关。

在DGI（方程6.4）中使用的损失函数只是一个更广泛的例子，一类非监督目标，在GNNs的背景下取得了成功[Hu等人，2019年，Sun等人[2020].这些无监督的训练策略通常涉及训练GNN，以最大限度地利用不同表示级别之间的相互信息，或区分真实和损坏对嵌入。 从概念上讲，这些训练前的方法-有时也被用作监督训练过程中的辅助损失-与“内容掩蔽”训练前的方法相似，这些方法在自然语言处理中迎来了一种新的艺术状态[Devlin等人，2018年]。 然而，GNN培训前方法的推广和改进是一个开放和活跃的研究领域。

**6.2 效率关注和节点抽样**

在第五章中，我们主要从节点级消息传递方程的角度来讨论GNN。 然而，基于这些方程直接实现GNN在计算上可能是低效的。 例如，如果多个节点共享邻居，如果我们独立地为图中的所有节点实现消息传递操作，我们可能最终会进行冗余计算。 在本节中，我们将讨论一些可以有效地实现GNN的策略。

**6.2.1 图级实现**

在最小化运行消息传递所需的数学操作数量方面，最有效的策略是使用GNN方程的图级实现。 我们在上一章的5.1.3节中讨论了这些图级方程，其关键思想是实现基于稀疏矩阵乘法的消息传递操作。 例如，基本GNN的图级方程由

 （6.5）

在哪里(t) 是一个矩阵，包含图中所有节点的层-k嵌入。 使用这些方程的好处是没有冗余计算-也就是说，我们计算嵌入h(k) 在运行模型时，对于每个节点u精确一次。 然而，这种方法的局限性在于它需要同时在整个图和所有节点特征上操作，这可能由于内存限制而不可行。此外，使用图级方程基本上将一个限制为满批（而不是最小的）梯度下降。

**6.2.2 次采样和迷你匹配**

为了限制GNN的内存占用，并方便进行小批处理培训，可以在消息传递过程中使用节点子集。 从数学上讲，我们可以把它看作是在每个批处理中运行图中节点子集的节点级GNN方程。 通过仔细的工程可以避免冗余计算，以确保我们只计算嵌入h(k)对于批处理中的每个节点u，运行时最多一次模型。

然而，挑战是，我们不能简单地运行传递在图中节点子集上的消息而不丢失信息。 每次删除节点时，我们也会删除它的边（即修改邻接矩阵）。 不能保证选择节点的随机子集甚至会导致连接图，并且为每个小批选择节点的随机子集会对模型性能产生严重的不利影响。

Hamilton等人[2017b]提出了一种通过对节点邻域进行次采样来克服这一问题的策略。 其基本思想是首先为批处理选择一组目标节点，然后递归地对这些节点的邻居进行采样，以确保图的连通性得到保持。 为了避免对一个批处理采样过多节点的可能性，Hamilton等人。 [2017b]建议对每个节点的邻居进行子样本，使用固定的样本大小来提高批处理张量操作的效率。 在后续工作中提出了更多的次采样想法[Chen等人，2018年]，这些方法对于使GNN可扩展到大规模的现实世界图至关重要[Ying等人，2018a]。

**6.3 参数共享和正则化**

正则化是任何机器学习模型的关键组成部分。 在GNNs的背景下，许多标准正则化方法都很好地工作，包括L2正则化、辍学[Srivastava等人，2014年]和层归一化[Ba等人，2016年]。然而，也有一些特定于GNN设置的正则化策略。

**跨层参数共享**

在具有多层消息传递的GNN中经常使用的一种策略是参数共享。 其核心思想是在GNN中的所有AGGREGATE和UPDATE函数中使用相同的参数。 一般来说，这种方法在六层以上的GNNs中是最有效的，它经常与门控更新函数一起使用（见第5章）[Li等人，2015年，Selsam等人，2019年]。

**边缘缀学**

另一种GNN特定的策略被称为边缘丢失。在这种正则化策略中，我们在邻接矩阵中随机删除（或掩码）边训练，直觉认为这将使GNN不太容易过度拟合，并且对邻接矩阵中的噪声更健壮。 这种方法在将GNN应用于知识图方面特别成功[Schlichtkrull等人，2017年；Teru等人，2020年]，这是原始图注意网络(GAT)工作中使用的一项基本技术[Veliˇckovi´c等人，2018年]。还请注意，其中讨论的邻域子抽样方法第6.2.2节将这种正则化作为副作用，使其成为大规模GNN应用中非常常见的策略。

**第七章**

**理论动机**

在这一章中，我们将访问图神经网络(GNNs)的一些理论基础)。 GNNs最有趣的方面之一是它们是从不同的理论动机中独立发展起来的。 从一个角度来看，GNN是基于图信号处理理论开发的，作为欧氏卷积对非欧氏图域的推广[Bruna等人，2014年]。 然而，同时，神经消息传递方法-它构成了大多数现代GNNs的基础-被类比为图形模型中概率推理的消息传递算法[Dai等人，2016]。 最后，基于与Weisfeler-Lehman图同构检验的连接，GNNs在几项工作中得到了激励[Hamilton等人，2017b]。

将三个不同的区域合并成一个单一的算法框架是显著的。 也就是说，这三种理论动机中的每一种都有自己的直觉和历史，人们所采用的观点可以对模型的发展产生实质性的影响。事实上，我们推迟审议并非偶然，这些理论动机的描述，直到GNN模型本身引入之后。 在这一章中，我们的目标是介绍关键的想法，在这些不同的理论动机的基础上，让感兴趣的读者自由地探索和结合这些直觉和动机，因为他们认为合适。

**7.1.1 卷积和傅里叶变换**

为了将卷积的概念推广到图，我们首先必须定义我们想要推广的内容，并提供一些简短的背景细节。设f和h是两个函数， 我们可以定义一般连续卷积运算为

 （7.1）

卷积运算的一个关键方面是它可以通过两个函数的傅里叶变换在元素上的乘积来计算:

 （7.2）

这里面

 （7.3）

是f(X)的傅里叶变换，其逆傅里叶变换被定义为

 （7.4）

在有限域上的一元离散数据的简单情况下(即限制于有限脉冲响应滤波器）我们可以将这些操作简化为离散的圆形卷积**（注释1）**

 （7.5）

和离散傅里叶变换(DFT)

 （7.6）

 （7.7）

在这里是对应于序列的傅里叶系数， 在方程（7.5）中，我们使用符号N 强调这是在有限域{0,1，...，N-1}上定义的圆形卷积，但为了符号简单起见，我们经常省略这个下标。

**（注释,1，为了简单起见，我们将自己限制在f和h的有限支持上，并使用模量算子和圆形卷积来定义边界条件。）**

解释（离散)傅里叶变换傅里叶变换本质上告诉我们如何将我们的输入信号表示为(复杂）正弦波的加权和。如果我们假设这两个输入数据它的傅里叶变换是实值的，我们可以解释序列作为傅里叶级数的系数。在这个观点里面，告诉我们是复正弦分量的振幅，是频率（弧度）。我们通常会讨论高频组件，它有一个很大的k,并且变化非常快，同样我们也会讨论低频组件，它的k<<N并且变化非常缓慢。这种低和高的概念频率分量也将在图域中有一个模拟，其中我们将考虑在图中节点之间传播的信号。

在信号处理方面，我们可以将离散卷积fh看作是序列通过滤器h的滤波操作。一般情况下，我们认为该系列与信号的值始终对应，卷积算子应用一些滤波器（例如带通滤波器）来调制这个时变信号。

卷积的一个关键特性，我们会依赖于下面的十字，事实上它们是平移（或转换）等价变换：

 （7.8）

这一特性意味着转换信号，然后通过滤波器对其进行卷积，相当于转换信号，然后转换结果。 注意，作为推论卷积，也等于差分运算：

 (7.9)

在这个式子里

 （7.10）

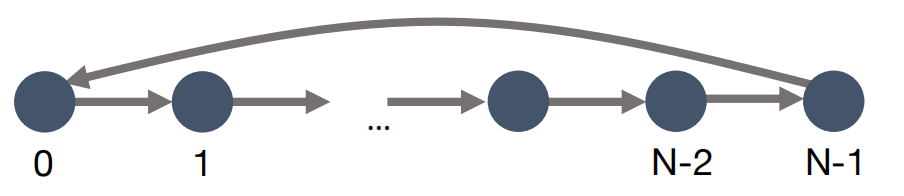
是离散单变量信号上的拉普拉斯（即差分）算子。

这些滤波和平移等差的概念是数字信号处理(DSP)的核心，也是卷积神经网络(CNNs)的直觉，它利用二维数据上的离散卷积。 我们不会试图在这里涵盖数字SIG-NAL处理、傅里叶分析和谐波分析的一小部分领域，我们将读者指向关于这些主题的各种教科书[Grafakos，2004年；Katznelson，2004年；Oppenhe im等人，1999年；Rabiner和Gold，1975年]。

**7.1.2 从时间信号到图形信号**

在上一节中，我们（简要）介绍了关于离散时变信号的滤波和卷积的概念。我们现在讨论如何将离散时变信号与图上的信号连接起来。 假设我们有一个离散的时变信号f(T0)，f(t2)，...，f(tn−1)。 查看此信号的一种方法是对应于链（或循环)图(图7.1），其中

时间t中的每个点表示为一个节点，每个函数值f(T)表示当时/节点的信号值。 取此视图，方便将信号表示为向量f∈R n，每个维度对应链图中的不同节点。 换句话说，我们有f[t]=f(t)（作为一种轻微的符号滥用）。 因此，图中的边表示信号是如何传播的，即信号在时间上向前传播。**（注释2）**



**图7.1：（循环）时间序列作为链图的表示**

将时变信号看作链图的一个有趣的方面是，我们可以使用图的邻接矩阵和拉普拉斯矩阵来表示操作，例如时间变换。 特别是，该链图的邻接矩阵对应于循环矩阵Ac与

 （7.11）

和（未归一化的）LaplacianLc对于这个图可以定义为

 （7.12）

我们可以用邻接矩阵表示时间位移

 （7.13）

以及拉普拉斯乘法的差分运算

 （7.14）

通过这种方法，我们可以看到图的邻接矩阵和拉普拉斯矩阵之间存在着密切的联系，以及信号的位移和差异的概念。 用邻接矩阵乘法信号将信号从节点传播到节点，Laplacian乘法计算每个节点上的信号与其近邻之间的差异。

给定这种基于图的通过矩阵乘法变换信号的视图，我们可以类似地将滤波器h的卷积表示为向量f上的矩阵乘法：

 （7.15）

 （7.16）

**（注释2，注意，我们将链中最后一个节点和第一个节点之间的连接作为边界条件，以保持域有限。）**

在这里是滤波器函数h的卷积运算的矩阵表示，还有是函数f的向量表示。 因此，在这个观点中，我们考虑卷积，它可以表示为图中每个节点上信号的矩阵变换**（注释3）**。当然，要取得方程（7.15)和方程(7.16）之间的平等。 矩阵 必须有一些特定的属性。 特别是，我们要求这个矩阵的乘法满足平移等差，它对应于循环邻接矩阵的交换性，也就是说，我们需要

 （7.17）

与差分算子的等价性类似地定义为

 （7.18）

可以证明，对于真实矩阵Q，这些要求是满足的h如果

 （7.19）

即如果是邻接矩阵的多项式函数 。 在数字信号处理术语中，这相当于将一般滤波器表示为移位算子的多项式函数的思想[Ortega等人，2018年]。**（注释4）**

**（注释3，这假设一个实值滤波器h）**

**（注释4，然而，请注意，有些卷积滤波器（例如复值滤波器）不能用这种方式表示。）**

**推广到一般图形**

我们现在已经看到了如何基于链图的邻接矩阵和拉普拉斯矩阵来表示时变离散信号的位移和卷积。 考虑到这一观点，我们可以很容易地将这些概念推广到更一般的图中。

特别是，我们看到时变离散信号对应于链图，平移/差分等差的概念对应于该链图的邻接/Laplacian交换性质。 因此，我们可以通过考虑任意邻接矩阵和拉普拉斯来将这些概念推广到链图之外。 虽然信号只是在链图中及时向前传播，但在任意图中，我们可能有多个节点相互传播信号，这取决于邻接矩阵的结构。 基于这一思想，我们可以将一般图上的卷积滤波器定义为矩阵与邻接矩阵或拉普拉斯通勤。

更准确地说，对于具有邻接矩阵A的任意图，我们可以将卷积滤波器表示为以下形式的矩阵：

 （7.20）

直观地说，这给了我们一个卷积滤波器在图上的空间构造。 特别是，如果我们乘一个节点特征向量通过这样的卷积矩阵,然后我们就得到了

 （7.21）

这意味着卷积信号 在每个节点u处的x[u]将对应于节点的N-Hop邻域中信息的某些混合，并具有α0，...，αn控制来自不同跳的信息强度的术语。

我们可以很容易地将图卷积的概念推广到高维节点特征。 如果我们有一个节点特征矩阵XR|V|×m然后，我们可以类似地应用卷积滤波器作为

 （7.22）

从信号处理的角度来看，我们可以将节点特征的不同维度看作是不同的“通道”。

**图卷积和消息传递GNN**

方程（7.22）还揭示了我们在第5章中引入的消息传递GNN模型与图卷积之间的联系。 例如，在基本的GNN方法（参见方程6.5）中，每一层消息传递基本上对应于简单卷积滤波器的应用。

 （7.23）

结合一些可学习的权重矩阵和非线性。 一般来说，传递GNN体系结构的每一层消息聚合来自节点的本地邻域的信息，并将这些信息与节点的当前表示相结合（参见方程5.4）。我们可以将这些消息传递层看作是方程（7.23）中简单线性滤波器的推广，其中我们使用了更复杂的非线性函数。 此外，通过叠加多个消息传递层，GNNs能够隐式地对邻接矩阵的高阶多项式进行操作。

**邻接矩阵，拉普拉斯，还是归一化变体？**

方程（7.22）定义了卷积矩阵Qh 对于任意图作为邻接矩阵的多项式。 这样定义Qh就保证了我们的滤波器与邻接矩阵交互，满足平移等方差的广义概念。然而，在一般情况下，与邻接矩阵(即平移等方差)的交换性并不一定意味着与拉普拉斯L = D-A(或其任何标准化变体)的交换性。在这个链图的特殊情况下，我们能够定义过滤矩阵Qh同时与A和L交换，但是对于更一般的图，我们可以选择是根据邻接矩阵还是Laplacian的一些版本来定义卷积。一般来说，在这种情况下没有“正确”的决定，根据所做的选择，可能存在经验上的权衡。理解这些权衡的理论基础是一个开放的研究领域[Ortega等人2018]。

在实际应用中，研究人员经常使用对称归一化拉普拉斯算子或者对称归一化邻接矩阵定义卷积过滤器。对于为什么这些对称标准化矩阵是可取的有两个原因。首先，这两个矩阵都有界谱，这使它们具有理想的数值稳定性。除此之外，可能更加重要的是这两个矩阵同时可对角化，这就意味着它们共享相同的特征向量。事实上，我们可以很容易地验证它们的特征分解之间存在一种简单的关系，因为



 (7.24)

其中U是特征向量的共享集合，Λ是对角矩阵包含拉普拉斯特征值。这意味着基于其中一个矩阵定义滤波器意味着与另一个矩阵的交换性，这是一个非常方便和理想的特性。

**7.1.3 光谱图合成**

我们已经看到了如何将信号和卷积的概念推广到图域。 我们通过类比离散卷积的一些重要性质（例如平移等差）来这样做，这一讨论使我们产生了将图卷积表示为邻接矩阵(或Laplacian)的多项式的想法). 然而，我们在上一小节中忽略的卷积的一个关键性质是卷积与卷积之间的关系傅里叶变换。在本节中，我们将考虑图上的谱卷积的概念，其中我们通过傅里叶变换的扩展构造图卷积。我们会看到这个光谱透视图恢复我们以前讨论过的许多相同的结果，同时也揭示了图卷积的一些更一般的概念。

**傅里叶变换和拉普拉斯算子**

为了激励傅里叶变换向图的推广，我们依赖于傅里叶变换和拉普拉斯（即差分）算子之间的联系。 我们以前看到了拉普拉斯算子的定义∆在一个简单的离散时变信号（方程7.10）的情况下，但是这个算子可以推广应用于任意光滑函数因为

 （7.25）

 （7.26）

该算子计算梯度f(X)的散度。 直观地，拉普拉斯算子告诉我们在一点的函数值和在这一点周围的函数值之间的平均差。

在离散时间设置中，拉普拉斯算子简单地对应于差分算子（即连续时间点之间的差值）。 在一般离散图的设置中，这个概念对应于拉普拉斯，因为根据定义

 （7.27）

它测量节点i处某些信号x[i]的值与其所有相邻的信号值之间的差异。 这样，我们就可以将Laplacian矩阵看作是Laplace算子的离散模拟，因为它允许我们量化节点上的值与该节点邻居上的值之间的差异。

现在，拉普拉斯算子的一个非常重要的性质是它的本征函数对应于复指数。 也就是说

 （7.28）

所以的特征函数是相同的复指数构成模式的频域傅里叶变换(即sinusoidal面波)，对应的特征值表示频率。事实上，我们甚至可以验证链图的循环拉普拉斯的特征向量为，其中。

**图的傅里叶变换**

拉普拉斯算子的特征函数与傅里叶变换之间的联系使我们能够将傅里叶变换推广到任意图。 特别是，我们可以通过考虑一般图Laplacian的eigendecomposition来推广傅里叶变换的概念：

 （7.29）

其中，我们将特征向量U定义为图傅里叶模，作为基于图的傅里叶模的概念。 假定矩阵Λ具有对应沿对角线的特征值，这些特征值提供了一个基于图的不同频率值的概念。换句话说，因为本征函数一般的Laplace算子对应于傅里叶模，即傅里叶级数中的复指数，我们根据图Laplacian的特征向量定义了一般图的傅里叶模。

因此，信号（或函数）的傅里叶变换在图表上可以被计算为

 （7.30）

并将其逆傅里叶变换计算为

****  （7.31）

在变换的傅里叶空间中，通过点积来定义谱域中的图卷积。 换句话说，给出了图傅里叶系数Ut 信号f的f以及图傅里叶系数Ut在一些滤波器**h**中，我们可以通过元素的乘积来计算一个图形的卷积

 （7.32）

其中U是LaplacianL的特征向量矩阵，我们使用了×g表示这个卷积是特定于图G的。

基于方程（7.32）我们可以根据函数h的傅里叶系数来表示频谱域中的卷积

 （7.33）

 （7.34）

其中diag(θh)是具有θ值的矩阵h在对角线上。然而，以这种非参数方式定义的滤波器对图的结构没有真正的依赖关系，并且可能不满足我们想要的许多属性卷积。 例如，这种过滤器可以是任意非本地的。

为保证光谱滤波器的θh对应于有意义的卷积在图上，一个自然的解决方案是参数化θh 基于拉普拉斯的特征值。 特别是，我们可以将光谱滤波器定义为pnΛ，因此它是Laplacian特征值的一个N多项式。 以这种方式定义光谱卷积可以确保我们的卷积与Laplacian通信，因为

 （7.35）

 （7.36）

此外，这一定义确保了局部性概念。 如果我们使用一个k度多项式，那么我们确保每个节点上的滤波信号取决于其k跳邻域中的信息。

因此，最后，从光谱的角度推导出图卷积，我们可以恢复图卷积可以表示为的关键思想 拉普拉斯的多项式（或其归一化变体之一）。 然而，光谱视角也揭示了更一般的策略来定义图上的卷积。

**将拉普拉斯特征向量解释为频率**

在斯坦达德傅里叶变换我们可以将傅里叶系数解释为对应于不同的频率。 在一般的图情况下，我们不能再用这种方式解释图傅里叶变换。 然而，我们仍然可以对高频和低频分量进行类比。特别是，我们可以回想一下拉普拉斯方程的特征向量解决了最小化问题

 （7.37）

由瑞利-里兹定理。 我们有这个

 (7.38)

通过第一章讨论的拉普拉斯的性质与事实表明，Laplacian的最小特征向量对应于图上从节点到节点变化最少的信号，第二个最小特征向量对应于变化第二个最小数量的信号，以此类推。 事实上，当我们执行谱聚类时，我们在第1章中利用了拉普拉斯特征向量的这些性质。 在这种情况下，我们证明了Laplacian特征向量可以用来向社区分配节点，从而最小化社区之间的边缘数。 我们现在可以从信号处理的角度来解释这个结果：拉普拉斯特征向量定义了在整个图中以平滑的方式变化的信号，最平滑的信号表示图的粗粒度社区结构。

**7.1.4卷积-旋转GNN**

前面的小节将卷积的概念推广到图。 我们看到图上的基本卷积滤波器可以表示为（归一化）邻接矩阵或拉普拉斯的多项式。 我们看到了这一事实的空间和光谱动机，我们看到了如何利用光谱透视来定义基于图傅里叶变换的更一般形式的图卷积。 在本节中，我们将简要回顾如何基于这些连接开发和启发不同的GNN模型。

**纯粹的卷积方法**

关于GNNs的一些最早的工作可以直接映射到前面小节的图卷积定义。 这些方法的关键思想是，它们使用方程（7.34)或方程(7.35）来定义卷积层，并通过将多个卷积层与非线性叠加和组合来定义一个完整的模型。 例如，在早期工作中，Bruna等人。 [2014]实验了非参数谱滤波器（方程7.34)以及参数谱滤波器(方程7.35），其中它们定义了多项式pn （Λ）通过三次样条方法。 在这项工作之后，Defferrard等人。 [2016]基于方程7.35定义的卷积pn(L)使用Chebyshev多项式。这种方法得益于Chebyshev多项式有一个有效的递归公式，并且具有各种形式，使它们适合于多项式近似的性质[Mason和Handscomb，2002]。在相关的方法中，廖等人。[2019b]根据Lanczos算法学习Laplacian的多项式。

还有一些方法超越了拉普拉斯的实值多项式（或邻接矩阵）。 例如，Levie等人。 [2018]考虑Laplacian和Bianchi等人的Cayley多项式。[2019]考虑ARMA过滤器。这两种方法都采用了更一般的参数有理com-拉普拉斯函数（或邻接矩阵）。

**图卷积网络和与消息传递的连接**

在他们的开创性工作中，Kipf和Welling[2016a]建立了图卷积的概念，以定义最流行的GNN体系结构之一，通常称为图卷积网络(GCN)。 GCN方法的关键洞察力是，我们可以通过叠加非常简单的图卷积层来建立强大的模型。 在Kipf和Welling[2016a]中定义了一个基本的GCN层

 （7.39）

其中是邻接的归一化变体， 矩阵（带有自循环）和W(k) 是一个可学习的参数矩阵。 该模型最初是作为一个简单的图卷积(基于多项式I+A)的组合，具有可学习的权重矩阵和非线性。 正如第5章所讨论的，我们还可以将GCN模型解释为基本GNN消息传递方法的变体。 一般来说，如果我们考虑结合一个简单的图卷积定义的多项式I+A与非线性和可训练的权重矩阵，我们恢复了基本的GNN：

 （7.40）

换句话说，基于I+A的简单图卷积相当于从邻居那里聚合信息，并将其从节点本身与信息相结合。 因此，我们可以将消息传递的概念看作是对应于一个简单的图形卷积形式，并结合附加的可训练权重和非线性。

**作为低通卷积滤波器的过平滑** 在第五章中，我们介绍了GNN中的过平滑问题。在过度平滑中，直觉的想法是，经过过多轮的消息传递，所有节点的嵌入开始看起来相同，并且相对不具信息性。 基于消息传递GNNs与图卷积之间的联系，我们现在可以从图信号处理的角度来理解过度平滑。

关键的直觉是，在基本GNN中叠加多轮消息传递类似于应用低通滤波器，该滤波器在图上产生输入信号的平滑版本。 特别是，假设我们将基本GNN（方程7.40）简化为以下更新方程：

 （7.41）

与方程（7.40）中的基本GNN相比，我们通过消除非线性和消除在每个消息传递步骤中添加“自”嵌入来简化了模型。 为了数学上的简单性和数值稳定性，我们还假设我们使用对称指标规范化邻接矩阵而不是非规范邻接矩阵。该模型类似于简单的GCN在Kipf和Welling[2016a]中提出的方法，基本上等于在每一轮消息传递时对邻居嵌入的平均值。现在，很容易看出，在基于方程（7.41）的K轮消息传递之后，我们将得到一个依赖于邻接矩阵的Kth幂的表示：

 （7.42）

其中W是一些线性算子，X是输入节点特征的矩阵。为了理解过度平滑和卷积滤波器之间的联系，我们只需要识别乘法输入的节点特征由邻接矩阵的高功率解释为基于图Laplacian的最低频率信号的卷积滤波器。

例如，假设我们使用足够大的K值，使我们达到以下递归的固定点;

 (7.43)

当使用归一化邻接矩阵时，可以验证这个定点是可以实现的，因为的主导特征值等于一个。我们可以看到，在这个固定点上，所有的节点特征都会收敛到完全由的主导特征向量定义更一般地说，的更高权力将强调这个矩阵的最大特征值。 此外，我们知道的最大特征值对应于其对应的最小特征值，对称归一化LaplacianL (例如，见方程式7。24)。 这些事实加在一起意味着一个信号乘以的的高功率对应于a 基于L的最低特征值（或频率）的卷积滤波器赛姆也就是说，它产生一个低通滤波器！

因此，我们可以从这个简化的模型中看到，堆叠多轮消息传递会导致低通的卷积滤波器，在最坏的情况下，这些滤波器只是将所有节点表示收敛到图上连接组件内的常量值（即拉普拉斯的“零频率）。

当然，在实践中，我们使用更复杂的消息传递形式，通过将每个节点以前的嵌入包含在消息传递更新步骤中，这个问题得到了部分缓解。 尽管如此，理解如何以天真的方式将“更深”的卷积叠加在图上实际上可以导致更简单，而不是更复杂的卷积滤波器是有指导意义的。

**没有消息传递的GNN**

受图卷积连接的启发，最近的几项工作也提出了通过删除迭代消息传递过程来简化GNN。 在这些方法中，模型通常被定义为

 （7.44）

其中 是邻接矩阵A的一些确定性函数，MLP表示密集神经网络，是输入节点特征的矩阵，是学习节点表示的矩阵。例如，在Wu等人中[2019]，他们定义

 （7.45）

其中是对称归一化邻接矩阵（添加自循环）。在一项密切相关的工作中，Klicpera等人[2019]通过类比将f定义为个性化页面排名算法**（注释5）**

** （7.46）**

****  （7.47）

**（注释5，注意，方程（7.46)和(7.47）之间的等式要求(IαA)的主导特征值在1以上有界。 在实践中，Klicpera等人[2019]使用幂迭代来逼近方程（7.46)中的反演）**

这些方法背后的直觉是，我们通常不需要将可训练神经网络与图卷积层交织在一起。 相反，我们可以简单地使用神经网络来学习模型开始和结束时的特征变换，并应用确定性卷积层来利用图结构。这些简单的模型能够在许多分类基准上优于更多参数化消息传递模型(例如GATS或图形SAGE。

还有越来越多的证据表明，使用具有自环的对称归一化邻接矩阵会导致有效的图卷积，特别是在这种不通过消息的简化设置中。 吴等人。[2019]和Klicpera等人。[2019]发现基于A的卷积˜取得了最佳的经验绩效。Wu等人。[2019]也为这些结果提供了理论支持。它们证明了通过减小占优特征值的大小来缩小相应图拉普拉斯的谱。 直观地说，添加自循环减少了远距离节点的影响，使滤波后的信号更依赖于图上的局部邻域。

**7.2GNN和概率图形模型**

GNN作为图形结构数据卷积的扩展，被很好地理解和激励。 然而，GNN框架有其他的理论动机，可以提供有趣和新颖的观点。 一个突出的例子是基于与概率图形模型(PGMs)中变分推理的连接的GNNs的动机)。

在这个概率视角下，我们查看嵌入，对于每个节点作为我们试图推断的潜在变量。我们假设我们观察图结构(即邻接矩阵A)和输入节点特征X和我们的目标是推断潜在变量(即嵌入zv )可以解释这些观测数据。然后，作为GNNs基础的消息传递操作可以看作是某些消息传递算法的神经网络模拟，这些算法通常用于变分推理，以推断潜在变量上的分布。 Dai等人首先指出了这种联系。[2016]，目前的大部分讨论都是基于他们的工作。

请注意，本节的介绍假设了PGMs的大量背景，我们推荐Wainwright和Jordan[2008]作为感兴趣的读者的好资源。然而，我们希望并期望即使是一个不了解PGMS的读者也能从下面的讨论中收集有用的见解。

**7.2.1希尔伯特空间嵌入分布**

为了理解GNNs与概率推理之间的联系，我们首先（简要）介绍了Hilbert空间中嵌入分布的概念[Smola等人，2007年]。设p(X)表示定义的概率密度函数在随机变量. 给定任意（可能是无限维）特征映射在此特征映射下，我们可以根据其期望值来表示密度p(X)：

 (7.48)

用Hilbert空间嵌入分布的关键思想是，方程（7.48）将是内射的，只要使用合适的特征映射φ。 这意味着µx可以作为p(X)的一个足够的统计量，我们想要在p(X)上执行的任何计算都可以等价地表示为of的函数嵌入µx. 一个著名的特征映射的例子，将保证这种内射性质是由高斯径向基函数(RBF)核诱导的特征映射[Smola等人，2007年]。

对分布的希尔伯特空间嵌入的研究是一个丰富的统计领域。 然而，在与GNNs连接的上下文中，关键的提取是，我们可以将分布p(X)表示为嵌入µx 在一些特征空间。 我们将使用这个概念来激励GNN消息传递算法，作为学习表示节点锁存器p(z)上分布的嵌入的一种方法v).

**7.2.2图形作为图形模型**

取图数据的概率视图，我们可以假设给出的图结构定义了不同节点之间的依赖关系。 当然，我们通常用这种方式来解释图形数据。 连接的节点在图中，通常假定以某种方式相关。 然而，在概率设置中，我们以一种形式的概率方式看待节点之间依赖的概念。

准确地说，我们说图G=(V，E)定义了马尔可夫随机场：

 (7.49)

其中Φ和Ψ是非负势函数，我们使用{xv}作为集合{x的速记v∀v∈V}。方程（7.49）表示分布p({xv}，{zv })节点特征和节点嵌入根据图结构进行分解。 直观地说，Φ(xv ，zv )表示节点特征向量x的可能性v 给定其潜在节点嵌入zv 而Ψ控制连接节点之间的依赖关系。 因此，我们假设节点特征是由它们的潜在嵌入决定的，并且我们假设连接节点的潜在嵌入是相互依赖的（例如，连接节点可能具有嵌入）。

在标准概率建模设置中，Φ和Ψ通常被定义为基于领域知识的参数函数，最常见的是，这些函数被假定来自指数家族，以确保可处理性[Wainwright和Jordan，2008]。然而，在我们的演讲中，我们是不可知论Φ和Ψ的确切形式，我们将通过利用上一节中讨论的Hilbert空间嵌入思想来隐式学习这些函数。

**7.2.3嵌入平均场推理**

给定方程（7.49）定义的马尔可夫随机场，我们的目标是推断潜在嵌入p(Z)的分布v对于所有节点vV，同时隐式学习潜在的函数Φ和Ψ。更直观地说，我们的目标是推断图中所有节点的潜在表示，这些节点可以解释观察到的节点特征之间的依赖关系。

为了做到这一点，一个关键步骤是计算后验等，计算给定观测特征的一组特定潜在嵌入的可能性。 一般来说，计算这个后验是计算困难的-即使Φ和Ψ是已知的和明确定义的-所以我们必须求助于近似方法。

一种流行的方法-我们将在这里使用-是采用平均场变分推理，其中我们使用一些函数q近似后验v基于假设：

 （7.50）

每个Q的位置v是一个有效的密度。平均场推理的关键直觉是，我们假设潜在变量上的后验分布分解为V个独立分布，每个节点一个。

以获得近似qv在平均场近似中是最优的函数，标准方法是最小化近似后验和真实后验之间的Kullback-Leibler(KL)散度：

 （7.51）

KL散度是测量概率分布之间距离的一种典型方法，因此求Qv 最小化方程（7.51）的函数给出了在平均场假设下尽可能接近真实后验的近似后验。当然，直接最小化方程（7.51）是不可能的，因为评估KL散度需要了解真正的后验。

然而，幸运的是，变分推理的技术可以用来证明Qv(zv）最小化KL必须满足以下定点方程：

 （7.52）

我们可以通过对有效概率分布的一些初始猜测进行初始化并迭代计算来逼近这个不动点解

（7.53）

方程（7.52）背后的理由超出了本书的范围。然而，为了本书的目的，基本的想法如下：

1. 我们可以近似真实的后验p(zvxv 在潜在嵌入使用平均场假设，其中我们假设后验因式分解为|V|独立分布p({zv}|{xv}) ≈

2.在平均场假设下，由方程（7.52）中的固定点给出了最优近似，每个潜在节点嵌入的近似后验是(i)节点的特征和(ii)节点邻居嵌入的边缘分布的函数。

此时，与GNN的连接开始出现。 特别是，如果我们检查方程（7.53）中的定点迭代，我们就会看到更新的边际分布是节点特征xv的函数(通过潜在函数Φ)以及来自上一次迭代的邻域边缘集的函数(通过潜在函数)。这种形式的消息传递非常类似于在GNN中传递的消息！ 在每个步骤中，我们根据节点邻域中的值更新集合每个节点的值。 关键的区别在于平均场消息传递方程在分布上工作，而不是嵌入，这些分布用于标准GNN消息传递。

我们可以利用我们在7.2.1节中介绍的希尔伯特空间嵌入，使GNNs和平均场推理之间的连接更加紧密。假设我们有一些内射特征映射φ，可以表示所有的边缘作为嵌入

 （7.54）

有了这些表示，我们可以重写方程（7.52）中的定点迭代

 （7.55）

其中f是向量值函数。请注意，f聚合了来自相邻嵌入集（即）的信息以及使用此聚合数据更新节点的当前表示（即）。这样，我们就可以看到，嵌入式平均场推理正好对应于一种通过图形传递的神经消息形式！

现在，在通常的概率建模场景中，我们将使用一些领域知识定义潜在的函数Φ和Ψ，以及特征映射φ。 并给出了一些Φ、Ψ和ψ，然后我们可以尝试解析地导出方程（7.55）中的f函数，这将允许我们使用一个嵌入版本的平均场推理。 然而，作为一种选择，我们可以简单地尝试学习嵌入µv 以端到端的方式使用一些监督信号，我们可以将f定义为一个任意的神经网络。 换句话说，我们不能简单地指定具体的概率模型

学习嵌入µv这可能对应于一些概率模型。 基于这个想法，戴等人[2016]以类似于基本GNN的方式定义f

 （7.56）

因此，在每次迭代中，节点v的更新Hilbert空间嵌入是其邻居嵌入及其特征输入的函数。并且与基本GNN一样，更新过程中的参数和可以通过梯度下降对任意任务损失进行训练。

**7.2.4更普遍的是GNN和PGM**

在上一小节中，我们简要介绍了如何将基本GNN模型导出为一种嵌入形式的平均场推理-这是DAI等人首先概述的连接。[2016]. 然而，有进一步的方法来连接PGM和GNN。 例如，可以根据不同的近似推理算法(例如Dai等人讨论的Bethe近似)导出消息传递的不同变体。 [2016])，也有几项工作探索GNN如何更普遍地集成到PGM模型中[Qu等人，2019年，Zhang等人，2020年]。 一般来说，GNNs与更传统的统计关系学习之间的联系是一个丰富的领域，具有巨大的新发展潜力。

**7.3GNN和图形同构**

我们现在已经看到了如何基于与图形信号处理和概率图形模型的连接来激励GNN。 在本节中，我们将把注意力转向我们关于GNNs的第三个也是最后一个理论观点：基于与图同构测试的连接的GNNs的动机。

与前面的章节一样，这里我们将再次看到基本GNN如何作为现有算法的神经网络变体-在这种情况下，Weisfieler-Lehman(WL)同构算法。 然而，除了激励GNN方法之外，与同构测试的连接也将为我们提供工具，以正式的方式分析GNN的功率。

**7.3.1图同构**

图同构的检验是图论中最基本、研究最充分的任务之一。 给定一对图形G1还有G2图同构测试的目的是声明这两个图是否是同构的。在直观意义上，两个图是同构的，这意味着它们本质上是相同的。同构图表示完全相同的图结构，但它们可能只在相应的邻接矩阵中节点的排序上有所不同。 形式上，如果我们有两个图与邻接矩阵A1 和A2 以及节点特征X1 和X2我们说，两个图是同构的，当且仅当存在一个置换矩阵P，这样

 (7.57)

必须注意的是，同构图在其基本结构上确实是相同的。 邻接矩阵中节点的排序是我们使用代数对象（例如矩阵）表示图时必须作出的任意决定，但这种排序对基础图本身的结构没有影响。

尽管它的定义很简单，但对图同构的测试是一个根本的难题。例如，测试同构的天真方法将涉及以下优化问题;

 (7.58)

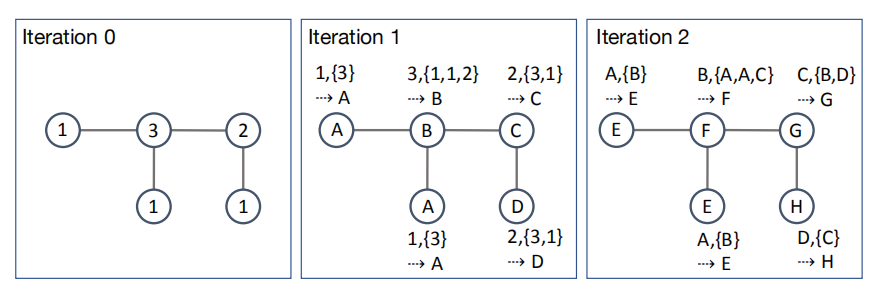
这种优化需要对整个置换矩阵进行搜索，以评估是否存在一个单一的置换矩阵P，从而导致两个图之间的等价性。这种天真的方去在上计算复杂度是巨大的!事实上，目前还没有多项式时间算法能正确地检验一般图的同构性。

图同构测试被正式称为NP-不确定(NPI)。 已知不是NP完全，但对于该问题没有一般多项式时间算法。 (整数分解是另一个众所周知的问题，被怀疑属于NPI类。) 然而，有许多实用的图同构测试算法在广泛的类图上工作，包括我们在第1章中简要介绍的WL算法。

**7.3.2图同构和表示能力**

图同构测试理论对于图表示学习特别有用。它给了我们一种方法来量化不同学习方法的代表性能力。 如果我们有一个算法，例如GNN，可以生成表示对于图的，我们可以通过测试这些表示在测试图同构中有多有用来量化这个学习算法的能力。特别是，给定学习表示 和 对于两个图，一个“完美”的学习算法将有这个：

当且仅当 和 图同构时， = （7.59）



**图7.2：一个图上WL迭代标记过程的示例**

一个完美的学习算法将生成两个图的相同嵌入，当且仅当这两个图实际上是同构的。

当然，在实践中，没有一种表示学习算法是“完美的”(除非P=NP)。 然而，通过将表示学习算法连接到图同构测试来量化它的功率是非常有用的。 尽管图同构测试在一般情况下是不能解决的，但我们确实知道几种用于近似同构测试的强大和理解良好的方法，并且我们可以通过将它们与这些方法进行比较来洞察GNN的功率。

**7.3.3维斯费勒-雷曼算法**

将GNN连接到图同构测试的最自然的方法是基于与Weisfieler-Lehman(WL)算法家族的连接。 在第一章中，我们讨论了图核上下文中的WL算法。 然而，WL方法更广泛地被称为近似同构测试的最成功和最受理解的框架之一。 最简单的WL算法版本-通常称为1-WL-包含以下步骤:

1. 给定两个图G1还有G2 ,我们分配了一个初始标签给每个图中的每个节点。 在大多数图中，这个标签只是节点度，即，但如果我们有离散特征(即一个热特征xv)与节点相关联，然后我们可以使用这些特性来定义初始标签。

2.接下来，我们迭代地为每个图中的每个节点分配一个新的标签，方法是将节点邻域内当前标签的多组散列，以及节点的当前标签：

 (7.60)

其中双括号用于表示多集和HASH函数将每个唯一的多集映射到一个唯一的新标签。

3.我们重复步骤2，直到两个图中所有节点的标签收敛，即直到我们到达迭代K，其中



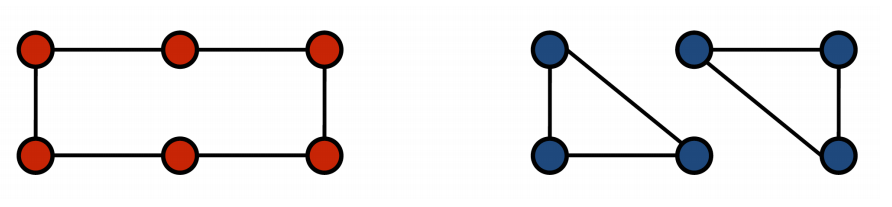
4. 最后，我们构造了多集



总结每个图中的所有节点标签，并声明G1还有G2图同构当且仅当两个图的多集相同时，即当且仅当LG1 =LG2 .

图7.2说明了一个图上WL标记过程的示例。 在每次迭代时，每个节点都收集其本地邻域中的多组标签，并基于该多组更新自己的标签。 在这个标记过程的K迭代之后，每个节点都有一个标签来总结其K跳邻域的结构，这些标签的集合可以用来描述整个图或子图的结构。

已知WL算法在大多数迭代中收敛，并已知成功地测试了广义图的同构[Babai和Kucera，1979]。但是，有一些众所周知的测试失败的案例，如图7.3所示的简单示例。



**图7.3：用基本WL算法无法区分的两个图来示例**

**7.3.4GNN和WL算法**

在WL算法和神经消息传递GNN方法之间有明确的类比。 在这两种方法中，我们迭代地聚合来自本地节点邻域的信息，并使用这些聚合信息来更新每个节点的表示。 这两种方法的关键区别在于WL算法聚合和更新离散标签（使用哈希函数），而GNN模型使用神经网络聚合和更新节点嵌入。 事实上，GNN是作为WL算法的连续和可微模拟的动机和导出的。

在下面的定理中，GNNs与WL算法之间的关系（在7.3.3节中描述）可以形式化：

**定理4**([Morris等人，2019年，徐等人，2019年])将消息传递GNN(MP-GNN)定义为由以下形式的K消息传递层组成的任何GNN：

 （7.61）

其中AGGREGATE是一个可微置换不变函数，UPDATE是一个可微函数。 此外，假设我们只有离散特征输入到初始层，即。然后我们就有了 只有当节点u和v在经WL算法K次迭代后有不同的标签时。

在直观方面，定理7.3.4指出，当我们有离散信息作为节点特征时，GNNs不比WL算法强大。如果WL算法将相同的标签分配给两个节点，那么任何消息传递GNN时还将向这两个节点分配相同的嵌入。 节点标记的这一结果也扩展到同构测试。 如果WL测试不能区分两个图，那么MP-GNN也不能区分这两个图。 我们也可以在另一个方向上显示出更积极的结果：

**定理5**([Morris等人，2019年，徐等人，2019年])存在一个MP-GNN，使得 当且仅当两个节点u和v经过WL算法的K次迭代后具有相同的标签。

该定理表明存在与WL测试一样强大的消息传递GNN。

**哪些MP-GNNs最强大？** 以上状态的两个定理说明消息传递GNN最多和WL算法一样强大，并且存在与WL算法一样强大的消息传递GNN。 那么，哪些GNNs实际上获得了这个理论上界？ 有趣的是，我们在第5章开头介绍的基本GNN足以满足这一理论。特别是，如果我们定义消息传递更新如下：

 （7.62）

然后，这个GNN足以匹配WL算法的功率[Morris等人，2019年]。

然而，第5章中讨论的大多数其他GNN模型都不如WL算法强大。 从形式上讲，要像WL算法一样强大，AGGREGATE和UPDATE函数需要是内射的[Xu等人，2019年]。 这意味着AGGREGATE和UPDATE操作符需要将每个唯一的输入映射到一个唯一的输出值，对于我们讨论的许多模型来说，情况并非如此。 例如，使用邻居嵌入的（加权）平均值的AGGREGATE函数不是内射；如果所有邻居都有相同的嵌入，那么a（加权）平均值将无法区分不同大小的输入集。

徐等人。[2019]详细讨论了各种GNN体系结构的相对功率。 他们还定义了一个“最小”GNN模型，它的参数很少，但仍然像WL算法一样强大。他们把这个模型称为图形同构网络(GIN)，它由以下更新定义：

 （7.63）

在这里 是一个可训练的参数。

**7.3.5超越WL算法**

上一小节强调了消息传递GNNs(MP-GNNs)的一个重要的负面结果：这些模型不比WL算法强大。然而，尽管这一负面结果，调查我们是如何可以使GNN比WL算法更强大，是一个活跃的研究领域。

**关系池**

一种激励更强大GNN的方法是考虑WL算法的失败案例。 例如，我们可以在图7.3中看到，WL算法-因此所有MP-GNN-无法区分连接的6周期和一组两个三角形。 从消息传递的角度来看，这种限制源于AGGREGATE和UPDATE操作无法检测两个节点何时共享邻居。 在图中的示例中7.3，每个节点可以从消息传递操作推断它们有两个度-2邻居，但这些信息不足以检测节点的邻居是否相互连接。 这一限制不仅仅是图7.3所示的一个角状情况。 消息传递方法通常无法识别图中的封闭三角形，这是一个关键的限制。

为了解决这一限制，Murphy等人。 [2019]考虑增加具有唯一节点ID特征的MPGNN，如果我们使用MP-GNN(A，X)来表示输入邻接矩阵A和节点特征X上的任意MP-GNN. 添加节点ID相当于将MP-GNN修改为以下内容：

 （7.64）

其中I是dd维恒等矩阵，表示列向矩阵级联。换句话说，我们简单地为每个节点添加一个唯一的、一热的指示器特性。在图7.3的情况下，添加唯一的节点ID将允许MP-GNN识别两个节点何时共享邻居，这将使这两个图是可区分的。

然而，不幸的是，这种添加节点ID的想法并不能解决这个问题。 事实上，通过添加唯一的节点ID，我们实际上引入了一个新的、同样有问题的问题：MP-GNN不再是等变量的置换。 对于一个标准的MP-GNN，我们有

 （7.65）

其中是任意排列矩阵。 这意味着标准MP-GNN是置换等变量。 如果我们对邻接矩阵和节点特征进行置换，那么得到的节点嵌入就会以等价的方式进行置换。 然而，具有节点ID的MP-GNN并不是置换不变的，因为通常情况下

 （7.66）

关键问题是，给每个节点分配一个唯一的ID，会为图修复一个特定的节点排序，这破坏了排列的等差性。

为了缓解这个问题，Murphy等人。 [2019]提出了关系池(RP)方法，该方法涉及在所有可能的节点排列上边缘化。 给定任何MP-GNN，此GNN的RP扩展由

 （7.67）

总结所有可能的置换矩阵恢复置换不变性，我们保留了添加唯一节点ID的额外表示能力。 事实上，墨菲等人。 [2019]证明了MPGNN的RP扩展可以区分WL算法无法区分的图。

RP方法的局限性在于其计算复杂度。 朴素评价方程（7.67）的时间复杂度为，这在实践中是不可行的 然而，尽管有这一限制，Murphy等人，[2019]表明，RP方法可以通过各种近似来降低计算成本（例如，采样排列的子集)，从而获得较强的结果。

**k-WL 测试 and k-GNNs**

上面讨论的关系池(RP)方法可以产生GNN模型，比WL算法强大得多。 然而，RP方法有两个关键限制：

1.整个算法在计算上是难以解决的。

2. 我们知道RP-GNN比WL测试更强大，但我们没有办法描述它们有多强大。

为了解决这些局限性，有几种方法考虑了通过调整WL算法的推广来改进GNN。

我们在7.3.3节中介绍的WL算法实际上只是k-WL算法家族中最简单的算法。 事实上，我们前面介绍的WL算法通常被称为1-WL算法，它可以推广到k>1的k-WL算法。 k-WL算法背后的关键思想是，我们标记大小为k的子图，而不是单个的子图节点。 该k-WL算法通过以下步骤生成图形G的表示：

1.设是一个元组，它定义一个大小为k的子图，其中。通过子图的同构类定义每个子图的初始标签，(即，当且仅当两个子图同构时，它们才有相同的标号)。

2.接下来，我们迭代地为每个子图分配一个新的标签，方法是在这个子图的邻域内散列当前标签的多集



其中j代子图邻域被定义为

 （7.68）

3. 我们重复步骤2，直到所有子图的标签收敛，即直到我们到达迭代K，其中每个k元组的节点

4. 最后，我们构造了一个多集合



总结图中的所有子图标签。

与1-WL算法一样，总结Lg 由k-WL算法生成的多集可以通过比较两个图的多集来检验图的同构性。 还有基于k-WL检验的图核方法[Morris等人，2019年]，类似于第一章中介绍的WL-Kernel。

关于k-WL算法的一个重要事实是它引入了表示容量的层次结构。对于任何，我们都认为（k+1）-WL测试严格地比k-WL测试更强大。**（注释6）** 因此，一个自然的问题是，我们是否可以设计与k>2的k-WL测试一样强大的GNN，当然，一个自然的设计原则是通过类比k-WL算法来设计GNN。

Morris等人[2019]试图做到这一点：他们开发了一种k-GNN方法，它是k-WL算法的可微和连续模拟。k-GNN学习与子图相关的嵌入-而不是节点-并且消息传递是根据子图邻域发生的（例如，如方程7.68中定义的）。Morris等人。[2019]证明了k-GNN可以像k-WL算法一样具有表现力。然而，对于k-WL测试和k-GNNs也有严重的计算问题，因为消息传递的时间复杂度随着k的增加而组合爆炸。这些计算问题需要各种近似才能使k-GNN在实践中易于处理[Morris等人，2019年]。

**(注释6然而，请注意，运行k-WL需要解决大小为k的图的图同构，因为k-WL算法中的步骤1需要根据它们的同构类型标记图。因此，为k>3运行k-WL通常是计算上难以解决的。)**

**不变和等变k阶GNN**

另一项工作是由构建与k-WL测试一样强大的GNNs的想法驱动的，这是Maron等人提出的不变量和等变量GNNs。 [2019]. 消息传递GNNs(MP-GNNs；如定理7.3.4所定义的)的一个关键方面是它们与节点排列相等，这意味着

 （7.69）

对于任何置换矩阵。 这个等式表示，将输入置换到MP-GNN只会导致以类似的方式置换输出节点嵌入的矩阵。

除了这种等差的概念外，我们还可以在图级上定义MP-GNN的排列不变性的类似概念。 特别是，MPGNN可以用一个函数来扩展（见第5章），它将学习的节点嵌入矩阵映射到整个图的嵌入。 在这个图级设置中，我们有MP-GNN是置换不变的，即

 （7.70）

这意味着当使用不同的节点顺序时，集合图级嵌入不会改变。

基于不变性和等变性的思想，Maron等人。[2019]提出一种基于置换等变量/不变量的张量运算的类GNN模型的一般形式。 假设我们有一个阶(k+1)张量，在这里我们假设这个张量的第一个k通道/模式是由图的节点索引的。 我们使用表示法来表示根据节点排列矩阵P对这个张量的第一个k通道进行排列的操作。然后，我们可以将线性等变层定义为线性算子（即张量）：

 （7.71）

其中我们用×表示广义张量积。 不变线性算子可以类似地定义为满足以下等式的张量L：

 （7.72）

请注意，等变线性算子和不变线性算子都可以表示为张量，但它们具有不同的结构。 特别是，一个等变量操作对应于张量，它具有由节点索引的2k通道（即，节点通道的数量是输入的两倍）。另一方面，一个不变算子对应于张量 它有由节点索引的k个通道（即与输入相同的数字）。有趣的是，从线性算子的这个张量观点来看，等价（方程7.71)和不变(方程7.72）性质可以组合成一个单一的要求，即张量在节点排列下是一个定点：

 （7.73）

换句话说，对于给定的输入，这个输入上的等变量和不变量线性算子将对应于满足方程（7.73）中的固定点的张量，但张量中的通道数将因它是等变算子还是不变算子而不同。

Maron等人。[2019]表明，满足方程（7.73）中固定点的张量可以构造为一组固定基元的线性组合。 特别是，任何满足方程（7.73）的阶l张量都可以写成

 （7.74）

在这里是一组固定基张量，是实值权值，b()是次贝尔数。 这些基张量的构造和推导在数学上涉及，与贝尔数理论密切相关来自组合学。 然而，一个关键的事实和挑战是，所需的基张数随着次Bell数的增加而增加，这是一个指数增长的序列。

使用这些线性等变层和不变层，Maron等人[2019]根据以下函数的组成来定义它们的不变k阶GNN模型：

 （7.75）

在这个组成中，我们应用m个等变线性层，其中和最大。在每一个线性等变层之间有一个表示为的元素非线性被应用。构图中倒数第二个函数，是不变的线性层，然后是多层感知器(MLP)作为组成中的最终函数。k阶不变GNN的输入是张量,其中前两个通道对应于邻接矩阵和其余通道编码初始节点特征/标签。

这种方法被称为k阶，因为等变线性层涉及多达k个不同通道的张量。然而，最重要的是，Maron等人[2019]证明了方程7.75后面的k阶模型与k-WL算法同样强大。 然而，与上一节讨论的k-GNN一样，为k>3构建k阶不变模型通常是计算上难以解决的。

**Chapter 9**

# 深度生成模型

上一章中讨论的传统图生成方法在许多设置中非常有用。它们可用于高效生成具有某些属性的合成图，并且它们可用于让我们深入了解某些图结构在现实世界中是如何产生的。然而，这些传统方法的一个最主要的限制在于它们依赖于固定的手工生成过程。简而言之，传统方法可以生成图，但它们缺乏从数据中学习生成模型的能力。

本章将介绍各种方法来准确应对这一挑战。这些方法将寻求学习基于一组训练图的图生成模型。这些方法避免将特定属性（如社区结构或度分布）手动编码为生成模型。相反，这些方法的目标是设计模型，这些模型可以观察一组图，并学会生成具有与该训练集相似特征的图。

我们将介绍一系列图的基本深度生成模型。这些模型将用三种最流行的方法来构建一般深层生成模型：变异自动编码器 （VAEs）、生成对抗网络（GANs）和自动回归模型。我们将重点介绍这些简单的模型和一些变体，强调high-level details，并在必要时提供文献支持。此外，虽然原则上这些生成技术可以彼此结合（例如VAEs通常与自动回归方法相结合）但在这里我们不会详细讨论这种组合。相反，我们将首先讨论图的基本VAE模型，其中我们寻求以自动编码器样式生成整个图。接下来，我们将讨论如何使用基于GAN的目标来代替变化损失，但仍要在一次生成图的情况下进行。这些一次生成模型类似于上一章中的ER和SBM生成模型，因为我们同时对图中的所有边进行采样。最后，本章最后将对自动回归方法进行讨论，该方法允许逐步生成图而不是一次生成图（例如按节点生成图形）。这些自动回归方法与上一章中的优先附件模型具有相似之处，因为在生成过程中的每个步骤添加某条边的概率取决于先前向图中添加了哪些边。

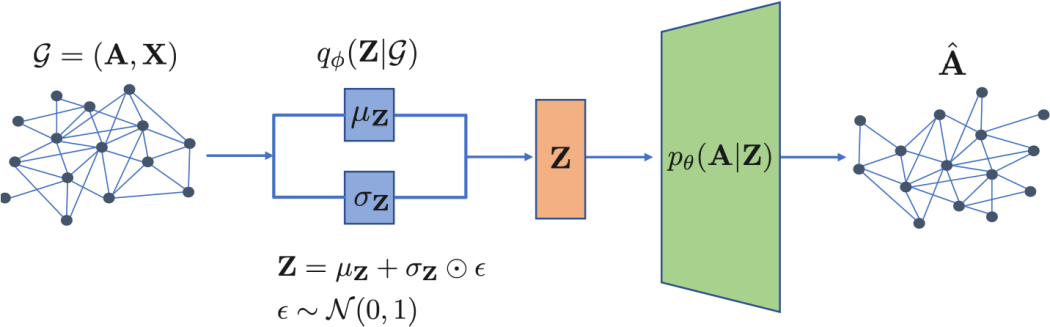


图 9.1: 应用于图设置的标准VAE模型的图示。编码神经网络将输入图映射成潜在变量**Z**上的后验分布。给定这个后验的样本解码模型pθ(**A∣Z**)试图重建邻接矩阵。

为简单起见，我们讨论的所有方法将只侧重于生成图结构（即邻接矩阵），而不是生成节点或边要素。本章假定基本熟悉VAEs、GANs和自动回归生成模型，例如基于LSTM的语言模型。我们推荐读者去阅读Goodfellow et al.[2016]，了解这些领域的背景知识。

在本书的所有topic中，图的深层生成模型在技术上涉及最多，在开发中也是最新生的。因此，本章的目标就是介绍启发这一领域早期研究的关键方法框架，同时重点介绍一些有影响力的模型。因此我们总是避免low-level details而倾向于对该子区域进行high-level的浏览。

## 9.1可变自动编码器方法

可变自动编码器(VAEs)是开发深层生成模型的最流行的方法之一[Kingma and Welling, 2013]。VAEs的理论和动机深深扎根于变分推理的统计领域，我们在第7章中对此进行了简要介绍。但是，出于本书的目的，将VAEs应用于图形的关键思想可以总结如下(图 9.1):我们的目标是训练概率解码器模型pθ(**A**|**Z**)，我们可以从中通过调整潜在变量**Z**采样逼真的图形（即邻接矩阵）. 在概率意义上，我们的目标是学习邻接矩阵的条件分布（分布开始以某些潜在变量为条件）。

为了训练VAE，我们将概率解码器与概率编码器模型相结合。此编码器模型将输入图 映射到潜在变量Z上的后验分布。我们的想法是，我们共同训练编码器和解码器，以使解码器能够在给定从编码器采样的潜变量**Z**∼的情况下重建训练图。然后，经过训练，我们可以通过从一些（无条件的）先验分布中采样潜在变量**Z**∼并将这些采样的潜在信号馈送到解码器，来丢弃编码器并生成新图。

在更正式和数学上的细节上，要为图构建VAE，我们必须指定以下关键组件：

1. 概率编码器模型qφ。对于图，概率编码器模型采用一个图作为输入，然后根据此输入，qφ 定义潜在表示上的分布qφ(**Z**|G)。通常，在VAE中，使用具有高斯随机变量的重新参数化技巧来设计概率qφ函数。也就是说，我们将潜在条件分布指定为 ，其中µφ和σφ是生成正态分布的均值和方差参数的神经网络，然后对潜在嵌入Z进行采样。
2. 概率解码器模型pθ。解码器采取潜在的表示**Z**作为输入并使用此输入指定图上的条件分布。在本章中，我们假设pθ定义了邻接矩阵项的条件分布，也就是我们可以计算。
3. 潜在空间上的先验分布p(**Z**)。在本章，我们将采用通常用于VAEs的标准高斯先验。

给定这些组件和一组训练图，我们可以通过最小化证据似然下限（ELBO）来训练VAE模型：

基本思想是我们试图最大化解码器的重构能力，即似然项 ，同时最小化我们后验潜在分布 和先验 p(**Z**)之间的KL散度。

ELBO损耗函数背后的动机是可变推理理论的根本[Wainwright and Jordan, 2008]。但是，我们想要在潜在表示上生成分布，以便满足以下两个（冲突的）目标：

1. 采样的潜在表示对足够的信息进行编码，以使我们的解码器可以重建输入。
2. 潜在分布尽可能接近先前分布。

第一个目标是当我们将训练图作为输入时，确保我们学会从编码的潜在表示中解码有意义的图。第二个目标充当正则化器，并确保即使从先验p(Z)的采样潜在表示时，我们也可以解码有意义的图。如果我们想要在培训后生成新的图形，那么第二个目标至关重要：我们可以通过从之前的采样和将这些潜在的嵌入馈送至解码器来生成新的图形，并且此过程只有在满足第二个目标的情况下才能有效。

在以下各节中，我们将描述可通过两种不同方式实例化图形的VAE想法。这些方法在定义编码器，解码器和潜在表示的方式上有所不同。但是，他们拥有使VAE模型适应图形的总体思路。

### 9.1.1节点级延迟

我们将研究的第一种方法是基于在第3章中介绍的基于节点嵌入的对图进行编码和解码的思想。该方法的关键思想是，编码器为图中的每个节点生成潜在表示。然后，解码器将成对的嵌入作为输入，并使用这些嵌入来预测在两个节点之间出现边缘的可能性。这个想法最初是由Kipf和Welling [2016b]提出的，被称为变图自动编码器（VGAE）。

##### 编码器模型

此设置中的编码器模型可以基于我们在第5章中讨论的任何GNN体系结构。特别是，给定一个邻接矩阵**A**和节点要素**X**作为输入，我们使用两个单独的GNN分别生成均值和方差参数，这些参数以以下输入为条件：

µ**Z** = GNNµ**(A**，**X)** logσ**Z** = GNNσ**(A**，**X)** (9.2)

µ**Z**是一个× d维的矩阵，该矩阵指定输入图中每个节点的均值嵌入值。 矩阵同样指定每个节点的潜在嵌入的日志方差1。

给定已编码的 µ**Z** 和 log σ**Z** 参数，我们可以通过计算来采样一组潜在的节点嵌入

(9.3)

是具有独立采样的单位法向条目的× d维矩阵。

##### 解码器模型

给定采样节点嵌入的矩阵 ，解码器模型的目标是预测图中所有边的概率。从形式上，解码器必须指定pθ(**A**|**Z**)（给定节点嵌入的邻接矩阵的后验概率）。同样，在这里，可以使用我们已经在本书中讨论的许多技术，例如在第3章中介绍的各种边缘解码器。在原始VGAE论文中，Kipf和Welling[2016b]使用了定义的简单点积解码器如下：

（9.4）

σ 用于表示sigmoid函数，但是请注意，只要这些解码器生成有效的概率值，就可以使用各种边解码器。

为了使用此方法计算方程（9.1）中的重建损耗，我们只需假设边之间的独立性，并在整个图上定义后验概率，如下所示：

这对应于边缘概率上的二进制交叉熵损失。为了在训练后生成离散图，我们可以基于等式（9.4）中的后伯努利分布对边缘进行采样。

##### 限制

### 上一节中概述的基本VGAE模型定义了有效的图形生成模型。在训练了该模型以重建一组训练图之后，我们可以从标准正态分布中采样节点嵌入Z，然后使用我们的解码器生成图。但是，这种基本方法的生成能力极为有限，尤其是在使用简单的点积解码器时。主要问题是解码器没有参数，因此如果没有训练图作为输入，则该模型将无法生成非平凡的图结构。确实，在针对该主题的最初工作中，Kipf和Welling [2016b]提出了VGAE模型作为生成节点嵌入的方法，但他们并不打算将其作为生成模型来对新图形进行采样。

### 一些论文提出通过使解码器更强大来解决作为生成模型的VGAE的局限性。例如，Grover等 [2019]提出用基于GNN的“迭代”解码器来增强解码器。但是，简单的节点级VAE方法尚未成为一种成功且有用的图形生成方法。它在重建任务和作为自动编码器框架方面均取得了出色的成绩，但是作为一种生成模型，这种简单的方法受到了严重限制。

### 9.1.2图级延迟

作为上一节中描述的节点级VGAE方法的替代方法，还可以基于图级潜在表示来定义变体自动编码器。在这种方法中，我们再次使用ELBO损耗（公式9.1）来训练VAE模型。但是，我们修改了编码器和解码器功能，以使用图级潜在表示。本节中描述的图级VAE最初是由Simonovsky和Komodakis [2018]提出的，名称为Graph VAE。

##### 编码器模型

图级VAE方法中的编码器模型可以是随池化层扩展的任意GNN模型。尤其,我们将让GNN：||V |×d表示任意k层GNN,它输出一个节点嵌入矩阵,我们将使用POOL：||V |×d表示将节点嵌入矩阵|V |×d映射到图级嵌入向量的池化函数，使用此表示法，我们可以通过以下方程式为图级VAE定义编码器：

在这里，我们再次使用两个单独的GNN来参数化潜在变量后验正态分布的均值和方差。请注意，此图级编码器和上一节中的节点级编码器之间的关键区别是：在这里，我们为单个图级嵌入生成平均值和方差，而在上一节中，我们为每个单个节点定义了后验分布。

##### 解码器模型

图级VAE中编码器模型的目标是定义,即给定图级潜在嵌入的特定图结构的后验分布。最初的GraphVAE模型建议通过将基本的多层感知器（MLP）与伯努利分布假设相结合来应对这一挑战[Simonovsky and Komodakis，2018]。在这种方法中，我们使用MLP将潜向量映射到边缘概率矩阵：

(9.7)

使用Sigmoid型函数σ来保证[0,1]中的条目。原则上，我们可以用类似于节点级的情况来定义后验分布：

其中表示图的真实邻接矩阵，而是我们预测的边缘概率矩阵。 换句话说，我们简单地假设每个边缘具有独立的伯努利分布，并且总体对数似然目标等效于每个边缘上的独立二进制交叉熵损失函数集。但是，在实践中实现公式（9.8）面临两个主要挑战：

1. 首先，如果我们使用MLP作为解码器，那么我们需要假定固定数量的节点。通常，通过假设最大数量的节点并使用屏蔽方法来解决此问题。特别地，我们可以假设最大节点数nmax,这将解码器MLP的输出尺寸限制为大小为nmax×nmax的矩阵。在训练期间用<nmax个节点解码图，我们仅掩盖（即忽略）中行或列索引大于的所有条目。为了在训练模型后生成大小可变的图，我们可以在支持{2，...，nmax}的图的大小上指定分布，并从该分布中采样以确定生成的图的大小。指定的简单策略是在训练数据中使用图的大小的经验分布。
2. 应用等式（9.8）的第二个主要挑战是，当我们计算重建损失时，我们不知道中行和列的正确顺序。矩阵是由MLP简单生成的，当我们想使用来计算训练图的似然性时，我们需要隐式地假设节点（即的行和列）上的某种排序。形式上，等式（9.8）中的损失要求我们指定一个节点，该节点对π∈Π进行排序，以对中的行和列进行排序。

这很重要，因为如果我们只是忽略此问题，则解码器可能会过度适合训练过程中使用的任意节点顺序。有两种流行的策略可以解决此问题。Simonovsky和Komodakis[2018]提出的第一种方法是应用图匹配启发式方法，尝试为每个训练图找到的节点排序，从而给出最大的可能性，从而将损失修改为

其中，我们使用表示特定节点顺序π下的预测邻接矩阵。但是，不幸的是，即使使用启发式逼近来计算等式（9.9）中的最大值，在计算上也很昂贵，并且基于图匹配的模型无法缩放到具有数百个节点的图。最近，作者倾向于使用启发式节点排序。例如，我们可以基于从最高度节点开始的深度优先或宽度优先搜索对节点进行排序。在这种方法中，我们只需指定一个特定的排序函数π，并使用该排序来计算损耗：

或者我们考虑小部分启发式排序π1, ... ,πn并在这些排序上取平均值：

这些启发式排序不能解决图形匹配问题，但在实践中似乎效果很好。Liao et al.[2019a]提供了对这些启发式排序方法的详细讨论和比较，并将此策略解释为变分近似。

##### 限制

与节点级VAE方法一样，基本图级框架也有严重的局限性。最显着的是，如上所述，使用图级潜在表示会引入指定节点顺序的问题。目前，这个问题连同MLP解码器的使用一起，将基本图级VAE的应用限制在具有数百个或更少节点的小型图形上。但是，图级VAE框架可以与更有效的解码器（包括我们在9.3节中讨论的一些自回归方法）结合使用，从而可以建立更强大的模型。在第9.5节中，当我们强调生成分子图结构的特定任务时，我们将提到这种方法的一个突出示例。

## 9.2对抗方法

变体自动编码器（VAE）是用于深度生成模型的流行框架，不仅适用于图形，而且适用于图像，文本和各种数据域。 VAE具有明确的概率动机，并且有许多工作可以利用和分析VAE模型学习到的潜在空间的结构。 但是，众所周知，VAE受到严重限制，例如VAE在图像域中产生模糊输出的趋势。许多最新的生成模型都利用其他生成框架，生成对抗网络（GAN）是最受欢迎的生成网络之一。[Goodfellow et al., 2014].

通用的基于GAN的生成模型背后的基本思想如下。首先，我们定义一个可训练的发电机网络。通过将随机种子作为输入（例如，来自正态分布的样本），训练该生成器网络以生成现实的（但虚假的）数据样本。同时，我们定义了一个鉴别器网络。鉴别器的目的是区分真实数据样本和由生成器生成的样本。在这里，我们将假定鉴别器输出给定输入为假的概率。为了训练GAN，在对抗游戏中会同时优化生成器和鉴别器：

φ （9.10）

其中pdata(**x**)表示实际数据样本的经验分布（例如，训练集中的均匀样本），pseed(**z**)表示随机种子分布（例如，标准多元正态分布）。公式（9.10）表示极小极大优化问题。生成器试图最小化鉴别器的鉴别能力，而鉴别器试图最大化其检测假样本的能力。 GAN minimax目标的优化（以及最近的变化）具有挑战性，但是与此主题相关的文献很多[Brock et al., 2018, Heusel et al., 2017, Mescheder et al., 2018]。

**生成图的基本GAN方法**

在图生成的背景下，基本的GAN方法（第一次由[De Cao and Kipf, 2018提出]）与上一节中讨论的图级VAE相似。例如，对于生成器，我们可以使用简单的多层感知器（MLP）来生成给定种子向量**z**的边缘概率矩阵：

(9.11)

给定边缘概率矩阵，我们可以生成离散邻接矩阵。通过对每个边采样独立的伯努利变量，其概率由的项给出；即。对于鉴别器，我们可以采用任何基于GNN的图分类模型。然后可以使用用于GAN优化的标准工具根据公式（9.10）训练生成器模型和判别器模型。

**GAN方法的优点和局限性**

与VAE方法一样，可以多种方式扩展用于图形生成的GAN框架。 可以使用功能更强大的生成器模型（例如，利用下一节中讨论的自回归技术），甚至可以将节点特征合并到生成器和鉴别器模型中。 [De Cao and Kipf, 2018].

## 基于GAN的框架的一个重要优点是，它消除了在损耗计算中指定节点顺序的复杂性。 只要鉴别器模型是置换不变的（几乎每个GNN都是这种情况），那么GAN方法就不需要指定任何节点顺序。 如果鉴别符是置换不变的，则由生成器生成的邻接矩阵的顺序并不重要。 然而，尽管有这个重要的好处，到目前为止，基于GAN的图生成方法比其变种方法受到的关注和成功更少。 这可能是由于基于GAN的方法所需的minimax优化所涉及的困难，而研究基于GAN的图形生成的限制目前是一个未解决的问题。

**9.3自动回归方法**

前两节详细介绍了如何将变分自动编码（VAE）和生成对抗网络（GAN）的思想应用于图生成。 但是，我们讨论的基于GAN和基于VAE的基本方法都使用简单的多层感知器（MLP）生成邻接矩阵。 在本节中，我们将介绍更复杂的自回归方法，这些方法可以从潜在表示中解码图结构。 我们在本节中介绍的方法可以与我们先前介绍的GAN和VAE框架相结合，但是它们也可以作为独立的生成模型进行训练。

### 9.3.1建模边缘依赖性

前面各节中讨论的简单生成模型假定边是独立生成的。从概率的角度，我们通过将整体似然分解为一组独立的边缘似然来定义给定一个潜在表示**z**的图的似然，如下所示：

假设边缘之间的独立性很方便，因为它简化了似然模型并允许有效的计算。但是，这是一个强有力的限制假设，因为真实世界的图形在边缘之间显示出许多复杂的相关性。例如，现实世界中的图形具有较高聚类系数的趋势很难在独立于边缘的模型中捕获。为了缓解此问题（同时仍保持易处理性），自回归模型放松了边缘独立性的假设。

取而代之的是，在自动回归方法中，我们假设边缘是按顺序生成的，并且每个边缘的可能性可以以先前生成的边缘为条件。为了使这个想法更精确，我们将使用L表示邻接矩阵**A**的下三角部分。假设我们使用的是简单图形，则**A**和**L**包含完全相同的信息，但是在以下方程式中使用**L**会很方便。然后，我们将使用符号来表示与节点对应的**L**的行，并假定**L**的行由节点索引。注意，由于**L**的下三角性质，我们将使，这意味着我们只需要考虑为任何行; 其余的可以简单地用零填充。给定这种表示法，自回归方法相当于对整体图似然性的以下分解：

### 换句话说，当我们生成行时，我们以j<i为条件对所有先前生成的行作条件。

### 9.3.2图生成的递归模型

现在，我们将讨论自动回归生成思想的两个具体实例。这两种方法建立在最初Li et al.[2018]提出的想法的基础上，并且通常表明一个人可以为此任务采用的策略。在第一个模型中，我们将回顾（Graph RNN [You et al., 2018]），我们使用递归神经网络（RNN）对自回归依赖性进行建模。在第二种方法（graph recurrent attention network (GRAN) [Liao et al., 2019a]）我们通过使用GNN对到目前为止已生成的邻接矩阵进行条件生成图。

**Graph RNN**

使用自动回归生成方法的第一个模型是Graph RNN[You et al., 2018]. Graph RNN方法的基本思想是使用分层RNN对公式（9.13）中的边依存关系建模。

层次模型中的第一个RNN（称为图级RNN）用于对到目前为止已生成的图的状态进行建模。形式上，图级RNN保持隐藏状态**h**i, 在生成邻接矩阵的每一行后更新：

(9.14)

我们使用RNNgraph来表示一个通用的RNN状态以**h**i 对应隐藏状态d **L**[vi,L]对应观察状态来更新.[[1]](#footnote-0) You et al. [2018]的原始公式,一个固定的初始隐藏状态 **h**0 =**0** 被用来初始化图级RNN，但是原则上，这种初始隐藏状态也可以通过图形编码器模型获悉，或者以VAE方式从潜在空间中进行采样。

第二个RNN（称为节点级RNN或RNNnode）以自动回归方式生成的条目。RNNnode 采用图级隐藏状态**h**i 作为初始输入值，然后顺序生成的二进制值,假设每个条目都有条件的伯努利分布。整个Graph RNN方法被称为分层方法，因为节点级RNN在每个时间步都使用图级RNN的当前隐藏状态进行初始化。

图级RNNgraph 和节点级RNNnode可以使用教学强迫策略[Williams and Zipser，1989]进行优化，以最大程度地提高训练图的可能性（方程式9.13），这意味着**L**的地面真实值在训练过程中始终用于更新RNN。为了控制生成图的大小，还对RNN进行了训练，以输出序列结束标记，这些标记用于指定生成过程的结束。请注意，与第9.1节中讨论的图级VAE方法一样，在公式（9.13）中计算似然度要求我们对生成的节点进行特定排序。

### 9.3.3自动回归方法

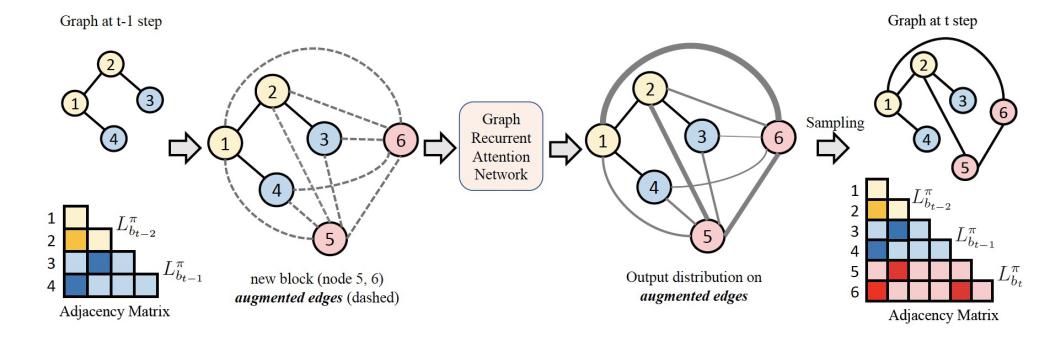


图9.2: GRAN生成方法的说明[Liao et al., 2019a].

经过训练以最大化训练图的可能性（公式9.13）之后，可以通过简单地从固定的初始隐藏状态**h**0开始运行分层RNN，使用Graph RNN模型在测试时生成图。由于边级RNN涉及随机采样过程以生成离散边缘，因此，即使使用固定的初始嵌入，Graph RNN模型也能够生成各种图样本。但是，如上所述，Graph RNN模型原则上可以分别用作VAE或GAN框架内的解码器或生成器。

##### 图循环注意力网络（GRAN）

上面讨论的Graph RNN方法的主要好处是，它可以对边之间的依赖性进行建模。 使用自回归建模假设（方程式9.13），GraphRNN可以根据在生成步骤1，... ,i–1中已经生成的图的状态，在生成步骤i中限制边的生成。通过这种方式进行调节，可以更轻松地生成复杂的图案和规则的图形结构，例如网格。例如，在图9.3中，我们可以看到，与基本的图级VAE相比，Graph RNN更具有生成网格状结构的能力（第9.1节）。但是，Graph RNN方法仍然存在严重的局限性。如图9.3所示，在网格样本上训练时Graph RNN模型仍然会生成不切实际的工件（例如长链）。此外，由于需要通过RNN递归的许多步骤进行反向传播，因此Graph RNN可能难以训练和缩放为大型图。

为了解决Graph RNN方法的一些局限性，Liao et al. [2019a]提出了GRAN模型。GRAN（代表图循环注意力网络）保持生成过程的自回归分解。但是，GRAN不是使用RNN来建模自回归生成过程，而是使用GNN。 GRAN中的关键思想是，我们可以通过在迄今生成的图上运行GNN来对邻接矩阵每一行的条件分布进行建模（图9.2）：

(9.15)

我们使用表示在生成步骤之前已生成的图的下三角邻接矩阵。公式（9.15）中的GNN可以通过多种方式实例化，但至关重要的要求是，它会生成边缘概率的向量，从中我们可以在生成过程中对离散的边缘实现进行采样。Liao et al. [2019a] 使用图注意力网络（GAT）模型的变体（请参阅第5章）来定义此GNN。最后，由于不存在与所生成的节点相关联的节点属性，因此，到GNN的输入特征矩阵可以简单地包含随机采样的向量（对区分节点很有用）。

通过使用教师强迫（teacher forcing）最大化训练图的可能性（公式9.13），可以以与GraphRNN类似的方式训练GRAN模型。像Graph RNN模型一样，我们还必须指定节点的排序以计算训练图的可能性，Liao et al.[2019a]提供了有关此挑战的详细讨论。最后，与Graph RNN模型一样，我们可以在训练后仅通过运行随机生成过程（例如，从固定的初始状态）将GRAN用作生成模型，但此模型也可以集成到VAE或基于GAN的框架中。

与Graph RNN相比，GRAN模型的主要好处是无需在图级RNN中维护长而复杂的历史记录。相反，GRAN模型在每个生成步骤中使用GNN在已生成的图形上显式条件。Liao et al.[2019a]还详细讨论了如何优化GRAN模型，以方便生成具有数十万个节点的大型图。例如，一个关键的性能改进是可以在单个块中同时添加多个节点block，而不是一次添加一个节点。图9.2阐述了此想法。

## 9.4 评估图生成

前三节介绍了一系列基于VAE，GAN和自回归模型的日益复杂的图形生成方法。在介绍这些方法时，我们暗示了某些方法相对于其他方法的优越性。我们还提供了图9.3中生成的图形的一些示例，这些示例暗示了不同方法的不同功能。但是，我们实际上如何定量比较这些不同的模型？我们如何说一种图生成方法比另一种更好？评估生成模型是一项艰巨的任务，因为没有自然的准确性或错误概念。例如，我们可以在保留的图上比较重建损失或模型似然，但是由于在不同生成方法之间缺乏统一的似然定义，这使情况变得复杂。

对于一般图生成，当前的做法是分析生成图的不同统计量，并将生成图的统计量分布与测试集进行比较[Liao et al., 2019a]。从形式上来说，假设我们有一组图统计，其中每个统计假定在上定义单变量分布。

## 9.5. 分子生成

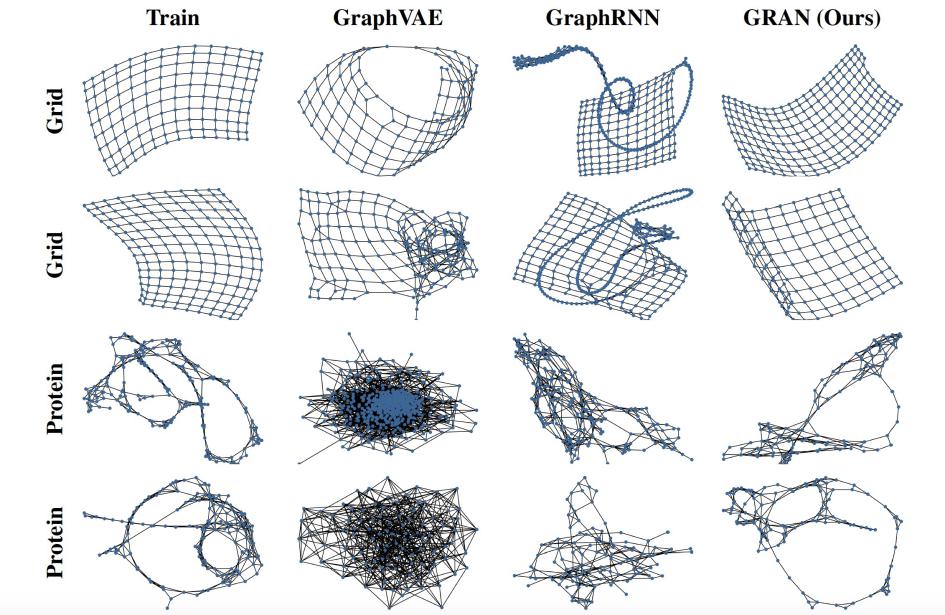


图9.3：由基本图级VAE（第9.1节）以及Graph RNN和GRAN模型生成的图形示例。每行对应一个不同的数据集。第一列显示了来自数据集的真实图的示例，而其他列是由相应模型生成的图的随机选择的样本 [Liao et al., 2019a].

对于给定的图形（例如，对于给定的图形，我们可以计算度分布、聚类系数的分布以及不同图案或图形的分布。给定特定统计数据si—在两个测试图上计算和一个生成的图我们可以使用分布度量值（如总变化距离）计算测试图上的统计分布与生成的图形之间的距离：

(9.16)

为了测量性能，我们可以计算测试集中一组生成的图和图之间的平均成对分布距离。现有工作将此策略用于图统计，例如度分布，图形小数和光谱特征，并使用总变异分数和第一个Wasserstein距离的变体计算分布距离[Liao et al., 2019b, You et al., 2018]。

**9.5分子生成**

到目前为止，我们介绍的所有图生成方法都可用于一般图生成。前面的部分没有假定特定的数据域，我们的目标只是基于给定的训练图集生成逼真的图结构（即邻接矩阵）。但是，值得注意的是，图生成的一般领域内的许多工作都专门针对分子生成的任务。

分子产生的目的是产生既有效（例如，化学稳定）又理想地具有某些所需性质（例如，药物性质或溶解性）的分子图结构。与一般的图形生成问题不同，分子生成的研究可从模型设计和评估策略的领域特定知识中受益匪浅。例如，Jin等[2018]提出了图级VAE方法的高级变体（第9.1节），该方法利用了有关已知分子基序的知识。与一般图形相比，鉴于对特定领域知识的高度依赖以及分子生成所面临的独特挑战，我们在这里将不详细介绍这些方法。但是，重要的是要突出此域作为图形生成增长最快的子区域之一。

**结论**

本书对图形表示学习进行了简短（且可能不完整）的介绍。确实，即使在撰写本文时，在这一领域也出现了许多新的重要著作，而且我期望对图形表示学习的正确概述在未来很多年内永远不会真正完成。我希望这些章节为那些有兴趣成为这些技术的从业者或寻求探索该领域新方法学前沿的人们提供了足够的基础和概述。

我的目的还在于为这些章节提供图形表示学习的快照，因为它代表了这个新兴领域的关键时刻。近年来，已经见证了将图形表示学习形式化为机器学习社区中一个真正且可识别的子领域。在这个问题上越来越受到研究关注的刺激下，图神经网络（GNN）现在已成为一种相对标准的技术。现在有数十种图的深层生成模型；而且，我们对这些技术的理论理解正在迅速巩固。但是，随着某些方法变得根深蒂固，研究的重点越来越狭窄，这种固化也带来了停滞的风险。

为此，我将在本书的结尾部分简要讨论未来工作的两个关键领域。这些当然不是该领域中唯一重要的研究领域，但是我认为这是推动图表示学习的基础发展的两个领域。

**延迟图形推理**

总的来说，本书中介绍的技术假定以图形结构作为输入。 正如我已经介绍的那样，图表示学习的挑战是如何有效地嵌入或表示这样的给定输入图。 但是，同样重要且互为补充的挑战是从非结构化或（半结构化）输入推论图或关系结构的任务。 这个任务有很多名称，但是在这里我将其称为潜在图推断。 潜在图推理是图表示学习的一项基本挑战，主要是因为即使没有输入图，它也可以使我们使用类似GNN的方法。 从技术角度来看，该研究方向可能会建立在本书第三部分中介绍的图形生成工具上。

在这一领域已经有了很有希望的初步工作，例如Kipf等人提出的神经关系推理（NRI）模型[2018]和Wang等人推断的最近邻图[2019]。关于该研究方向的最令人兴奋的事实可能是，初步发现表明，即使我们有输入图，潜在图推断也可能会提高模型性能。在我看来，建立可以推断潜在图结构超出我们提供的输入图的模型是推动图表示学习的关键方向，这也可能会打开无数新的应用程序域。

**打破消息传递的瓶颈**

就专用空间量而言，本书中最大的主题也许是神经消息传递方法，该方法在第5章中首次介绍。这种消息传递形式主义（节点将邻居的消息聚集在一起，然后以迭代方式更新它们的表示形式）是当前GNN的核心，并且已成为图形表示学习中的主要范例。

但是，神经信息传递范例也有严重的缺陷。正如我们在第7章中讨论的那样，消息传递GNNs的功能固有地受Weisfeiler-Lehman（WL）同构测试的限制。此外，我们知道这些消息传递GNNs在理论上与相对简单的卷积滤波器有关，可以由（规范化）邻接矩阵的多项式形成。根据经验，研究人员不断发现消息传递GNNs遭受过度平滑问题的困扰，而这种过度平滑问题可以看作是邻域聚合操作的结果，而邻域聚集操作是当前GNNs的核心。实际上，消息传递GNNs的核心本质是受聚合和更新消息传递范例的限制。这种范例引发了与WL同构测验以及简单图卷积的理论联系，但也基于这些理论构造而对这些GNNs的能力产生了限制。在更直观的层次上，我们可以看到GNNs的聚合和更新消息传递结构固有地引起了树状结构的计算（例如参见图5.1）。GNNs中每个节点的嵌入取决于邻域信息的迭代聚合，这些迭代信息可以表示为植根于该节点的树状计算图。注意到GNNs仅限于树形结构的计算图，这提供了它们局限性的另一种观点，例如它们无法一致地识别周期以及它们无法捕获图中节点之间的长期依赖性。

我认为，消息传递GNNs的核心限制（即，受WL测试限制，仅限于简单的卷积过滤器以及仅限于树形结构的计算图）实际上都是共同基础的不同方面原因。为了推动图表示学习的发展，有必要了解这些理论视图之间的更深层次的联系，我们将需要找到可以打破这些理论瓶颈的新架构和范式。

1. You et al. [2018] use GRU-style RNNs but in principle LSTMs or other RNN architecture could be employed. [↑](#footnote-ref-0)