图嵌入研究进展综述

刘正涛[1)]

[1)]华中科技大学计算机科学与技术学院 武汉 430074

摘 要 图作为一种重要的数据表示形式，出现在各种各样的真实场景中，如社交网络、知识图谱等。有效的图分析可以让用户更深入地了解数据背后的内容，因此可以帮助许多下游任务的推进。然而，大多数图分析方法都存在较高的计算和空间成本。图嵌入是解决图分析问题的一种有效而高效的方法，它将图数据嵌入到一个低维空间，并最大限度地保留了图的结构信息和图属性信息。目前主流的研究方向集中在两个方面，第一个是突破大规模图嵌入计算的性能制约；第二个是如何获取高质量的图嵌入表示的研究，研究人员结合图神经网络等深度学习方法，提出各种获取高质量图嵌入的学习框架。针对这两个研究方向，LightNE提出一种使用共享内存，仅需CPU参与运算的图嵌入架构；Marius利用分区缓存和缓冲区感知数据排序来最小化磁盘访问；APAN提出一种用于实时时序图嵌入的异步传播注意力网络；CompactWalks利用域和任务特定信息来增强知识图谱的嵌入；GIE注重几何空间的图嵌入表示。这些框架的提出，推动了图嵌入的进一步发展。

关键词 图嵌入；LightNE；Marius；APAN；CompactWalks；GIE

中图法分类号 TP **DOI号:** *投稿时不提供DOI号

A Survey of Graph Embedding

Zhengtao Liu[1)]

[1)]Department of Computer Science and Technology,Huazhong University of Science and Technology Wuhan 430074

Abstract

Graphs, as an important form of data representation, appear in a variety of real-world scenarios, such as social networks, knowledge graphs, etc. Effective graph analysis can give users a deeper understanding of the content behind the data, and thus can help the advancement of many downstream tasks. However, most graph analysis methods suffer from high computational and space costs. Graph embedding is an effective and efficient method to solve graph analysis problems, which embeds graph data into a low-dimensional space and preserves the structural information and graph attribute information of the graph to the greatest extent. The current mainstream research direction focuses on two aspects. The first is to break through the performance constraints of large-scale graph embedding calculations; the second is to study how to obtain high-quality graph embedding representations. Researchers combine deep learning methods such as graph neural networks to propose various learning frameworks for acquiring high-quality graph embeddings. For these two research directions, LightNE proposes a graph embedding architecture that uses shared memory and only requires CPU to participate in operations; Marius uses partition cache and buffer-aware data sorting to minimize disk access; APAN proposes a real-time temporal graph Embedded Asynchronous Propagation Attention Network; CompactWalks utilizes domain and task-specific information to enhance knowledge graph embedding; GIE focuses on graph embedding representation of geometric space. The proposal of these frameworks has promoted the further development of graph embedding.

Keywords Graph embedding; LightNE; Marius; APAN; CompactWalks; GIE

收稿日期： - - ；最终修改稿收到日期： - - .

1 引言

图结构数据自然存在于各种各样的现实世界场景中，例如社交媒体网络中的社交图、研究领域中的引用图、电子商务领域中的用户兴趣图、知识图等。分析这些图可以深入了解如何更好地利用图中隐藏的信息，因此在过去几十年中受到了极大的关注。有效的图分析可以使许多应用受益，例如节点分类、节点聚类、节点检索/推荐、链接预测等。例如，通过分析基于社交网络中的用户交互构建的图（例如，Twitter中的转发/评论/关注），我们可以对用户进行分类、检测社区、推荐朋友，并预测两个用户之间是否会发生交互。

尽管图分析是实用的和必要的，但大多数现有的图分析方法都具有较高的计算和空间成本。许多研究工作都致力于高效地进行昂贵的图分析。包括分布式图形数据处理框架（例如，GraphX、GraphLab）、新的空间高效的图存储（其加快了I/O和计算成本）等。

随着神经网络的兴起和人们对图数据表示研究的深入，图嵌入（Graph Embedding）成为了一种有效而高效的方法来解决图分析问题。具体地说，图嵌入将图转换为保存图信息的低维空间表示，通过将图表示为一个（或一组）低维向量，各种图算法就能高效地执行。由于现实中存在不同类型的图（例如，同质图、异构图、属性图等），因此图嵌入的输入在不同的场景中有所不同，而图嵌入的输出是表示图的一部分（或整个图）的低维向量。

将图嵌入低维空间并不是一项简单的任务。图嵌入的挑战取决于问题设置，问题设置包括嵌入输入和嵌入输出。不同类型的嵌入输入携带要在嵌入空间中保存的不同信息，因此对图嵌入问题提出了不同的挑战。例如，当嵌入仅具有结构信息的图时，节点之间的连接是要保留的目标。然而，对于具有节点标签或属性信息的图，辅助信息从其他角度提供了图属性，因此也可以在嵌入期间考虑。与给定和固定的嵌入输入不同，嵌入输出是任务驱动的，因此，针对不同的下游任务，图嵌入的输出也不尽相同。

目前对图嵌入的研究集中在系统和模型两个方面（当然也有将模型设计和系统性能设计结合在一起的，例如本文介绍的APAN），在系统研究方面，传统的图嵌入（图计算）认为多机系统和GPU分布式计算等硬件需求是必需的，然而本文介绍的LightNE和Marius框架致力于突破大规模图嵌入计算的性能制约，在单机系统上进行高效的图嵌入框架的设计，APAN设计的异步邮件传播机制更是在特定领域（金融领域）达到了系统毫秒级的推断速度，而CompactWalks和GIE则是在模型研究方面入手，其针对图结构中最具有代表性的知识图谱数据，设计出信息保留度更大、更合理高效的模型框架以进行图嵌入的学习表示。

2 背景

2.1 图嵌入基础知识

2.1.1 图嵌入的定义

定义1. 图的定义

图可以定义为$\mathcal{G} = (V, E)$，其中$v \in V$是节点，$e \in E$是边，$\mathcal{G}$和一个节点映射函数$f_v : V \rightarrow \mathcal{T}^v$以及一个边映射函数$f_e : E \rightarrow \mathcal{T}^e$相关联，$\mathcal{T}^v$和$\mathcal{T}^e$分别表示节点类型集合和边类型集合，每个节点$v_i \in V$属于一种特定的类型，类似的，每条边也属于一种特定的类型。

定义2. 图嵌入的定义

给定输入图$\mathcal{G} = (V, E)$和一个预先定义好的嵌入维度$d\,(d \ll |V|)$，图嵌入是将$\mathcal{G}$转化到一个d维空间，同时最大限度的保留原图的信息。可以使用诸如第一和高阶接近度的接近度度量来量化图属性，每个图表示为d维向量（对于整个图）或一组d维向量，每个向量表示图的一部分（例如，节点、边、子结构）的嵌入。

2.1.2 图嵌入的特征

- 图嵌入是用于快速比较相似数据结构信息的一种从高维映射到低维的数据结构，太大的图嵌入会占用更多的RAM和更长的时间来进行比较。

- 图嵌入压缩了图中某个顶点周围数据的许多复杂特征和结构，包括该顶点的所有属性以及主顶点周围的边和顶点的属性，围绕一个顶点的数据称为“上下文窗口”。

- 图嵌入使用机器学习算法计算，像其他机器学习系统一样，模型拥有的训练数据越多，图嵌入就越能体现一个特定任务的独特性。

- 创建一个新的嵌入向量的过程被称为“编码”或“编码一个顶点”。从嵌入中重新生成顶点的过程称为“解码”或“生成顶点“。在找到相似物体的过程中测量嵌入效果的度量被称为“损失函数”。

- 在嵌入中，可能没有与每个数字相关联的“语义”或意义，嵌入可以被认为是向量空间中一个项的低维表示，在嵌入空间中相邻的项被认为与现实世界中的项相似，嵌入关注的是性能，而不是可解释性。

- 嵌入是“模糊”匹配问题的理想选择。

- 图嵌入与其他图算法相辅相成，对于集群或分类问题，可以使用图嵌入作为附加工具来提高这些算法的性能和质量。

2.2　图嵌入框架的设计

在2000年代早期，图嵌入算法主要设计用于通过假设数据位于低维流形中来降低非关系数据的高维维数。给定一组非关系高维数据特征，基于成对特征相似性构造相似性图。然后，将图中的每个节点嵌入到低维空间中，其中连接的节点彼此更接近。自2010年以来，随着图在各个领域的扩散，图嵌入的研究开始以图为输入，并利用辅助信息来促进嵌入。一方面，其中一些集中于将图的一部分（例如，节点、边、子结构）表示为一个向量，为了获得这种嵌入，他们要么采用最先进的深度学习技术，要么设计目标函数以优化边缘重建概率。另一方面，对于图级应用，也有一些工作集中于将整个图嵌入为一个向量，图内核通常是为了满足这一需求而设计的。

近些年来，随着深度学习席卷整个学术界和工业界，由于图嵌入是作为各种深度学习下游任务的网络输入，因此对于图嵌入的模型和性能要求越来越高，研究人员在各个具体领域对于图嵌入的研究也越来越深入。由此设计出了如LightNE和Marius等致力于突破大规模图嵌入计算的性能制约，在单机系统上进行高效的图嵌入的计算框架，在知识图谱领域的CompactWalks和GIE等模型嵌入表示学习方法。

3　研究进展

3.1　LightNE

LightNE是一个经济高效、可扩展、高质量的网络嵌入系统框架，可在单个机器上扩展到具有数千亿条边的图结构。主流观点认为分布式架构和GPU是高质量大规模网络嵌入所必需的，LightNE证明，使用共享内存、仅CPU的架构，也可以实现更高的质量、更好的可扩展性、更低的成本和更快的运行时间。LightNE结合了两种理论上已经落地的嵌入方法NetSMF和ProNE，并首次将以下技术引入到网络嵌入中：（1）一种新提出的下采样方法，以降低NetSMF的采样复杂度，同时保持其理论优势；（2）高性能并行图形处理堆栈GBBS，以实现高内存效率和可扩展性；（3）稀疏并行哈希表，以在存储器中聚合和维护矩阵稀疏器；以及（4）Intel MKL，用于高效随机化SVD和频谱传播。

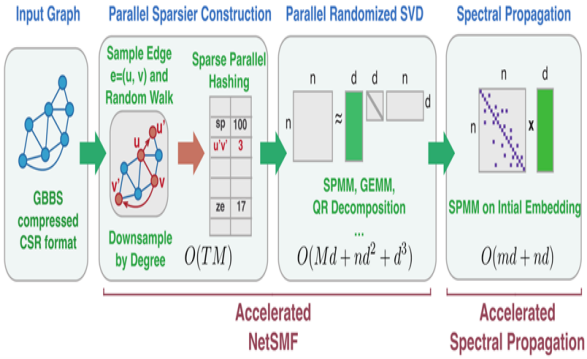

图1　LightNE系统总览

3.1.1　并行稀疏器的构建

稀疏并行图处理　LightNE涉及密集的图操作，例如执行随机游走、查询顶点度数、随机访问顶点的邻居等。在这项工作中，LightNE构建了基于图的基准套件（GBBS），该套件使用额外的纯函数基元（如顶点和图上的映射、缩减和过滤器）扩展了Ligra接口。

LightNE的一个重要设计考虑是在一台机器上嵌入非常大的图。虽然CSR格式通常被认为是一种良好的压缩图表示，但需要进一步压缩这种数据结构并减少内存使用。LightNE建立在最先进的并行图压缩技术的基础上，该技术支持快速并行图编码和解码。特别是，LightNE采用了Ligra+中的并行字节格式。在顺序字节编码中，LightNE通过对连续顶点进行差分编码来存储顶点的相邻列表，其中第一个顶点相对于源进行了差分编码。解码器一次一个地处理每个差异，并将差异相加为一个连续的和，该和给出下一个邻居的ID。Ligra+中的并行字节格式将高阶顶点的邻居划分为块，其中每个块包含可配置数量的邻居。每个块相对于源进行内部差分编码。由于每个块可以具有不同的压缩大小，因此该格式还存储从顶点起点到每个块起点的偏移。

稀疏并行哈希　因为需要计算每个不同边缘被采样的频率。LightNE考虑了在共享内存设置中解决此聚合问题的几种不同技术，包括（1）生成每个处理器的边缘列表，然后使用GBBS中引入的高效稀疏直方图合并列表，以及（2）将边缘和部分计数存储在周期性合并的每个处理器哈希表中。

LightNE的并行哈希表为每个采样的边缘存储一个不同的条目，以及一个计数。线程可以并行访问表，并且使用线性探测来解决冲突。当为单个边缘绘制多个样本时，使用原子XADD指令原子递增计数。注意到，当单个内存位置存在争用时，XADD指令比while循环中使用比较和交换的

获取和添加指令的更简单的实现要快得多，而在轻负载情况下，XADD的速度仅慢得可以忽略不计。LightNE的实现是无锁的，并确保计算每个边缘的精确计数，因为LightNE的实现使用原子指令来确保计算每个样本。

3.1.2 随机SVD与光谱传播

随机SVD 在构造稀疏器之后，下一步是有效地执行随机化SVD并获得初始嵌入。随机化SVD涉及过多的线性代数运算，“Intel MKL”库非常支持并高度优化了这些运算。LightNE利用此库执行执行随机化SVD，算法流程如下：

Algorithm 3: Randomized SVD.

Procedure RandomizedSVD(A, d)
- Sample Gaussian random matrix O and P // vsRngGaussian
- Gaussian random projection $Y = A^{\top}O$ // mkl_sparse_s_mm
- Orthonormalize Y // LAPACKE_sgeqrf, LAPACKE_sorgqr
- Compute $B = AY$ // mkl_sparse_s_mm
- Gaussian random projection $Z = BP$ // cblas_sgemm
- Orthonormalize Z // LAPACKE_sgeqrf, LAPACKE_sorgqr
- Compute $C = Z^{\top}B$ // cblas_sgemm
- Run SVD on $C = U\Sigma V^{\top}$ // LAPACKE_sgesvd
- **return** ZU, Σ, YV // cblas_sgemm

频谱传播 除了随机化SVD，谱传播步骤还涉及线性代数运算。频谱传播步骤是高效的，它不需要计算L的高次幂，而是只在稀疏矩阵之间应用重复的稀疏矩阵矩阵乘法（SPMM）n × n 拉普拉斯矩阵L和稠密n × d 嵌入矩阵，这也可以由MKL稀疏BLAS例程处理。

3.2 Marius

用于学习大规模图嵌入的系统受到数据移动的阻碍，这导致了资源利用率低和训练效率低，且这些限制要求用最先进的系统在多台机器上进行分布式训练。基于此Marius被提出，这是一种用于图嵌入的有效训练的系统，它利用分区缓存和缓冲区感知数据排序来最小化磁盘访问，并将数据移动与计算交错以最大化利用率。将Marius与两个最先进的工业系统在各种基准上进行比较，Marius达到了同样的精度，但速度快了一个数量级。同时Marius可以将训练扩展到数据集层面，其数量级超过了单机的GPU和CPU内存容量，从而可以在具有16GB GPU内存和64GB CPU内存的单机上训练超过十亿条边和550GB总参数配置的图。

3.2.1 管道式训练架构

Marius的训练分为五个阶段，四个阶段负责数据移动操作，一个阶段负责模型计算和GPU内参数更新。四个数据移动阶段具有可配置数量的工作线程，而模型计算阶段仅使用单个工作线程以确保GPU上存储的关系嵌入同步更新。

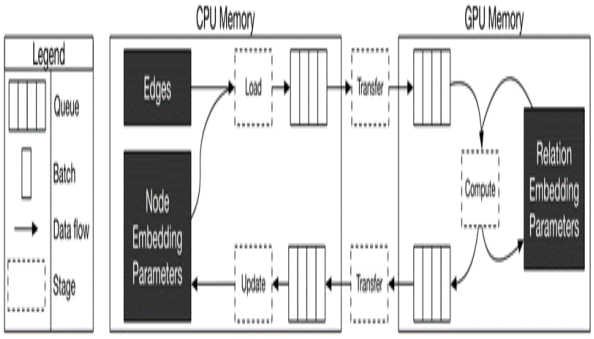

图2 Marius管道训练框架

阶段1：加载 此阶段负责加载边（即，与一对节点ID和连接它们的边类型相对应的条目）和形成用于训练的一批输入的对应节点嵌入向量。在此阶段中构造的边缘有效载荷包括图中出现的真实边缘和形成损失函数所需的负边缘（即假边缘）的均匀样本。

阶段2：传输 此阶段的输入由边缘（节点id和边缘类型三元组）和前一阶段的节点嵌入组成。此阶段的工作线程使用cudaMemCpy将数据从CPU异步传输到GPU。

阶段3：计算 计算阶段是唯一不涉及数据移动的阶段。该阶段在GPU上进行，其中在阶段1中创建的边缘和节点嵌入的有效载荷与关系嵌入向量（对应于与每个条目相关联的边缘类型）组合，以形成完整的批次。然后，工作线程计算模型更新，并将更新应用于存储在GPU中的关系嵌入。节点嵌入的更新（即，需要添加到节点嵌入参数的先前版本的缩放梯度）被放置在输出队列上，以从GPU内存传输。

阶段4：传输 节点嵌入更新被传输回CPU，使用与第2阶段类似的机制。

阶段5：更新 管道训练的最后一个阶段将节点嵌入更新应用于CPU内存中存储的参数。

3.2.2 Out-of-memory训练

由于来自磁盘的IO可能很慢（例如，分区大小可能为10 GB左右），因此最好隐藏IO等待时间，并尽量减少从磁盘到内存的交换次数。Marius通过将训练管道与构成分区内存缓冲区的分区缓冲区集成，有效地隐藏IO等待时间，同时通过开发遍历图数据的新顺序来最小化从磁盘到内存的交换次数。

给定一个分区图，有许多边缘桶排序可以用于遍历。为了最小化需要从磁盘加载分区的次数，Marius提出一种新的BETA排序方法，以最小

化所需的分区交换次数并通过以下方式进一步减少IO开销：1）根据近期需要预取节点分区，以及2）使用最佳缓冲区逐出策略，该策略将删除未来使用最远的分区。

缓冲区感知边缘遍历算法（BETA）是一种计算边缘桶排序的算法，实现了接近最优的分区交换数量，并改进了希尔伯特空间填充曲线等局部感知排序，其算法流程如下：

Algorithm 3: *BETA* Ordering Buffer Sequence

```
PartitionBufferSequence = {};
CurrentBuffer = [0 . . . c − 1];
OnDisk = [c . . . p − 1];
PartitionBufferSequence.append(CurrentBuffer);
while OnDisk.size() > 0 do
    for i in range(OnDisk.size()) do
        swap(CurrentBuffer[−1], OnDisk[i]);
        PartitionBufferSequence.append(CurrentBuffer);
    n = 0;
    for i in range(c − 1) do
        if i ≥ OnDisk.size() then
            break;
        n = n + 1;
        CurrentBuffer[i] = OnDisk[i];
        PartitionBufferSequence.append(CurrentBuffer);
    OnDisk = OnDisk[n : end];
return PartitionBufferSequence;
```

3.3 APAN

为了捕获高阶结构特征，大多数基于GNN的算法学习k跳邻居信息的节点表示。由于查询k跳邻居的时间复杂度很高，对于巨大的密集时序网络，大多数图算法无法执行毫秒级模型推理。这个问题极大地限制了在某些领域应用图算法的潜力，特别是金融欺诈检测。异步传播注意力网络（APAN）是一种用于实时时序图嵌入的异步连续时间动态图算法。传统的图模型通常执行两个连续的操作：首先是图查询，然后是模型推理。与以往的图算法不同，APAN将模型推理和图计算解耦，以减轻繁重的图查询操作对模型推理速度的影响。APAN的框架由基于注意力的编码器、基于多层感知器（MLP）的解码器和异步邮件传播器模块组成。

3.3.1 基于注意力的编码器

Positional Encoding　考虑到收到邮件的到达顺序，对每个邮件进行位置编码。通过设置的邮箱中的最大邮件数，将位置信息转换为one-hot编码格式，然后将其输入到嵌入查找层。

Multi-head Attention　通过使用点积注意力机制，APAN模型可以捕捉上次嵌入之间的关系Z 和邮箱信息M, 这意味着注意力模块可以根据从节点的时序相邻的邻居接收的邮件来确定如何更新节点

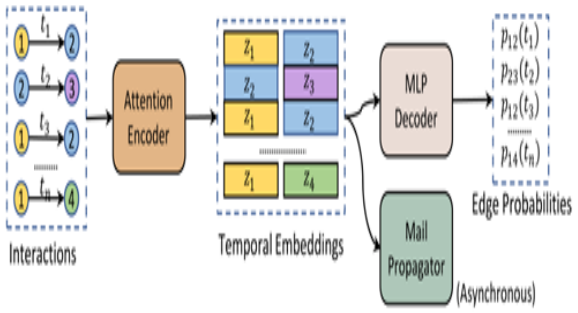

图3　APAN总体架构

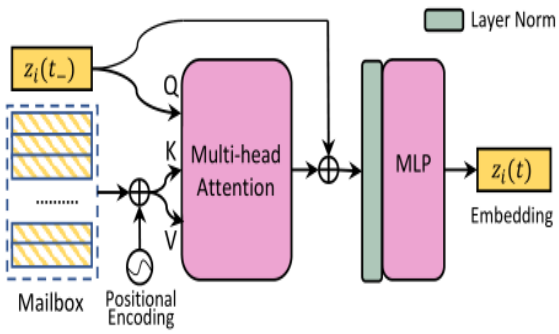

图4　编码器

嵌入。

$$Attn\left(Q,K,V\right) = soft\max\left(\frac{QK^{\top}}{\sqrt{d}}\right)V,$$

$$Q = z\left(t-\right)W_{Q},$$

$$K = \hat{\mathcal{M}}\left(t\right)W_{K}, V = \hat{\mathcal{M}}\left(t\right)W_{V}$$

Layer Normalization　由于不同节点的注意力输出是不同的，使用一个归一化层来限制输出的均值和方差。

3.3.2 异步邮件传播器

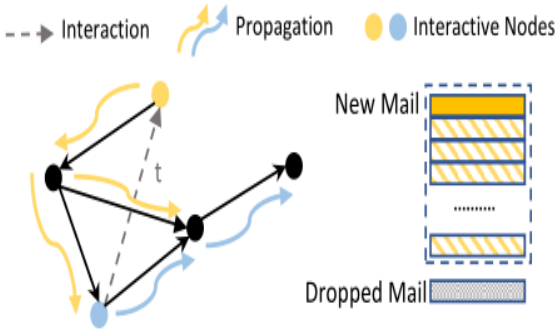

图5　异步邮件传播器

Mail Generation 一旦一个节点参与了交互，APAN的目标是生成一封邮件，记录该节点在交互中发生的情况。邮件的生成只是两个交互节点的当前嵌入和当前交互的特征的简单的总和。

Temporal Neighbors Sampling 向所有邻居投递邮件是低效的，文章采取最新的邻居采样策略（most-recent neighbor sampling）应用于APAN，因为CTDG方法旨在建模快速变化趋势并更新节点嵌入。

Mail Passing 在APAN中，邮件传递函数被简单地设置为标识函数。

Mail Reducing 一个节点通常在邮件传播期间接收多封邮件，活动（高级别）节点通常比非活动（低级别）节点接收更多邮件，为了避免这种不平衡，使用“平均”操作的缩减函数将多封邮件转换为一封邮件。

Mailbox Updating 节点收到邮件后，应更新其邮箱以汇总节点邻居的历史状态。为了尽可能简洁，采用“先进先出队列”数据结构来更新邮箱，通过此队列结构，邮箱将保留最新信息并丢弃旧邮件。

3.4 CompactWalks

知识图谱（KG）嵌入已成为解决现代生物医学研究面临的挑战的一种解决方案，包括治疗需求和可用治疗之间的日益增长的差距。KG嵌入在图分析中的流行程度正在上升。为了解决现实生活中KG嵌入工具中的混杂性挑战和记录的运行时性能挑战，CompactWalks使用域和任务特定信息来指定定义感兴趣的KG节点邻域的正则表达式路径。CompactWalks框架使用这些语义子图在基于随机行走的KG嵌入方法中实现有意义的紧凑行走。CompactWalks方法有可能解决将嵌入工具应用于现实生活中、生物医学领域以及其他领域的大规模KGs中的混杂性和运行时性能挑战。

3.4.1 CompactWalks框架

CompactWalks算法为输入KG的选定节点生成嵌入向量。然后，这些向量可以用作下游分析的输入，这些输入适合用户的目的。也就是说，输入集的N个节点包括标记为药物的节点，并且可以基于余弦相似度度量对算法输出之间的距离进行排序。

借助于输入的域和任务特定信息的正则语言L，CompactWalks对KG的输入节点集N中的每个元素n构建一个语义子图Sg，一旦N集中节点n的子图Sg被生成，算法将黑盒嵌入工具E输入应用到每个子图中。该工具为每个子图创建一个随机游走集，然后，工具E使用为所有语义子图创建的随机游走来生成集合N中所有节点的嵌入向量，算法如下：

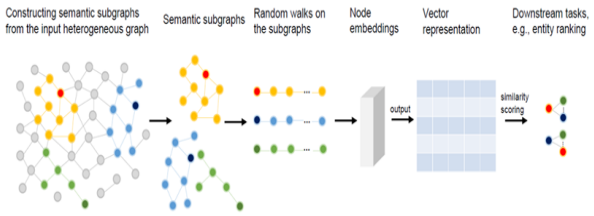

图6 CompactWalks框架

Algorithm 1: Building embeddings in semantic subgraphs

Data: Knowledge graph G; set N of nodes of G; embedding tool $\mathcal{E}$; domain- and task-specific regular language L.

Result: Set $\mathcal{V}$ of embedding vectors for the elements of N in the multidimensional space created by $\mathcal{E}$.

```
begin
    V ← ∅; // initialization
    for n ∈ N do
        Sg ← GenerateSemanticSubgraph(G, n, L);
        V ← V ∪ E(Sg, n); // building an embedding
            vector for n via compact walks of E in Sg
    return V;
```

3.4.2 CompactWalks正则语言

CompactWalks框架中语义子图设计的灵感来源于形式化的药物-疾病对的临床结果路径（COP）概念。药物-疾病COP可以被视为连接抽象KG中“药物”和“疾病”类型节点的常规路径查询，其中包含所有可能的相关生物医学信息；直观地，COP描述了药物对疾病作用的可能生化机制。例如，Swanson的ABC三角形是一个COP，其中a代表“药物”类型的节点，B代表“目标”类型（例如，蛋白质）的节点，C代表“疾病”类型节点，意思是药物通过作用于与疾病相关的靶点来治疗疾病。COP可以用作所提出的CompactWalks框架的域和任务特定正则语言中的正则表达式。

3.5 GIE

知识图谱（KG）嵌入在链接预测任务的实体和关系的学习表示中显示出强大的能力。先前的工作通常将KGs嵌入到单个几何空间中，例如欧几里得空间（零弯曲）、双曲空间（负弯曲）或超球面空间（正弯曲），以保持其特定的几何结构（例如，链、层次和环结构）。然而，KGs的拓扑结构似乎很复杂，因为它可能同时包含多种类型的几何结构。因此，在单个空间中嵌入KGs，无论是欧氏空间、双曲空间还是超球面空间，都无法准确捕捉KGs的复杂结构。为了克服这一挑战，几何交互知识图嵌入（GIE）被提出，它可以在欧氏空间、双曲空间和超球面空间之间交互学习空间结构。理

论上，GIE可以捕获一组更丰富的关系信息，对关键推理模式进行建模，并实现跨实体的表达性语义匹配。

3.5.1 几何交互

几何相互作用在捕捉空间的可靠结构方面起着重要作用。首先，几何信息传播是几何交互的一个先决条件，鉴于欧几里得空间、双曲空间和超球面空间的不同空间性质，它们的空间结构是通过不同的空间度量（例如欧几里得度量或双曲度量）建立的，这将影响这些空间之间的几何信息传播。幸运的是，指数和对数映射可以用作这些空间之间的桥梁，用于几何信息传播。

几何交互可以概括为三个步骤：（1）首先为知识图中的实体e生成空间嵌入，包括欧氏空间中的嵌入e、双曲空间中的嵌入式H和超球面空间中的内嵌S。（2）我们通过对数映射将嵌入H和S分别从双曲空间和超球面空间映射到切线空间，以便于几何信息传播。然后，我们使用注意力机制来交互欧几里德空间和正切空间的几何信息。（3）我们捕捉不同空间的特征，然后根据交互式几何信息自适应地形成可靠的空间结构。

由于两个相反方向（即从头部实体到尾部实体以及从尾部实体到头部实体）的传播消息是不同的。为了解决这一问题，GIE分别利用头部实体和尾部实体在不同树中的嵌入来传播几何交互的几何消息。

3.5.2 多空间连接

几何交互机制允许GIE灵活地利用欧几里得空间、双曲空间和超球面空间。对于KGs中的链结构，GIE可以调整几何交互中嵌入空间的局部结构，使其更接近欧氏空间。如果实体的嵌入位置同时具有多个结构特征（例如，层次结构和环结构），GIE可以集成不同的几何信息，然后通过如上所述的几何交互来相应地形成可靠的空间结构。

同时，GIE可以将欧几里得空间中建立的逻辑规则扩展到几何相互作用中的双曲空间和超球面空间。鉴于此，几何交互使GIE能够方便地在不同空间中建模关键关系模式。

4 总结

随着对图嵌入研究的深入，越来越多图嵌入框架被提出，本文综述了5个目前最新的性能卓越的图嵌入框架，对其方法和思想做了一定的阐述和总结。

LightNE是一种单机共享内存系统，它显著提高了最先进网络嵌入技术的效率、可扩展性和准确性。与最近的三个网络嵌入系统GraphVite、PyTorch BigGraph和NetSMF相比，LightNE结合了两种先进的网络嵌入算法NetSMF和ProNE，在九个基准图数据集上实现了最先进的性能。通过结合稀疏并行图处理技术和其他并行算法技术（如稀疏并行哈希和高性能并行线性代数），LightNE能够在几个小时内学习具有数千亿条边的图的高质量嵌入。

Marius是一个用于在单机上计算大规模图嵌入模型的新框架，证明了图嵌入的可扩展训练的关键是优化数据移动。为了优化数据移动并最大化GPU利用率，Marius提出了一种利用分区缓存和BETA排序的管道结构，这是一种新的缓冲软件数据排序方案。标准基准测试表明，Marius达到了相同的精度的情况下比现有系统快了一个数量级。在单个AWS P3.2xLarge实例上，Marius可以扩展到具有超过10亿条边和多达550 GB模型参数的图实例。

APAN是第一个可以实现毫秒级推理的GNN算法，可以帮助在在线分布式图形数据库中实现超大规模的推理。它可能会推动行业未来在不同领域的设计解决方案，如推荐系统、金融系统、社交网络等。注意力编码器和邮件传播器中的几乎每个模块仍有很大的改进空间，与APAN提出的这些简单模块相比，其他更复杂的模块可能更有潜力改进异步CTDG框架。

CompactWalks方法使用域和任务特定信息来指定正则表达式路径，以定义感兴趣的KG节点的邻域；所得到的语义子图在基于随机游走的嵌入方法中实现有意义的紧凑游走，CompactWalks方法有可能解决将嵌入工具应用于大规模KGs、生物医学领域以及其他领域中的混杂性和运行时性能挑战。

知识图嵌入模型GIE可以同时利用欧氏、双曲和超球面空间中具有交互作用的知识图的结构。理论上，GIE的优势在于它能够以自适应的方式学习可靠的空间结构特征。此外，GIE可以包含实体之间丰富而富有表现力的语义匹配，并满足关系表示学习的关键。

致 谢 感谢施展、童薇和胡燏翀老师的悉心教导。

参 考 文 献

[1] Qiu J, Dhulipala L, Tang J, et al. Lightne: A lightweight graph processing system for network embedding[C]//Proceedings of the 2021 international conference on management of data. 2021: 2281-2289.

[2] Wang X, Lyu D, Li M, et al. APAN: Asynchronous propagation attention network for real-time temporal graph

embedding[C]//Proceedings of the 2021 International Conference on Management of Data. 2021: 2628-2638.

[3] Cao Z, Xu Q, Yang Z, et al. Geometry Interaction Knowledge Graph Embeddings[C]//AAAI Conference on Artificial Intelligence. 2022.

[4] Mohoney J, Waleffe R, Xu H, et al. Marius: Learning massive graph embeddings on a single machine[C]//15th USENIX Symposium on Operating Systems Design and Implementation (OSDI 21). 2021: 533-549.

[5] Hou P Y, Korn D R, Melo-Filho C C, et al. Compact Walks: Taming Knowledge-Graph Embeddings with Domain- and Task-Specific Pathways[C]//Proceedings of the 2022 International Conference on Management of Data. 2022: 458-469.