

图嵌入综述

谭磊

摘 要 图模型越来越广泛地应用于数据管理、知识发现和信息服务等问题中, 图嵌入作为图分析和应用的重要技术手段, 成为了人工智能领域研究的热点之一. 本文从图嵌入研究中面临的挑战出发, 主要介绍了基于矩阵分解、基于随机游走和基于深度学习的图嵌入方法. 接着, 介绍了图嵌入的典型应用. 最后, 总结了图嵌入未来研究的趋势和方向.

关键词 图嵌入, 表示学习, 深度学习

Abstract Graph models are more and more widely used in data management, knowledge discovery and information services, and graph embedding, as an important technical means for graph analysis and application, has become one of the hot spots in the field of artificial intelligence. Starting from the challenges in graph embedding research, this paper mainly introduces graph embedding methods based on matrix decomposition, random walk-based and deep learning. Then, the typical application of graph embedding is introduced. Finally, the trend and direction of graph embedding research in the future are summarized.

key word Graph embedding, representing learning, deep learning

1 引言

图广泛存在于真实世界的多种场景中, 即节点和边的集合. 对于图的研究可以解决下面的一些问题: 比如社交网络中新的关系的预测, 在 QQ 上看到的推荐的可能认识的人; 生物分子中蛋白质功能、相互作用的预测; 通信网络中, 异常事件的预测和监控以及网络流量的预测. 如果要解决以上的问题, 我们首先需要做的是对图进行表示. 除此之外, 我们最常见的一张图片、一个句子也可以抽象地看作是一个图模型的结构, 图结构可以说是无处不在, 用图中的节点或节点属性表示真实网络系统中的实体或实体标签, 用图中的边表示真实网络中的实体关系, 通过图嵌入 (Graph Embedding, GE) 研究, 可更好地分析真实网络系统的结构和性质. 通过对它们的分析, 我们可以深入了解社会结构、语言和不同的交流模式, 因此图一直是学界研究的热点. 图嵌入任务可以大致抽象为以下四类: (a) 节点分类, (b) 链接预测, (c) 聚类, 以及 (d) 可视化. 其中, 节点分类旨在基于其他标记的节点和网络拓扑来确定节点的标签 (也称为顶点). 链路预测是指预测缺失链路或未来可能出现的链路的任务. 聚类用于发现相似节点的子集, 并将它们分组在一起; 最后, 可视化有助于深入了解网络结构.

真实的图 (网络) 往往是高维、难以处理的, 20 世纪初, 研究人员发明了图形嵌入算法, 作为降维技术的一部分. 他们首先根据实际问题构造一个 D 维空间中的图, 然后将图的节点嵌入到 d ($d \ll D$) 维向量空间中. 嵌入的思想是在向量空间中保持连接的节点彼此靠近. 拉普拉斯特征映射 (Laplacian Eigenmaps) 和局部线性嵌入 (Locally Linear Embedding, LLE) 是基于这一原理的算法的例子. 然而, 可伸缩性是这种方法的一个主要问题, 它的

时间复杂度是 $O(|V|^2)$.

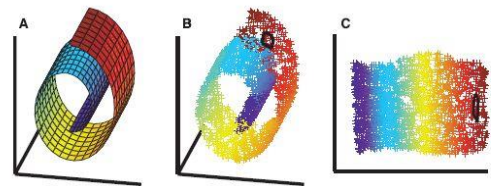


图 1 降维

大数据背景下, 图结构或图数据呈现出了海量、高维、稀疏、异构、复杂和动态等特点, 图嵌入也面临以下五个方面的挑战: 节点海量性、节点属性信息的融合、图的异构性、节点动态增量性和模型的非线性. 在图数据分析的研究中, 可采用包括图嵌入、模型构建以及问题求解这三个步骤的支撑技术, 图嵌入是最关键的步骤, 是一种图表示学习 (Graph Representation Learning) 方法, 需要将高维向量映射到低维空间, 而图表示学习未必要求嵌入向量映射到低维空间. 图嵌入方法根据其对象不同可分为两类, 一类是整个图的嵌入、只涉及小规模图, 另一类是大规模图中节点或边的嵌入. 图嵌入方法需实现以下两方面的目标: 一是在低维空间中学习得到的嵌入向量可重构原图结构, 二是学习得到的嵌入向量可支持图的推理分析.

针对现有的图嵌入方法及应用, 全面阐述各类方法的核心技术、分析对比其优缺点, 并概括其测试数据集、评测标准和应用场景, 深入研究图嵌入的新方法、探索面向实际应用的图嵌入解决方案和支撑技术, 仍具有重要的意义.

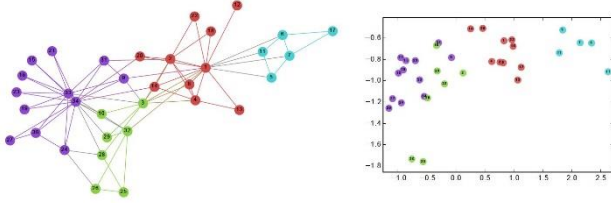
2 图嵌入问题、挑战及定义

2.1 图嵌入问题

图嵌入是一种将图数据 (通常为高维稠密的矩阵) 映射为低维稠密向量的过程, 如图 2. 图嵌入需

要捕捉到图的拓扑结构, 顶点与顶点的关系, 以及其他的信息 (如子图, 连边等)。更多的属性嵌入编码可以在以后的任务中获得更好的结果。

在嵌入的过程中存在着一种共识: 向量空间中保持连接的节点彼此靠近。基于此, 研究者提出了拉普拉斯特征映射 (Laplacian Eigenmaps) 和局部线性嵌入 (Locally Linear Embedding, LLE)。



(a) Input: Karate Graph (b) Output: Representation
图 2 图嵌入映射

总的来说图嵌入技术大致可以分为两种:

(1) 顶点嵌入: 每个顶点 (节点) 用其自身的向量表示进行编码。这种嵌入一般用于在顶点层次上执行可视化或预测。比如, 在 2D 平面上显示顶点, 或者基于顶点相似性预测新的连接。

(2) 图嵌入: 用单个向量表示整个图。这种嵌入用于在图的层次上做出预测, 可者想要比较或可视化整个图。例如, 比较化学结构。

2.2 图嵌入面临的挑战

如前所述, 图嵌入的目标是发现高维图的低维向量表示, 而获取图中每个节点的向量表示是十分困难的, 并且具有几个挑战, 这些挑战一直在推动本领域的研究:

(1) 属性选择: 节点的“良好”向量表示应保留图的结构和单个节点之间的连接。第一个挑战是选择嵌入应该保留的图形属性。考虑到图中所定义的距离度量和属性过多, 这种选择可能很困难, 性能可能取决于实际的应用场景。它们需要表示图拓扑、节点连接和节点邻居。预测或可视化的性能取决于嵌入的质量。

(2) 可扩展性: 大多数真实网络都很大, 包含大量节点和边。嵌入方法应具有可扩展性, 能够处理大型图。定义一个可扩展的模型具有挑战性, 尤其是当该模型旨在保持网络的全局属性时。好的嵌入方法需要在大型图上高效。

(3) 嵌入的维数: 实际嵌入时很难找到表示的最佳维数。例如, 较高的维数可能会提高重建精度, 但具有较高的时间和空间复杂性。较低的维度虽然时间、空间复杂度低, 但无疑会损失很多图中原有的信息。用户需要根据需求做出权衡。

2.3 图嵌入相关定义

定义 1 图表示为 $G = (V, E)$, 其中 $V = \{V_1, V_2, \dots, V_n\}$ 为 G 中节点的集合, $E = \{e_{ij} | \forall i, j \in [n], i \neq j\}$ 为 G 中边的集合。 G 的邻接矩阵 W 包含了其中所有边的非负权值, 即 $W_{ij} \geq 0$ 。若节点 v_i 和节点 v_j 之间没有边直接相连, 则 $W_{ij} = 0$ 。在无向图中, $W_{ij} = W_{ji}$ 。一般地, 用权重 W_{ij} 表示 v_i 和 v_j 之间的相似性, 权重越大、节点之间就越相似。

定义 2 节点 v_i 和 v_j 的一阶相似性定义为 v_i 和 v_j 之间边上的权重, 即 $S_{ij}^{(1)} = W_{ij}$ 。将节点 v_i 与其他节点的一阶相似性记为 $S_i^{(1)} = [S_{i1}^{(1)}, \dots, S_{i2}^{(1)}, \dots, S_{i|V|}^{(1)}]$ 。

定义 3 若节点 v_i 和 v_j 拥有相似的邻居结构, 则称 v_i 和 v_j 间具有二阶相似性, 表示为 $S_{ij}^{(2)}$, $S_{ij}^{(2)}$ 是 $S_i^{(1)}$ 与 $S_j^{(1)}$ 相似性计算的结果。高阶相似性类似于二阶相似性, 记作 $S_{ij}^{(k)}, k > 2$ 。

定义 4 $G = (V, E)$ 的嵌入是将其映射到一个低维向量空间, 其映射函数被定义为 $f: v_i \rightarrow Y_i \in R^d, \forall i \in [n], d \ll |V|$, f 能充分保存图的 $k (k \geq 1)$ 阶相似性。

表 1 符号及含义

| Notation | Description |
|------------------------------|--|
| $G = (V, E)$ | G is the graph; V is the set of the nodes; E is the set of edges |
| $v_i, e_{i,j}$ | Nodes $v_i, v_j \in V (i \neq j)$, an edge $e_{i,j} \in E$ connecting v_i and v_j |
| W | Adjacent matrix with $ V $ rows and $ V $ columns |
| D | Diagonal matrix with $ V $ rows and $ V $ columns |
| d | Dimensionality of the embedding |
| Y | Embedding of the nodes in G |
| Y_i | Embedding of node v_i |
| Y_s | Source embedding of a directed graph |
| Y_t | Target embedding of a directed graph |
| L | Graph Laplacian ($L = D - W$) |
| $\langle Y_i, Y_j \rangle$ | Inner product of Y_i and Y_j |
| S | Similarity matrix of G |
| $s_{ij}^{(1)}, s_{ij}^{(2)}$ | First-order and second-order proximity between node v_i and v_j |

3 图嵌入方法

这部分将介绍四种图嵌入方法, 其中三种属于顶点嵌入, 而另外一种属于向量嵌入。这些嵌入方式一定程度上用了 Word2vec 的嵌入原则。顶点嵌入的方法可以分为三大类: 因式分解方法、随机游走方法和深度方法, 它们在实践中经常使用, 并且通常提供最好的结果。最后一种方法嵌入整个图。

3.1 基于矩阵分解的顶点嵌入方法

基于矩阵分解的方法是将节点间的关系用矩阵的形式加以表达, 通过矩阵分解方法对图的关联矩阵进行降维, 为每个节点生成一个低维向量。通常用于表示节点关系的矩阵包括邻接矩阵, 拉普拉斯矩阵, 节点转移概率矩阵, 节点属性矩阵等。根据矩阵性质的不同适用于不同的分解策略。

3.1.1 基于拉普拉斯映射的矩阵分解

图拉普拉斯特征映射的基本思想是用节点对的相似性来保持图的性质, 在图嵌入过程中, 如果相似的节点对被映射到向量空间中的距离更远, 该

方法就给予这对节点更大的罚值。基于这一思想,该方法使用如下目标函数来实现节点对的低维映射:

$$Y = \sum_{i \neq j} (Y_i - Y_j)^2 W_{ij} = 2Y^T L Y$$

其中, L 是 G 的拉普拉斯矩阵, $L = D - W$, D 是对角矩阵, $D = \sum_j W_{ij}$, D_{ij} 值越大, Y_i 的嵌入越重要。目标函数的最小化问题可描述为:

$$Y = \min_{Y^T D Y = 1} Y^T L Y = \min_{Y^T D Y = 1} \frac{Y^T L Y}{Y^T D Y}$$

式中的约束条件是 $Y^T D Y = 1$, 可消除嵌入中的平凡解。以下几种流形学习方法都可归结为拉普拉斯特征映射方法, 如经典的 LE、CGE 和 Isomap 等。这些方法的不同之处在于, 对邻接矩阵 W_{ij} 采用的计算方法不同, 或者是在目标函数中是否引入线性函数 $Y = X^T a$ 等。

3.1.2 基于节点相似性的矩阵分解

基于节点相似性的矩阵分解用矩阵分解方法在低维空间中渐进地模拟图中相似的节点, 从而保存节点的相似性, 在给定图的原嵌入向量矩阵 Y 情况下, 最小化以下损失函数获得最终的 d 维嵌入向量:

$$Y^* = \min \| Y - Y_s Y_t^T \|$$

其中, Y_s 和 $Y_t \in R^{|V| \times d}$, Y_s 和 Y_t 分别是低维空间中源节点和目标节点的嵌入向量。通常, 使用 SVD(Singular Value Decomposition)对原嵌入向量矩阵 Y 分解, 方法如下:

$$Y = \sum_{i=1}^{|V|} \sigma_i u_i u_i^T \approx \sum_{i=1}^d \sigma_i u_i u_i^T$$

$\{\sigma_1, \sigma_2, \dots, \sigma_{|V|}\}$ 是以降序排序的奇异值, u_i 和 u_i^T 分别是奇异值 σ_i 对应的奇异向量, 嵌入的目的是获得最大的 d 个奇异值和奇异向量, 具体表示形式为:

$$Y_s = [\sqrt{\sigma_1} u_1, \dots, \sqrt{\sigma_d} u_d]$$

$$Y_t = [\sqrt{\sigma_1} u_1, \dots, \sqrt{\sigma_d} u_d]$$

节点 i 的嵌入形式为 Y_i 或 $[Y_i, Y_t]$, 采用何种形式主要取决于是否保持图的非对称结构。

3.2 基于随机游走的顶点嵌入方法

DeepWalk 使用随机游走生成嵌入。随机游走从一个选定的节点开始, 然后我们从一个当前节点移动到随机邻居节点, 移动一定数量的步。该方法主要包括三个步骤:

(1) 抽样: 用随机游动抽样一个图。从每个节点执行的随机游走很少。作者证明了从每个节点执行 32 到 64 个随机游走是足够的。一般良好的随机漫

步的长度约为 40 步。

(2) 训练跳跃图: 随机漫步与 word2vec 方法中的句子相似。跳跃图网络接受随机游走中的一个节点作为一个独热向量作为输入, 并最大化预测邻居节点的概率。它通常被训练来预测大约 20 个邻居节点——左边 10 个节点, 右边 10 个节点。

(3) 计算嵌入: 嵌入是网络隐含层的输出。

DeepWalk 计算图中每个节点的嵌入。

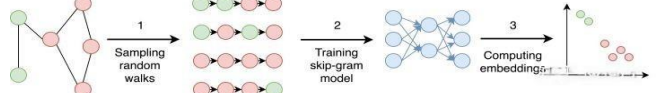


图 3 Deep walk 流程

3.3 基于深度学习的顶点嵌入方法

3.3.1 Structural deep network embedding (SDNE)

SDNE 建议使用深度自动编码器来保持一阶和二阶网络邻近度。它通过联合优化这两个近似值来实现这一点。该方法利用高度非线性函数来获得嵌入。模型由两部分组成: 无监督和监督。前者包括一个自动编码器, 目的是寻找一个可以重构其邻域的节点的嵌入。后者基于拉普拉斯特征映射, 当相似顶点在嵌入空间中彼此映射得很远时, 该特征映射会受到惩罚。

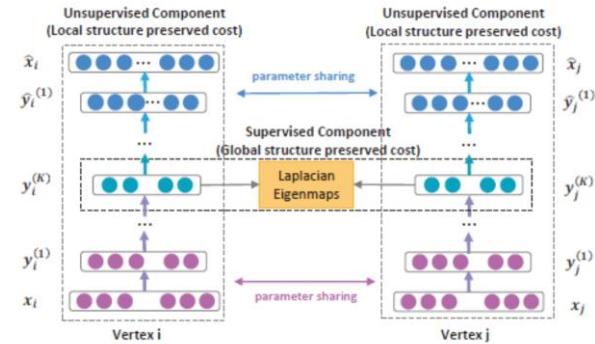


图 4 结构化网络嵌入示意

3.3.2 Deep neural networks for learning graph representations (DNNGR)

DNNGR 结合了随机游走和深度自动编码器。该模型由 3 部分组成: 随机游走、正点互信息 (PPMI) 计算和叠加去噪自编码器。在输入图上使用随机游走模型生成概率共现矩阵, 类似于 HOPE 中的相似矩阵。将该矩阵转化为 PPMI 矩阵, 输入到叠加去噪自编码器中得到嵌入。输入 PPMI 矩阵保证了自动编码器模型能够捕获更高阶的近似度。此外, 使用叠加去噪自编码器有助于模型在图中存在噪声时的鲁棒性, 以及捕获任务 (如链路预测和节点分类) 所需的底层结构。

3.3.3 Graph convolutional networks (GCN)

上面讨论的基于深度神经网络的方法, 即 SDNE 和 DNNGR, 以每个节点的全局邻域 (一行 DNNGR 的 PPMI 和 SDNE 的邻接矩阵) 作为输入。

对于大型稀疏图来说,这可能是一种计算代价很高且不适用的方法。图卷积网络(GCN)通过在图上定义卷积算子来解决这个问题。该模型迭代地聚合了节点的邻域嵌入,并使用在前一次迭代中获得的嵌入及其嵌入的函数来获得新的嵌入。仅局部邻域的聚合嵌入使其具有可扩展性,并且多次迭代允许学习嵌入一个节点来描述全局邻域。最近几篇论文提出了利用图上的卷积来获得半监督嵌入的方法,这种方法可以通过为每个节点定义唯一的标签来获得无监督嵌入。这些方法在卷积滤波器的构造上各不相同,卷积滤波器可大致分为空间滤波器和谱滤波器。空间滤波器直接作用于原始图和邻接矩阵,而谱滤波器作用于拉普拉斯图的谱。

3.3.4 Variational graph auto-encoders (VGAE)

VGAE 采用了图卷积网络(GCN)编码器和内积译码器。输入是邻接矩阵,它们依赖于 GCN 来学习节点之间的高阶依赖关系。他们的经验表明,与非概率自编码器相比,使用变分自编码器可以提高性能。

3.3.4 顶点嵌入的局限性

(1) 无法获取不在训练集中的节点的嵌入

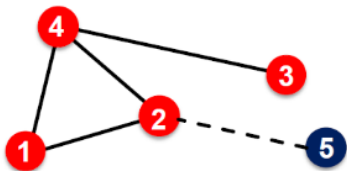


图 5

红色节点为训练集中的节点,蓝色节点为测试集中的节点。由于训练集训练的时候没有蓝色节点 5 的信息,因此测试集上的蓝色节点的嵌入无法进行计算。

(2) 无法捕捉到结构相似性

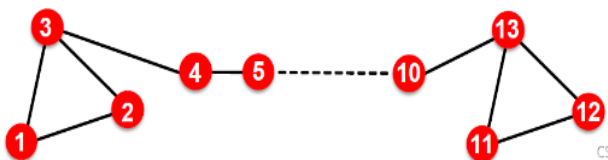


图 6

上图中的节点 1 与节点 11 都是结构相似的,但他们的嵌入结果差异非常大。而节点 1 几乎不可能随机游走到节点 11,因此不能在嵌入中刻画两者的信息。

(3) 不能利用节点、边和图的特征

例如,蛋白质相互作用图中的蛋白质属性,是有其本身的特征信息的,但嵌入只利用了图的结构。

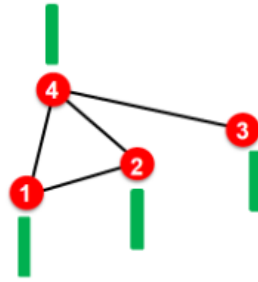


图 7

上图绿色的部分表示节点额外的特征信息,嵌入没有利用这部分的信息。

3.4 图嵌入方法

图嵌入是将整个图用一个向量表示的方法,其中最具有代表性的方法是 graph2vec。Graph2vec 是基于 skip-gram 思想,把整个图编码进向量空间,类似文档嵌入 doc2vec,doc2vec 在输入中获取文档的 ID,并通过最大化文档预测随机单词的可能性进行训练。

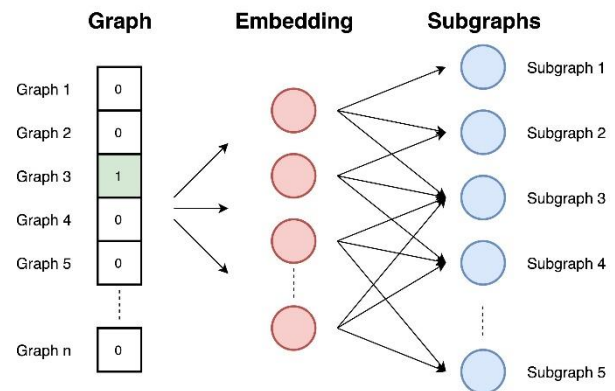


图 8 graph2vec 示意

Graph2vec 由三个步骤构成:

(1) 采样并重新标记图中的所有子图。子图是出现在所选节点周围的一组节点。子图中的节点距离所选边数不远。

(2) 训练 skip-gram 模型。类似于文档 doc2vec,由于文档是单词集,而图是子图集,在此阶段,对 skip-gram 模型进行训练。经过训练,可以最大程度地预测输入中存在于图中的子图的概率。

(3) 通过在输入处提供子图的 id 索引向量来计算嵌入。

3.5 其他方法

3.5.1 LINE

LINE 适用于任意类型的信息网络: 无向、有向和无权、有权。该方法优化了精心设计的目标函数,能够保留局部和全局网络结构。此外,LINE 中还提出了边缘采样算法,解决了经典随机梯度下降的局限性,提高了算法的有效性和效率。具体来说,LINE

明确定义了两个函数，分别用于一阶和二阶近似，并最小化了这两个函数的组合。一阶邻近函数与图分解（GF）相似，都是为了保持嵌入的邻接矩阵和点积接近。区别在于 GF 通过直接最小化两者的差异来实现这一点。相反，LINE 为每对顶点定义了两个联合概率分布，一个使用邻接矩阵，另一个使用嵌入。然后，LINE 最小化了这两个分布的 Kullback - Leibler（KL）散度。这两个分布和目标函数如下：

$$\begin{aligned} p_1(v_i, v_j) &= \frac{1}{1 + \exp(-\langle Y_i, Y_j \rangle)} \\ \hat{p}_1(v_i, v_j) &= \frac{W_{ij}}{\sum_{(i,j) \in E} W_{ij}} \\ O_1 &= KL(\hat{p}_1, p_1) \\ O_1 &= - \sum_{(i,j) \in E} W_{ij} \log p_1(v_i, v_j). \end{aligned}$$

用相似的方法定义了二阶近似的概率分布和目标函数：

$$\begin{aligned} p_2(v_j | v_i) &= \frac{\exp(\bar{u}_j^T \cdot \bar{u}_i)}{\sum_{k=1}^{|V|} \exp(\bar{u}_k^T \cdot \bar{u}_i)}, \\ O_2 &= \sum_{i \in V} \lambda_i d(\hat{p}_2(\cdot | v_i), p_2(\cdot | v_i)), \end{aligned}$$

4 图嵌入应用

作为图表示，嵌入可以用于各种任务。这些应用可以大致分为：网络压缩、聚类、链路预测、节点分类和可视化。

4.1 网络压缩

费德勒等人介绍了网络压缩的概念，又叫图形简化。对于图 G ，他们定义了一个压缩图 G^* ，它具有较少的边数。目的是更有效地存储网络，更快地运行图形分析算法。他们通过将原始图划分为二部团并用树代替它们来获得压缩图，从而减少了边的数目。多年来，许多研究人员使用基于聚合的方法来压缩图形。这一行的主要思想是利用图的链接结构对节点和边进行分组。纳瓦拉哈等人利用信息理论中的最小描述长度（MDL）将一个图汇总成一个图，并进行边缘校正。

类似于这些表示，图形嵌入也可以解释为图形的总结。通过从嵌入中重建原始图并评估重建误差，明确地验证了这一假设。它们表明，每个节点的低维表示（以 100s 的顺序）足以以高精度重建图形。

4.2 聚类

图聚类（又叫网络划分）可以分为两种类型：

（A）基于结构的聚类和（B）基于属性的聚类。前者可进一步分为两类，即基于社区的聚类和结构等效的聚类。基于结构的方法，旨在寻找具有较高的簇内边缘数和较低的簇间边缘数的密集子图。结构等价聚类相反，旨在识别角色相似的节点（如桥和离群点）。基于属性的方法除了观察到的链接外，还利用节点标签来集群节点。

在嵌入上使用 k 均值对节点进行聚类，并对在 Wordnet 和 NCAA 数据集上获得的聚类进行可视化，验证得到的聚类具有直观的解释。最近的嵌入方法还没有明确地评估它们在这项任务上的模型，因此它是图嵌入社区中一个很有前途的研究领域。

4.3 链路预测

网络是根据观察到的实体之间的相互作用构建的，这可能是不完整的或不准确的。难点往往在于识别虚假的交互和预测缺失的信息。链路预测是指预测在不断发展的网络中可能出现的缺失的交互或链路的任务。链路预测在生物网络分析中是普遍存在的，在这种情况下，验证节点之间是否存在链路需要昂贵的实验测试。将实验限制在由存在可能性排序的链接上已经被证明是非常具有成本效益的。在社交网络中，链接预测被用来预测可能的友谊，这可以用来推荐，并导致更满意的用户体验。

嵌入捕获网络的固有动态，无论是显式的还是隐式的，从而使应用程序能够链接预测。王等人 and Ou 等人预测在公开可用的协作和社交网络上的链接。此外，Grover 等人将其应用于生物学网络。它们表明，在这些数据集上，使用嵌入预测的链路比基于传统相似性的链路预测方法更准确。

4.4 节点分类

通常在网络中，一小部分节点被标记。在社交网络中，标签可能表示兴趣、信仰或人口统计。在语言网络中，文档可以标记主题或关键字，而生物网络中实体的标签可以表示功能。由于各种因素，对于很大一部分节点，标签可能是未知的。例如，在社交网络中，由于隐私问题，许多用户没有提供他们的人口统计信息。可以使用标记节点和网络中的链接推断丢失的标签。预测这些缺失标签的任务也称为节点分类。Bhagat 等人将节点分类方法分为两类，即基于特征提取和基于随机游走的方法。基于特征的模型根据节点的邻域和本地网络统计数据生成特征，然后应用 Logistic 回归和朴素贝叶斯等分类器来预测标签。基于随机游走的模型用随机游走传播标签。

嵌入可以解释为基于网络结构的自动提取节点特征,因此属于第一类。最近的工作评估了嵌入在各种信息网络上的预测能力,包括语言、社会、生物学和协作图。它们表明,嵌入可以高精度地预测丢失的标签。

4.5 可视化

可视化图的应用可以追溯到 1736 年,当时 Euler 用它来解决“Konigsberger Brucken 问题”。近年来,图形可视化在软件工程、电路、生物学和社会学中得到了应用。Battista 等人 and eades 等人调查用于绘制图表的一系列方法,并为此界定美学标准。赫尔曼等人概括这一点,并从信息可视化的角度来看待它。他们研究和比较了用于绘制图形的各种传统布局,包括基于树、三维和双曲的布局。

由于嵌入表示向量空间中的图,可以应用主成分分析(PCA)和 t 分布随机邻居嵌入(t-SNE)等降维技术对图进行可视化。Deep walk 的作者通过可视化 Zachary's Karate Club 网络来说明他们的嵌入方法的好处。LINE 的作者可视化了 DBLP 合著网络,并表明 LINE 能够将同一领域的作者聚集在一起。SDNE 的作者将其应用于 20-Newsgroup 文档相似性网络,以获得基于主题的文档集群。

5 总结与展望

本文从图嵌入的主要问题及挑战出发,介绍了当下流行的图嵌入方法,涵盖了三大类方法:基于矩阵分解的方法、基于随机游走的方法和基于深度学习的方法,并详细列出了图嵌入的各种应用场景。

图嵌入领域有三个有前途的研究方向:(1)探索非线性模型;(2)研究网络的演化;(3)生成具有现实世界特征的合成网络。如研究所示,一般非线性模型(例如基于深度学习的)在捕捉图形的固有动力学方面显示出巨大的前景。它们具有近似任意函数的能力,该函数最能解释网络边缘,这可能获得网络的高度压缩表示。这种方法的一个缺点是可解释性有限。进一步的研究集中在解释这些模型所学到的嵌入可以是非常有成效的。利用嵌入去研究图的演化是一个新的研究领域,需要进一步探索。产生具有现实世界特征的合成网络一直是一个流行的研究领域,主要是为了便于模拟。实图的低维向量表示可以帮助理解它们的结构,从而有助于生成具有现实世界特征的合成图。学习嵌入一个生成模型可以帮助我们在这方面。

参考文献

- [1]张文涛,苑斌,张智鹏,等.图嵌入算法的分布式优化与实现[J].软件学报,2021.
- [2] Jian T, Meng Q, Wang M, et al. LINE: Large-scale Information

Network Embedding: International World Wide Web Conferences Steering Committee, 10.1145/2736277.2741093[P]. 2015.

[3] Goyal P, Ferrara E. Graph Embedding Techniques, Applications, and Performance: A Survey[J]. Knowledge-Based Systems, 2017, 151(JUL.1):78-94.

[4] 王啸,崔鹏,朱文武.网络表征学习中的基本问题初探[J].中国计算机学会通讯,2018,14(3):11-15. WANG Xiao, CUI Peng, ZHU Wenwu. Preliminary research on the fundamental problems of network representation learning[J]. Communications of the CCF, 2018, 14(3): 11-15. (in Chinese)

[5] 杨东华,何涛,王宏志,等.面向知识图谱的图嵌入学习研究进展[J].软件学报,2022,33(9):21.

[6] Cai H, Zheng V W, Chang C C. A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications[J]. IEEE Transactions on Knowledge & Data Engineering, 2017.

[7] Goyal P, Ferrara E. Graph Embedding Techniques, Applications, and Performance: A Survey[J]. Knowledge-Based Systems, 2017, 151(JUL.1):78-94.

[8] Wang Q, Mao Z, Wang B, et al. Knowledge Graph Embedding: A Survey of Approaches and Applications[J]. IEEE Transactions on Knowledge & Data Engineering, 2017, 29(12):2724-2743.