中国科学: 信息科学 2021 年 第 51 卷 第 11 期: 1802–1839
SCIENTIA SINICA Informationis
论文

《中国科学》杂志社
SCIENCE CHINA PRESS
CrossMark click for updates

三元空间大数据网络关联表征

朱文武*, 王鑫

清华大学计算机科学与技术系, 北京 100084
* 通信作者. E-mail: wwzhu@tsinghua.edu.cn

收稿日期: 2020–03–11; 接受日期: 2020–05–18; 网络出版日期: 2021–11–12

国家重点基础研究发展计划 (批准号: 2015CB352300) 资助项目

摘要 三元空间是由信息空间、物理世界、人类社会所构成, 三元空间大数据由信息空间所产生的互联网数据、物理空间所产生的物联网数据和人类社会空间所产生的数据所构成. 本文介绍了三元空间大数据的关联复杂性, 并针对三元空间大数据关联复杂性这一本质困难, 提出解决三元空间异构数据的关联表征这一关键科学问题. 传统信息科学往往依据既有的先验信息进行特征表达, 并在先验表征空间内分析信息语义及其关联关系. 然而, 在表征层面所引入的先验偏见, 极大限制了信息理解和关联的广度和深度, 使得蕴含于三元空间大数据之中的超出人类现有经验的知识难以被发掘. 为解决上述难题, 本文提出将三元空间数据关联关系通过拓扑图理论表征成网络或图以实现三元空间大数据的关联表征和融合分析. 具体地, 利用数据驱动的深层网络表征对三元空间异构数据的弱先验关联关系进行深度建模以保持原始数据间的异构关联关系, 通过对非线性、非凸异构数据关联表达进行深度解离化计算以提升深层关联表征的鲁棒性与可解释性, 借助拓扑图理论挖掘三元空间大数据中蕴含的异构关联关系以达到对三元空间异构关系的精准刻画. 最后, 本文从知识与数据双驱动、自适应, 以及可推理三元空间大数据分析的角度对三元空间大数据关联表征的未来研究方向进行展望.

关键词 三元空间, 大数据, 关联表征, 深层表征, 网络表征

1 引言

互联网的诞生, 使得原本存在于人类社会的信息具备了网络传播的媒介, 引发了信息空间的膨胀并促进了人类社会与信息空间的交融, 给人类生产生活带来了一场 “互联网革命”. 近年来, 传感器技术的迅猛发展, 物理世界中的巨量信息被采集、分析和呈现, 促进了人类社会与物理世界的互动. 当前, 在物联网、社交网络等信息技术的驱动下, 信息空间、物理世界和人类社会逐渐形成高度融合的状态. 中国工程院及德国国家科学与工程院研究报告均指出, 当今世界正在经历一场巨变, 其本质即世界正在由原来的人类社会和物理世界所构成的两元空间, 进入由信息空间、物理世界和人类社会共同构成

引用格式: 朱文武, 王鑫. 三元空间大数据网络关联表征. 中国科学: 信息科学, 2021, 51: 1802–1839, doi: 10.1360/SSI-2020-0052
Zhu W W, Wang X. Cyber-physical-human big data correlational representation (in Chinese). Sci Sin Inform, 2021, 51: 1802–1839, doi: 10.1360/SSI-2020-0052

© 2021《中国科学》杂志社 www.scichina.com infocn.scichina.com

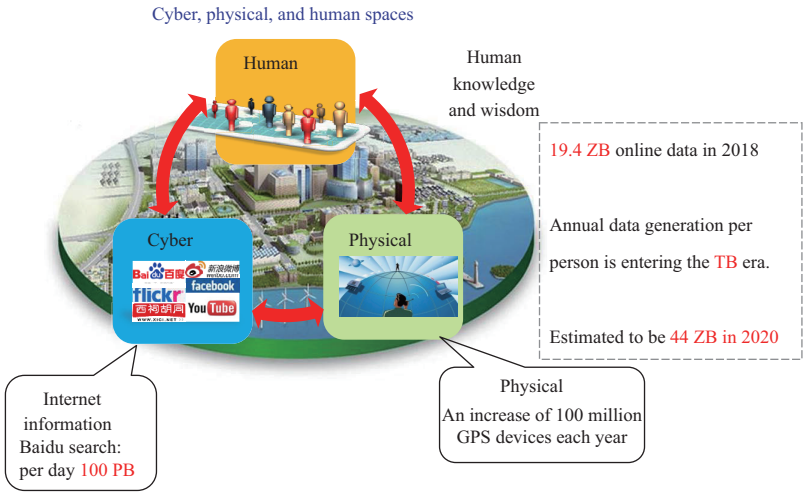

图 1 (网络版彩图) 大数据的产生:信息空间、物理世界、人类社会所组成的三元空间

Figure 1 (Color online) The generation of big data: cyber, physical, human spaces

的三元空间[1)] (如图 1 所示). 三元空间的交互融合以及对三元空间大数据的有效利用将对整个产业、城市乃至国家产生重大影响.

三元空间是由信息空间、物理世界和人类社会共同形成的动态复杂巨系统. 如何将计算对象从传统单一空间或二元空间进阶至三元空间, 并综合利用信息空间、物理世界及人类社会的计算资源、信息资源与知识资源, 通过三元空间高度融合完成计算目标, 从而揭示三元空间中异构数据关联映射机制, 已成为学术界及产业界共同面对的世界级难题. 特别地, 三元空间海量数据所呈现的跨时空、多尺度、动态关联等特性, 导致了现有信息特征表达方法的无力与匮乏. 因此迫切需要发展三元空间大数据计算的新理论与新方法. 基于以上考虑, 我们从三元空间大数据所具有的数据泛在性、数据关联性等特点出发, 将三元空间大数据相互映射与融合分析的本质困难归结为三元空间信息关联表征. 这是因为, 信息空间、物理世界、人类社会之间存在错综复杂的关联关系, 如何对其进行有效辨识, 根本在于如何对三元空间的异构信息进行有效关联表征. 而信息空间、物理世界和人类社会作为三元空间的 3 个主要组成部分[2)], 相互依存、相辅相成. 传统表征方法将该三元空间割裂开来, 在每个独立的维度上通过已有先验人工定义数据特征, 并以此为基础分析和解释各种现实现象. 这种特征表达方法存在两方面缺陷:

(1) 过于依赖先验信息定义数据表征, 缺乏对大数据全面、精准的刻画和抽象, 使得所抽取特征具有先验偏见, 影响后续知识生成和分析推理的精度;

(2) 缺乏对异构数据空间各维度关联关系的有效表达和推理, 使得面对大规模、复杂实际问题, 尤其是在面对社会管理领域以及智能城市等实际需求时, 无法精确、高效地进行数据分析和趋势预测.

另一方面, 信息空间、物理世界和人类社会形成了异构数据空间的 3 个主要组成部分, 因此发现三元空间异构信息之间的关联机制, 是实现三元空间大数据计算的重要基础; 而三元空间的异构信息关联表征, 是发现其关联机制的重要前提. 在三元空间中, 跨媒体 (跨模态) 的网络信息、多尺度的物理世界信息、模糊多维的人类社会信息, 既具有迥异的内部信息结构, 又展现出丰富的外部信息关联. 如

1) 信息空间 (cyper)、物理世界 (physical)、人类社会 (human) 所组成的三元空间, 又叫 CPH 三元空间.

2) 人类社会包含人类个体与群体, 简称 “人”, 物理世界包含地理位置信息数据, 简称 “地”, 信息空间包含互联网事件, 简称 “事件” 或 “事”.

何对其进行有效表征以支持三元空间信息的深度理解和分析, 对传统先验表征方法提出了严峻挑战, 因此迫切需要研究由数据驱动的新型信息表征理论与方法.

为解决以上难题, 本文研究三元空间异构大数据的关联表征, 利用拓扑图理论将三元空间大数据关联关系表示成网络 (图), 从而将三元空间异构大数据的弱先验关联表征转化为对三元空间异构网络 (图) 的关联表征. 具体地, 我们提出利用三元空间网络中可观测的异构关联对弱先验数据特征 (如原始数据的符号表示) 进行约束和学习, 从而形成三元空间数据的弱先验深层表征, 发现三元空间异构信息的可解释结构性深度关联, 并最终挖掘出三元空间大数据中蕴含的三元结构信息及其关联关系, 实现三元空间大数据有效的关联表征与融合分析.

综上所述, 本文主要从以下 3 方面对三元空间大数据关联表征进行研究:

(1) 研究三元空间数据的弱先验隐层空间关联表征建模, 提出 “自底向上” 数据驱动的深层特征抽取方法并结合结构化学习方法, 对数据逐层变换、层层抽象以获得最优的特征表达方式. 同时, 利用三元空间异构关系 “自顶向下” 对基于深层模型的特征表征学习进行 “指导” 和 “约束”, 使得学习得到的统一特征表示空间既保留原始数据的本质信息, 又能刻化数据间的异构关联关系.

(2) 研究三元空间大数据的特征空间与拓扑空间关联表征问题, 从异构信息关联表达的深度计算入手, 提出面向深度计算的大规模、非线性、非凸问题的可解释异构网络表征方法, 将解离表征思想引入网络表征学习, 通过对承载特征信息与拓扑信息的异构网络形成原因进行解离化表示, 大幅提升了三元空间深层关联表征的鲁棒性和可解释性.

(3) 研究三元空间的人类群体 (人) – 物理地点 (地) – 信息事件 (事) 结构信息关联分析, 以人 – 地 – 事跨空间结构关联图构建为切入点, 提出利用度量学习等统计手段对局部数据关系进行建模, 同时用主题模型对全局数据进行归类和信息挖掘, 通过拓扑图理论对统一表示空间上的人、地、事跨空间异构数据构建联通网络, 实现数据间关联关系的精准刻画.

需要指出的是, 本文中, “网络” 和 “图” 指的是同一概念, 因此, 我们在后文中会混用 “网络” 和 “图” 的表述.

2 弱先验深层网络表征

网络已经成为了最基本有效的数据组织类型之一, 越来越多的实际应用都需要借助网络数据进行表达. 例如: 微信上人与人之间的好友关系会构成社交网络, 淘宝上买家、卖家的交易关系会构成电子商务网络, 生物上蛋白质之间的作用也会构成生物网络. 而三元空间大数据之间存在复杂多样的关联关系, 这些关联关系与数据会共同构成网络. 因此, 将三元空间大数据表征成网络的形式, 更加有利于对其进行深入分析与理解. 另一方面, 网络数据往往包括异构的数据关联关系, 例如图 2 中, 样本与样本之间会构成一阶关系和二阶关系, 这些异构关系从不同侧面刻画出样本之间的相似关系, 因此如何处理异构关系在三元空间背景下具有重要的研究价值.

大多数机器学习算法主要针对向量表征的数据, 很难处理网络数据, 因此如何实现面向异构关系的网络学习算法依然是一个极具挑战且开放的问题. 我们主要介绍如何在表征学习中引入节点间的一阶关系与二阶关系这两个异构关系并利用深度神经网络对这种弱先验网络表征进行建模.

网络表征学习主要面临以下 3 个挑战:

(1) **高度非线性.** 正如 Luo 等 [1] 所陈述的, 真实网络的结构往往具有高度非线性的特点. 因此, 如何设计一个能对非线性网络结构的建模的模型是很困难的.

(2) **结构保持性.** 为了支持网络应用, 网络嵌入表征需要保持网络结构. 然而, 网络内在结构是很

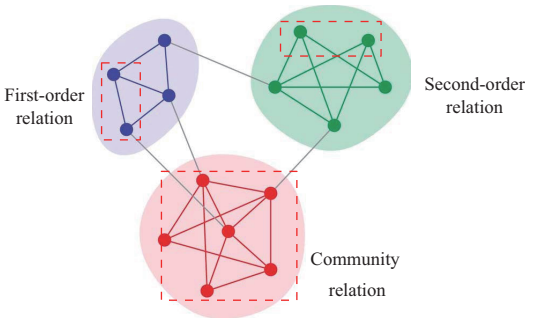

图 2 (网络版彩图) 网络中异构关系示意图

Figure 2 (Color online) Different relations in graphs

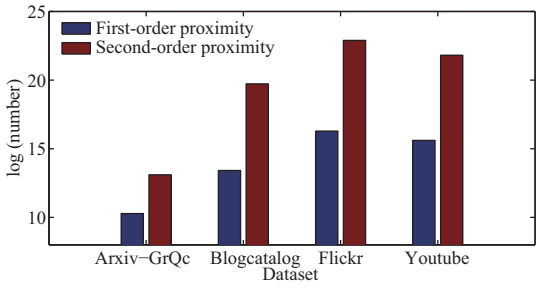

图 3 (网络版彩图) 不同数据集中拥有一阶相似度 (见定义 2) 和二阶相似度 (见定义 3) 的节点对的个数

Figure 3 (Color online) The number of pairs of vertexes which have first-order and second-order proximity in different datasets

复杂的 [2]. 节点之间的相似性和网络局部、全局的结构都是相关的. 因此, 如何利用信息保持局部和全局的结构是一个重要的问题.

(3) **稀疏性.** 许多现实场景的网络总是很稀疏的, 我们可观测的边是极其有限的, 它不能够达到一个满意的效果 [3].

近年来很多基于浅层模型的网络嵌入表征的方法被提出, 例如 IsoM [4], Laplacian eigenmaps (LE) [5] 和 LINE [6]. 然而浅层学习模型学习能力的不足的特点导致这类模型很难对高度非线性的网络结构进行有效建模 [7]. 虽然一些方法采用核方法 [8], 例如 Zhuang 等 [9] 的工作, 核方法依然是浅层模型同时具有较高的复杂度.

为了对高度非线性的网络结构关系进行建模, 我们提出结构化深度图嵌入 (structural deep network embedding, SDNE) 表征模型来生成网络节点的表征. 这是受深度神经网络最近在多个应用取得突破的启发, 深度学习被证明对学习数据中复杂的结构有很强的表达能力 [10], 同时在图片数据 [11]、文本数据 [12] 和音频数据 [13] 都取得了巨大的成功. 特别地, 本节的模型是一个包含多层非线性激活函数的多层结构, 它可以将数据从输入空间通过多个非线性函数映射到高层空间, 从而对非线性网络结构进行建模.

为了在深度模型中解决结构保持性和稀疏性的问题, 我们进一步在学习过程中引入了一阶和二阶相似度. 一阶相似度反映了网络中有边相连的节点之间的局部相似度, 它描述了局部的网络结构. 然而, 网络的稀疏性表明有边相连的点仅仅占一小部分. 因此, 一阶关系对刻画网络结构是不够的. 于是我们利用二阶相似度来弥补一阶相似度, 它表示节点间的邻居结构的相似度. 一阶相似度和二阶相似度可以较为完整地刻画出网络的全局和局部结构. 于是, 我们提出了一个半监督深度神经网络对网络的全局和局部结构进行建模, 其中非监督模块重构了节点间的二阶关系来维持网络的全局结构, 监督模块将一阶关系作为表征层的监督约束反映网络的局部结构. 通过这两个模块的联合优化, 模型能够学到局部 – 全局结构保持的表征. 另外, 正如图 3 所示, 拥有二阶相似度的节点对个数远远多于拥有一阶相似度的节点对个数. 所以二阶相似关系可以在反映网络结构上提供更丰富的信息, 在这种情况下我们的方法对稀疏网络更加鲁棒.

我们在 5 个真实网络数据集、4 个真实场景中进行了实验, 实验结果表明我们方法生成的网络表征能够更完全地重构网络, 另外我们的方法也在不同数据集的不同应用上的效果得到了显著提升, 包

括稀疏网络. 这表明在高度非线性的空间中学到的表征可以很好地保持网络结构, 同时对稀疏网络也很鲁棒.

2.1 结构化深度网络嵌入表征

在本小节中, 我们首先定义研究问题, 接着介绍我们提出的半监督深度学习模型 SDNE. 最后, 我们对模型的性质和复杂度进行进一步的讨论和分析.

2.1.1 问题定义

我们首先对图给出如下定义:

定义1 (图) 一个图被表示为 $G=(V,E)$, 其中 $V=\{v_1,\ldots,v_n\}$ 表示 n 个节点, $E=\{e_{i,j}\}_{i,j=1}^{n}$ 表示图的边. 每一条边 $e_{i,j}$ 都关联一个边权 $s_{i,j}\geqslant 0$[3]. 对于没有被边连接的节点 v_i 和 v_j 来说, $s_{i,j}=0$. 否则, 对于非权网络, $s_{i,j}=1$, 对于有权网络来说, $s_{i,j}>0$.

网络嵌入表征旨在将网络数据映射到低维隐空间, 网络中每个节点以低维空间中的向量来表示, 网络计算也可以在该空间之间进行. 正如我们之前所阐述的, 局部和全局的网络结构都有必要保持. 我们首先对描述网络局部结构的一阶相似度给出定义:

定义2 (一阶相似度) 一阶相似度描述了节点间的相互距离. 对于任意节点对来说, 如果 $s_{i,j}>0$, v_i 和 v_j 存在正向的一阶相似度. 否则, v_i 和 v_j 的一阶相似度为零.

自然的, 网络嵌入表征需要保持一阶相似度因为它表明了如果两个节点被一条边所连接, 那么它们总是相似的. 例如, 如果一篇论文引用了另一篇论文, 它们应该包含一些共同的话题. 然而, 实际场景中的网络数据集总是稀疏的: 能观察到的边只占一小部分. 有许多节点是相似的但确没有被边相连. 因此, 仅仅抓住一阶相似度是不够的. 于是, 我们引入了二阶关系来刻画网络的全局结构.

定义3 (二阶相似度) 一对节点的二阶相似度描述了它们邻居的相似度. $\mathcal{N}_u=\{s_{u,1},\ldots,s_{u,|V|}\}$ 表示节点 v_u 和其他节点的一阶相似度. 于是二阶相似度由 $\mathcal{N}_u$ 和 $\mathcal{N}_v$ 所定义.

直观上二阶相似度假设如果两个节点在网络中有许多共同邻居, 那么这两个节点倾向于相似. 这样一个假设已经在很多领域被证明是合理的[14,15]. 例如, 在语言学中, 如果两个词经常有处于类似的上下文中, 那他们很大程度是相似的[14]. 有共同好友的人总会成为好朋友[15]; 二阶相似度已经被证明是衡量一对节点相似度的有效的度量标准, 即使这对节点没有被边相连[16], 因此二阶相似度可以很好的丰富节点之间的关系. 因此, 二阶相似度的引入可以在刻画全局网络结构同时减轻稀疏度的问题.

通过一阶关系和二阶关系, 我们探索如何能够在做网络嵌入表征的时候同时保持这两种关系从而更全面的捕捉网络结构特性. 问题定义如下:

定义4 (网络嵌入表征) 给定一个图 $G=(V,E)$, 网络嵌入表征任务的目标是学习映射函数 $f: v_i\longmapsto \boldsymbol{y}_i\in\mathbb{R}^d$, 其中 $d\ll|V|$. 该映射函数的学习目标是让 $\boldsymbol{y}_i$ 和 $\boldsymbol{y}_j$ 的相似度保持 v_i 和 v_j 之间的一阶相似度和二阶相似度.

2.1.2 SDNE 模型框架

我们提出了一个半监督深度模型来进行网络嵌入表征, 其模型如图 4 所示. 首先为了对网络的非线性结构进行建模, 我们提出了一个多层深度神经网络, 它的多层非线性映射函数能将数据映射到高层隐空间来维持网络结构. 更进一步地, 对于保持网络结构和网络稀疏性的挑战, 我们提出的半监督模

3) 符号网络存在负边. 我们只关注于非负的边.

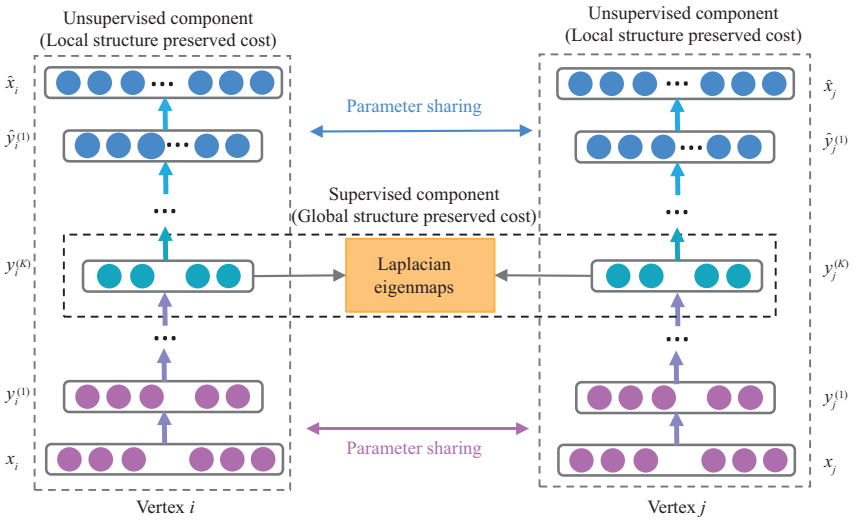

图 4 (网络版彩图) 半监督深度网络嵌入表征模型 SDNE 的框架

Figure 4 (Color online) The framework of structured deep network embedding (SDNE)

表 1 SDNE 模型相关的符号和表示

Table 1 Symbol and definition for SDNE

Symbol	Definition
n	Number of vertexes
K	Number of layers
$\boldsymbol{S}=\{\boldsymbol{s}_1,\ldots,\boldsymbol{s}_n\}$	The adjacency matrix for the network
$\boldsymbol{X}=\{\boldsymbol{x}_i\}_{i=1}^n$, $\hat{\boldsymbol{X}}=\{\hat{\boldsymbol{x}}_i\}_{i=1}^n$	The input data and reconstructed data
$\boldsymbol{Y}^{(k)}=\{\boldsymbol{y}_i^{(k)}\}_{i=1}^n$	The k-th layer hidden representations
$\boldsymbol{W}^{(k)}$, $\hat{\boldsymbol{W}}^{(k)}$	The k-th layer weight matrix
$\boldsymbol{b}^{(k)}$, $\hat{\boldsymbol{b}}^{(k)}$	The k-th layer biases
$\boldsymbol{\theta}=\{\boldsymbol{W}^{(k)},\hat{\boldsymbol{W}}^{(k)},\boldsymbol{b}^{(k)},\hat{\boldsymbol{b}}^{(k)}\}$	The overall parameters

型可以保持节点一阶和二阶的相似关系. 对于每个节点, 我们可以得到它的邻居. 因此, 我们设计了非监督的模块通过重构每个节点的邻居结构来保持二阶相似度. 同时, 对于部分有边相连的节点对, 模型中的监督模块将一阶关系作为监督约束来优化节点的隐空间表征. 通过这两部分的联合优化, SDNE 可以很好地利用高度非线性的学习能力来保持局部 – 全局的网络结构. 接下来我们将具体阐述该半监督深度学习模型的细节.

2.1.3 模型损失函数

在引入损失函数之前, 我们先在表 1 中定义了一些后面会使用的符号和表示. 注意参数上的 "$\hat{}$" 代表了解码器的参数.

现在我们介绍半监督模型的损失函数. 首先, 我们介绍非监督部分如何利用二阶相似度来维持全局的网络结构.

二阶关系刻画了节点对的邻居的相似程度. 因此, 为了建模二阶相似度, 模型需要建模每个节点的邻居结构. 给定一个网络 $G=(V,E)$, 我们能够得到这个网络的邻接矩阵 $\boldsymbol{S}$, 它包括 n 个实

例 $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$. 对于每个实例 $\boldsymbol{s}_i = \{s_{i,j}\}_{j=1}^n$, $s_{i,j} > 0$ 当且仅当 v_i 和 v_j 之间有条边. 因此, $\boldsymbol{s}_i$ 描述了节点 v_i 的邻居结构, $\boldsymbol{S}$ 提供了每个节点的邻居结构信息. 于是我们拓展传统的深度自编码器 [17] 来维持二阶相似度.

接下来我们简单回顾深度自编码器的核心思想. 它是一个由编码器和解码器两部分组成的非监督模型. 编码器的非线性函数可以将数据从原始特征空间映射到表征空间. 解码器同样包含多个非线性函数将表征空间的表征映射到重构空间. 于是给定输入 $\boldsymbol{x}_i$, 每一层的隐表征如下所示[4]:

$$\begin{aligned} \boldsymbol{y}_i^{(1)} &= \sigma(\boldsymbol{W}^{(1)}\boldsymbol{x}_i + \boldsymbol{b}^{(1)}), \\ \boldsymbol{y}_i^{(k)} &= \sigma(\boldsymbol{W}^{(k)}\boldsymbol{y}_i^{(k-1)} + \boldsymbol{b}^{(k)}),\ k = 2, \ldots, K. \end{aligned} \tag{1}$$

在得到 $\boldsymbol{y}_i^{(K)}$ 后, 我们可以通过反转编码器的计算过程来得到输出 $\hat{\boldsymbol{x}}_i$. 自编码器的优化目标是最小化输入样本和重构输出之间的误差, 其损失函数如下所示:

$$\mathcal{L} = \sum_{i=1}^{n} \|\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i\|_2^2. \tag{2}$$

正如 Salakhutdinov 等 [17] 所证明的, 虽然最小化重构误差并没有直接保持样本之间的相似度, 重构的想法可以抓住数据之间的线性关系, 因此保持了样本之间的相似度. 考虑到自编码器的性质, 我们将网络的邻接矩阵 $\boldsymbol{S}$ 作为其输入, 即 $\boldsymbol{x}_i = \boldsymbol{s}_i$, 因为样例 $\boldsymbol{s}_i$ 描述了节点 v_i 的邻居结构, 重构过程可以令有相似邻居结构的节点有相似的隐表征.

然而, 因为一些网络特定的特点, 这样一个重构过程并不能直接应用于我们的问题. 网络中存在可以观测到的少部分的边, 但是同时很多边的信息是缺失的, 这表明节点间的边可以反映他们所链接的点之间的相似性, 但是没有边相连并不代表他们之间不相似. 另外, 网络的稀疏性致使 $\boldsymbol{S}$ 中的非零项远远少于零项的数量. 因此以 $\boldsymbol{S}$ 为输入的自编码器会更倾向于重构 $\boldsymbol{S}$ 中的零元素. 然而, 这并不符合实际的需求. 于是我们在重构矩阵的非零元素时比重构零元素的时候附加更多的惩罚, 此时的目标函数如下:

$$\mathcal{L}_{2\text{nd}} = \sum_{i=1}^{n} \|(\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i) \odot \boldsymbol{b}_i\|_2^2 = \|(\hat{\boldsymbol{X}} - \boldsymbol{X}) \odot \boldsymbol{B}\|_{\text{F}}^2, \tag{3}$$

其中 $\odot$ 代表 Hadamard 积, 如果 $s_{i,j} = 0$, $b_{i,j} = 1$, 否则 $b_{i,j} = \beta > 1$. 通过将 $\boldsymbol{S}$ 作为改进后的深度自编码器的输入, 有相似的邻居结构的节点会被映射在表征空间的相邻位置从而保持二阶关系. 总的来说, SDNE 的非监督模块通过重构节点间的邻居信息来维持全局网络结构.

仅仅保持全局的网络结构是不够的, 局部网络结构也需要被维持. 我们用一阶相似度来表示局部网络结构. 一阶相似度可以看做是限制节点对的隐表征的监督信息. 因此, 我们设计了监督模块来利用一阶关系, 其损失函数为[5]

$$\mathcal{L}_{1\text{st}} = \sum_{i,j=1}^{n} s_{i,j} \|\boldsymbol{y}_i^{(K)} - \boldsymbol{y}_j^{(K)}\|_2^2 = \sum_{i,j=1}^{n} s_{i,j} \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2. \tag{4}$$

式 (4) 的目标函数借用了 Laplacian eigenmaps [5] 的思路, 它会对相似的节点在映射后的空间比较远的情况给予惩罚. 一些关于社交网络的工作 [18~20] 也采用了类似的想法. 然而, 从想法上, 我们引

4) 本工作中, 我们用 sigmoid 函数 $\sigma(x) = \frac{1}{1+\exp(-x)}$ 作为非线性激活函数.

5) 为了符号的便利, 我们将表征表示为 $\boldsymbol{Y}^{(K)} = \{\boldsymbol{y}_i^{(K)}\}_{i=1}^n$ as $\boldsymbol{Y} = \{\boldsymbol{y}_i\}_{i=1}^n$.

入了深度模型来将被边相连的节点映射到表征空间中相邻的位置. 因此, 该监督模型能够维持一阶相似度.

为了同时保持一阶和二阶相似度, 我们提出了联合式 (3) 和 (4) 的半监督模型, 并且通过最小化如下的目标函数来实现优化:

$$\begin{aligned}\mathcal{L}_{\text{mix}} &= \mathcal{L}_{\text{2nd}} + \alpha\mathcal{L}_{\text{1st}} + \nu\mathcal{L}_{\text{reg}} \\ &= \|(\hat{\boldsymbol{X}} - \boldsymbol{X}) \odot \boldsymbol{B}\|_{\text{F}}^2 + \alpha \sum_{i,j=1}^{n} s_{i,j}\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2 + \nu\mathcal{L}_{\text{reg}},\end{aligned} \tag{5}$$

其中 $\mathcal{L}_{\text{reg}}$ 是 $\mathcal{L}2$ – 范数正则项来防止过拟合, 定义如下:

$$\mathcal{L}_{\text{reg}} = \frac{1}{2}\sum_{k=1}^{K}(\|\boldsymbol{W}^{(k)}\|_{\text{F}}^2 + \|\hat{\boldsymbol{W}}^{(k)}\|_{\text{F}}^2).$$

2.1.4 模型优化方法

为了优化前述模型, 目标是优化以 $\boldsymbol{\theta}$ 为函数的 $\mathcal{L}_{\text{mix}}$. 具体来说, 关键步骤是计算 $\partial\mathcal{L}_{\text{mix}}/\partial\hat{\boldsymbol{W}}^{(k)}$ 和 $\partial\mathcal{L}_{\text{mix}}/\partial\boldsymbol{W}^{(k)}$ 的偏导, 计算方法如下:

$$\begin{aligned}\frac{\partial\mathcal{L}_{\text{mix}}}{\partial\hat{\boldsymbol{W}}^{(k)}} &= \frac{\partial\mathcal{L}_{\text{2nd}}}{\partial\hat{\boldsymbol{W}}^{(k)}} + \nu\frac{\partial\mathcal{L}_{\text{reg}}}{\partial\hat{\boldsymbol{W}}^{(k)}}, \\ \frac{\partial\mathcal{L}_{\text{mix}}}{\partial\boldsymbol{W}^{(k)}} &= \frac{\partial\mathcal{L}_{\text{2nd}}}{\partial\boldsymbol{W}^{(k)}} + \alpha\frac{\partial\mathcal{L}_{\text{1st}}}{\partial\boldsymbol{W}^{(k)}} + \nu\frac{\partial\mathcal{L}_{\text{reg}}}{\partial\boldsymbol{W}^{(k)}},\ k = 1, \ldots, K.\end{aligned} \tag{6}$$

首先看 $\partial\mathcal{L}_{\text{2nd}}/\partial\hat{\boldsymbol{W}}^{(K)}$. 它可以被重构成如下形式:

$$\frac{\partial\mathcal{L}_{\text{2nd}}}{\partial\hat{\boldsymbol{W}}^{(K)}} = \frac{\partial\mathcal{L}_{\text{2nd}}}{\partial\hat{\boldsymbol{X}}} \cdot \frac{\partial\hat{\boldsymbol{X}}}{\partial\hat{\boldsymbol{W}}^{(K)}}. \tag{7}$$

对于第 1 个式子, 根据式 (3) 可以得到

$$\frac{\partial\mathcal{L}_{\text{2nd}}}{\partial\hat{\boldsymbol{X}}} = 2(\hat{\boldsymbol{X}} - \boldsymbol{X}) \odot \boldsymbol{B}. \tag{8}$$

第 2 个式子 $\partial\hat{\boldsymbol{X}}/\partial\hat{\boldsymbol{W}}$ 的计算是简单的, 因为 $\hat{\boldsymbol{X}} = \sigma(\hat{\boldsymbol{Y}}^{(K-1)}\hat{\boldsymbol{W}}^{(K)} + \hat{\boldsymbol{b}}^{(K)})$. 于是 $\partial\mathcal{L}_{\text{2nd}}/\partial\hat{\boldsymbol{W}}^{(K)}$ 容易得到. 基于后向传播的算法, 我们可以通过迭代得到 $\partial\mathcal{L}_{\text{2nd}}/\partial\hat{\boldsymbol{W}}^{(k)}$, $k = 1, \ldots, K-1$ 和 $\partial\mathcal{L}_{\text{2nd}}/\partial\boldsymbol{W}^{(k)}$, $k = 1, \ldots, K$. 现在, 偏导 $\mathcal{L}_{\text{2nd}}$ 的计算就结束了.

接着, 我们计算偏导 $\partial\mathcal{L}_{\text{1st}}/\partial\boldsymbol{W}^{(k)}$. $\mathcal{L}_{\text{1st}}$ 的损失函数可以重构如下:

$$\mathcal{L}_{\text{1st}} = \sum_{i,j=1}^{n} s_{i,j}\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2 = 2\text{tr}(\boldsymbol{Y}^{\text{T}}\boldsymbol{L}\boldsymbol{Y}), \tag{9}$$

其中 $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{S}$, $\boldsymbol{D} \in \mathbb{R}^{n\times n}$ 是只有对角线有值的对角矩阵, $D_{i,i} = \sum_j s_{i,j}$.

进一步, 我们关注于 $\partial\mathcal{L}_{\text{1st}}/\partial\boldsymbol{W}^{(K)}$ 的计算:

$$\frac{\partial\mathcal{L}_{\text{1st}}}{\partial\boldsymbol{W}^{(K)}} = \frac{\partial\mathcal{L}_{\text{1st}}}{\partial\boldsymbol{Y}} \cdot \frac{\partial\boldsymbol{Y}}{\partial\boldsymbol{W}^{(K)}}. \tag{10}$$

由于 $\boldsymbol{Y} = \sigma(\boldsymbol{Y}^{(K-1)}\boldsymbol{W}^{(K)} + \boldsymbol{b}^{(K)})$, 第 2 个式子 $\partial\boldsymbol{Y}/\partial\boldsymbol{W}^{(K)}$ 的计算是显而易见的.

对于第 1 个式子 $\partial\mathcal{L}_{1\text{st}}/\partial\boldsymbol{Y}$, 我们可以得到

$$\frac{\partial\mathcal{L}_{1\text{st}}}{\partial\boldsymbol{Y}}=2(\boldsymbol{L}+\boldsymbol{L}^{\mathrm{T}})\cdot\boldsymbol{Y}. \tag{11}$$

类似的, 通过后向传播, 我们可以得到关于 $\mathcal{L}_{1\text{st}}$ 的偏导.

现在, 我们已经得到了所有参数的偏导的计算. 通过初始化参数后, 我们可以利用随机梯度下降来对深度学习模型进行参数. 为了避免模型的高度非线性导致陷入局部解的情况, 我们首先使用深度信念网络来对模型进行预训练 [21], 这已经被证明是一个有效地初始化深度网络的方法 [22]. 模型的整体算法如算法 1 所示.

Algorithm 1 Training algorithm for the semi-supervised deep model of SDNE

Require: the network $G=(V,E)$ with adjacency matrix $\boldsymbol{S}$, the parameters α and ν.
Ensure: network representations $\boldsymbol{Y}$ and updated parameters $\boldsymbol{\theta}$.
1: Pretrain the model through deep belief network to obtain the initialized parameters $\boldsymbol{\theta}=\{\theta^{(1)},\ldots,\theta^{(K)}\}$;
2: $\boldsymbol{X}=\boldsymbol{S}$;
3: **repeat**
4: Based on $\boldsymbol{X}$ and $\boldsymbol{\theta}$, apply Eq. (1) to obtain $\hat{\boldsymbol{X}}$ and $\boldsymbol{Y}=\boldsymbol{Y}^{(K)}$;
5: $\mathcal{L}_{\text{mix}}(\boldsymbol{X};\boldsymbol{\theta})=\|(\hat{\boldsymbol{X}}-\boldsymbol{X})\odot\boldsymbol{B}\|_{\mathrm{F}}^2+2\alpha\mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{Y})+\nu\mathcal{L}_{\text{reg}}$;
6: Based on Eq. (6), use $\partial\mathcal{L}_{\text{mix}}/\partial\boldsymbol{\theta}$ to back-propagate through the entire network to get updated parameters $\boldsymbol{\theta}$;
7: **until** converge
8: Obtain the network representations $\boldsymbol{Y}=\boldsymbol{Y}^{(K)}$.

2.2 模型分析和讨论

关于半监督模型 SDNE 的一些分析和讨论:

新的节点. 对于网络嵌入表征的一个实际问题是如何学习新出现节点的表征. 对于一个新的节点 v_k, 假如它与现有网络中的节点之间的边已知, 我们可以得到它的邻接向量 $\boldsymbol{x}=\{s_{1,k},\ldots,s_{n,k}\}$, 其中 $s_{i,k}$ 表示已经存在的节点 v_i 和新的节点 v_k 之间的相似度. 于是, 我们可以将 $\boldsymbol{x}$ 作为 SDNE 的输入, 然后用训练好的参数 $\boldsymbol{\theta}$ 得到该节点的表征 v_k. 该方法的复杂度是 $O(1)$. 如果 v_i 和已有的节点之间不存在连接, 我们的方法和基准方法都不能处理这种情况. 为了处理这个情况, 我们可以求助于边的信息, 例如节点的内容特征等, 这些留作未来工作来进行讨论.

训练复杂度. 我们的模型的训练复杂度是 $O(ncdI)$, 其中 n 表示节点的数量, d 是隐层的最大维数, c 是网络节点的平均度数, I 是迭代次数. 参数 d 一般是和嵌入表征向量相关的但是和节点个数不相关. I 和 n 也是独立的. 对于 c, 它总被认为在实际应用中是常数. 例如, 在社交网络中, 每个人的最大朋友个数是受限制的 [7]. 在 top-k 的相似性图中, $c=k$. 因此, cdI 与 n 是独立的, 因此完整的训练复杂度和网络的节点个数是线性的.

2.3 实验内容

我们在一些真实数据集和真实应用上评测不同的网络嵌入表征方法. 实验结果证明了我们提出的模型相较于基准方法有很显著的效果提升.

2.3.1 数据集

为了全面评估不同方法生成的网络表征的有效性, 我们使用了 5 个网络数据集, 包括 3 个社交网络、一个引用网络和一个语言网络, 同时在 3 个真实场景的应用上进行了实验, 包括多标签分类、链

表 2 数据集信息统计

Table 2 Dataset statistics

Dataset	#(V)	#(E)
Blogcatalog	10312	667966
Flickr	80513	11799764
Youtube	1138499	5980886
Arxiv GR-QC	5242	28980
20-Newsgroup	1720	Full-connected

接预测和可视化. 考虑到这些数据集的特点, 对于每个应用我们采用一个或者多个数据集来评估效果. 关于数据集的具体描述如下:

• Blogcatalog [23], Flickr [23] 和 Youtube [24]. 它们是在线社交网络, 每一个用户都属于至少一个分类. 对于 Blogcatalog 一共有 39 个不同的分类, Flickr 有 195 个不同的分类, 对于 Youtube 有 47 个不同的分类. 这些分类可以作为每个节点的事实. 因此, 它们可以在多分类任务上进行评测.

• Arxiv GR-QC [25]. 这是一个论文合作网络, 它包括 Arxiv 上物理方向的论文. 在这个网络中, 每个节点表示一个作者, 边表示作者间共同发表了论文. 这个数据集由于没有每一个节点的类别信息所以只能用来做链接预测的任务.

• 20-Newsgroup[6]. 这个数据集集成了大约 20000 篇新闻, 每篇新闻被标注了 20 个不同的分类中的一个. 我们用每个词的 tf-idf 来表征文档, 文档间用余弦相似度来衡量它们的距离. 我们可以根据该相似度来构建网络. 我们选择了被标注为 comp.graphics, rec.sport.baseball 和 talk.politics.gums 的文档作为可视化的数据集.

总得来说, 我们在有权网络和无权网络、稀疏和稠密网络、小型和大型网络上都进行了实验. 因此, 这些数据集可以比较综合地反映网络嵌入表征模型的特性. 表 2 展示了各个数据集的具体统计信息.

2.3.2 基准方法

我们利用以下 5 种模型作为基准方法. 前 4 个为网络嵌入表征模型, Common neighbor 只用来做链接预测.

• **DeepWalk** [3]**.** 它利用随机游走和 skip-gram 模型来生成网络表征.

• **LINE** [6]**.** 该方法分别定义了保持一阶关系和二阶关系的损失函数. 在对损失函数优化后, 该方法拼接了不同类型的表征.

• **GraRep** [26]**.** 该方法扩展原来的方法到高阶相似度, 同时利用 SVD 来训练模型. 它直接拼接了一阶和高阶的表征.

• **Laplacian eigenmaps (LE)** [5]**.** 它通过分解邻接矩阵的拉普拉斯矩阵来生成网络表征. 该方法只利用一阶关系来学习网络表征.

• **Common neighbor** [16]**.** 该方法只利用节点的共同邻居来衡量节点的相似度. 它只被用于链接预测的基准方法.

6) http://qwone.com/~jason/20Newsgroups/.

2.3.3 评测指标

本小节在网络重构、链接预测、多标签分类和可视化 4 个任务上进行了实验. 对于网络重构和链接预测任务, 我们使用 precision@k 和 mean average precision (MAP) 来评估实验效果, 定义如下:

• precision@k 是一个给予每个样本相同权重的指标, 它的定义如下:

$$\text{precision@}k(i) = \frac{|\{j \mid i, j \in V, \text{index}(j) \leqslant k, \Delta_i(j) = 1\}|}{k},$$

其中 V 是节点集, index(j) 是第 j 个节点的排序索引, $\Delta_i(j) = 1$ 表明 v_i 和 v_j 存在一条边.

• MAP 是一个具有很好的判别性和稳定性的评测指标. 和 precision@k 相比, 它更关注于排序在前面的实体的效果. 它的计算方法如下:

$$\begin{aligned}\text{AP}(i) &= \frac{\sum_j \text{precision@}j(i) \cdot \Delta_i(j)}{|\{\Delta_i(j) = 1\}|},\\ \text{MAP} &= \frac{\sum_{i \in Q} \text{AP}(i)}{|Q|},\end{aligned}$$

其中 Q 索引集.

对多标签分类的任务, 和别的工作 [23] 类似, 我们采用 micro-F1 和 macro-F1 作为评测指标. 具体来说, 对于一个标签 A, 我们用 TP(A), FP(A) 和 FN(A) 分别表示真正、假正、假负的被预测为 A 的样例个数. 假设 $\mathcal{C}$ 是所有的标签数据集. Micro-F1 和 Macro-F1 如下所定义:

• Macro-F1 给予每一个样本相同权重, 定义如下:

$$\text{Macro-F1} = \frac{\sum_{\text{A} \in \mathcal{C}} \text{F1(A)}}{|\mathcal{C}|},$$

其中 F1(A) 是被标注为 A 的 F1 值.

• Micro-F1 也是一个给予每一个样本相同权重的评测指标, 定义如下:

$$\begin{aligned}&\text{Pr} = \frac{\sum_{\text{A} \in \mathcal{C}} \text{TP(A)}}{\sum_{\text{A} \in \mathcal{C}} (\text{TP(A)} + \text{FP(A)})}, \quad R = \frac{\sum_{\text{A} \in \mathcal{C}} \text{TP(A)}}{\sum_{\text{A} \in \mathcal{C}} (\text{TP(A)} + \text{FN(A)})},\\ &\text{Micro-F1} = \frac{2 \times \text{Pr} \times R}{\text{Pr} + R}.\end{aligned}$$

2.3.4 参数设定

本工作中, 我们提出了一个多层深度结构, 每个数据集有不同的层数. 每一层的维度如表 3 所示. 对于 Blogcatalog, Arxiv GR-QC 和 20-Newsgroup, 使用 3 层网络; 对于 Flickr 和 Youtube, 使用了 4 层网络. 如果我们使用更深的网络, 效果保持不变或者更差.

对于我们提出的方法, 超参数 α, β 和 ν 通过在验证集上线性检索来进行设置. 基准方法的参数被调至最优. 对于 LINE, 起始的学习率为 0.025. 负样本的个数设置为 5, 所有采样个数设置为 100 亿. 另外, 根据 LINE [6], LINE 拼接一阶和二阶的表征会产生更好的结果. 我们遵循这样的方式得到 LINE 的结果. DeepWalk 的窗口值设置为 10, 游走长度为 40, 每个节点的游走次数为 40. 对于 GraRep, 设置最大转移步数为 5.

表 3 神经网络结构

Table 3 Neural network structures

Dataset	#nodes in each layer
Blogcatalog	10300-1000-100
Flickr	80513-5000-1000-100
Youtube	22693-5000-1000-100
Arxiv GR-QC	5242-500-100
20-Newsgroup	1720-200-100

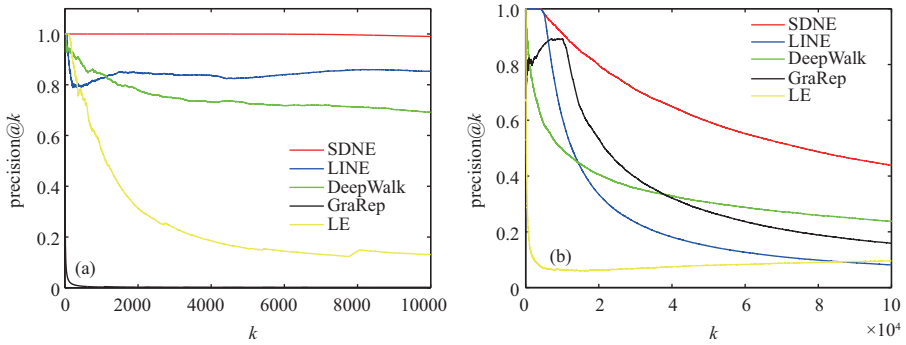

图 5 (网络版彩图) (a) Arxiv GR-QC 和 (b) Blogcatalog 上的 precision@k

Figure 5 (Color online) Precision@k for (a) Arxiv GR-QC and (b) Blogcatalog

表 4 Arxiv-GRQC 和 Blogcatalog 在重构任务上 MAP 的效果

Table 4 MAP on Arxiv-GRQC and Blogcatalog for reconstruction task

Method	MAP				
	SDNE	GraRep	LINE	DeepWalk	LE
Arxiv-GRQC	**0.836**	0.05	0.69	0.58	0.23
Blogcatalog	**0.63**	0.42	0.58	0.28	0.12

2.3.5 网络重构结果

在介绍 SDNE 模型在实际应用中的效果之前, 先对不同网络嵌入表征模型重构网络的能力给一个基本的评估. 做这个实验的原因是一个好的网络嵌入表征模型应该保证学到了表征能够保持原来网络的结构. 我们用语言网络 Arxiv GR-QC 和社交网络 Blogcatalog 作为该实验的数据集. 给定一个网络, 用不同的网络嵌入表征模型来学习网络表征, 然后去预测原来的网络的边. 由于原来网络的边是已知的, 它们可以作为事实以便评估重构的效果, 即在训练集上的误差. precision@k 和 MAP 作为评测指标. precision@k 的结果如图 5 所示, MAP 的结果如表 4 所示.

从以上结果中, 得到如下的分析和结论:

• 表 4 表明本文提出的方法在两个数据集上相较于基准方法在 MAP 上有显著提升. 图 5 表明当 k 增长的时候, 我们的方法的 precision@k 持续保持最高. 这证明了本文提出的方法能够很好地保持网络结构.

• 特别地, 对于网络 Arxiv GR-QC, 我们提出的方法的 precision@k 能够达到 100%, 同时保持

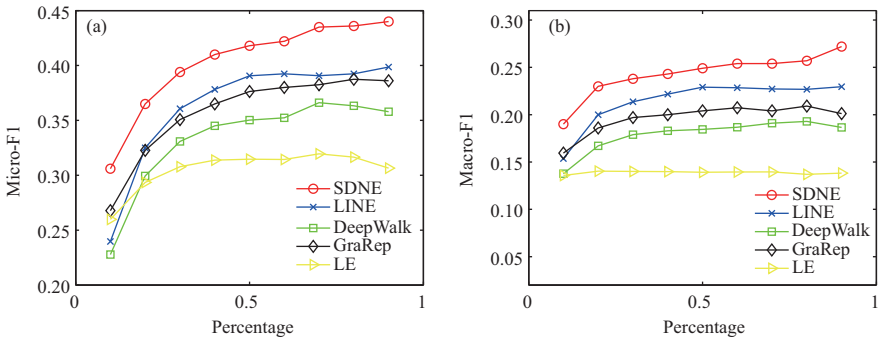

图 6　(网络版彩图) Blogcatalog 上的 (a) Micro-F1 和 (b) Macro-F1

Figure 6　(Color online) (a) Micro-F1 and (b) Macro-F1 on Blogcatalog

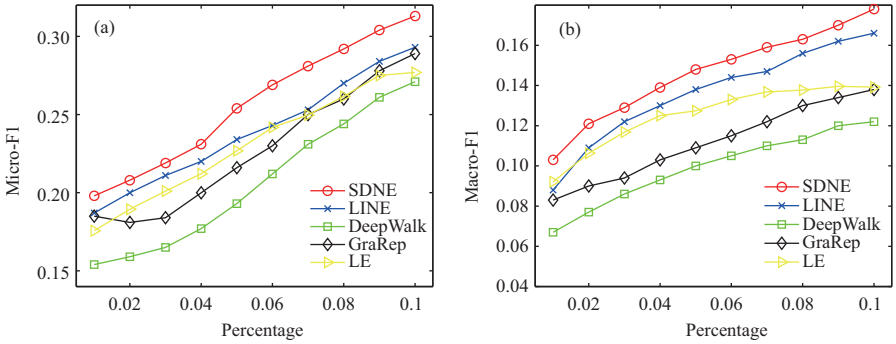

图 7　(网络版彩图) Flickr 上的 (a) Micro-F1 和 (b) Macro-F1

Figure 7　(Color online) (a) Micro-F1 and (b) Macro-F1 on Flickr

近 100% 直到 k 增加到 10000. 该结果表明我们的方法近乎在该数据集上完美重构原来的网络, 特别考虑到边的总数只有 28980 的时候.

• 虽然 SDNE 和 LINE 都利用一阶和二阶相似度来维持网络结构, SDNE 可以达到更好的重构效果. 原因有两个方面: 首先, LINE 采用浅层网络, 它很难对网络的复杂结构进行有效建模. 另外, LINE 直接将一阶和二阶的表征进行拼接, 这样的方法相比于 SDNE 的联合优化是次优的.

• SDNE 和 LINE 的结构都比 LE 好, 其中 LE 只利用了一阶相似度来保持网络结构. 这表明引入二阶相似度能够更好地保持网络结构.

2.3.6　多分类结果

分类是现实场景中一个基本而又核心的任务, 很多工作已经对它的相关算法和理论进行了很多的探索 [27]. 因此, 我们通过一个多标签分类的实验来评估不同网络嵌入表征的分类效果. 节点的表征由网络嵌入表征模型产生, 同时用作分类器的特征来将其分为一些类. 特别地, 我们采用 LIBLINEAR [28] 来训练分类器. 当训练分类器的时候, 一部分节点被随机采样为训练集, 余下节点作为测试集. 对于 Blogcatalog, 我们随机采样 10%~90% 的节点作为训练样本, 余下的节点作为测试样本. 类似地, 在 Flickr 和 Youtube 数据集上随机采样 1%~10% 的节点作为训练样本, 将其余节点作为评价分类效果的测试样本. 另外, 去除了 Youtube 数据集中没有被任何分类标注的节点. 我们重复 5 次该过程来报告平均的 Micro-F1 和 Macro-F1. 实验结果如图 6~8 所示.

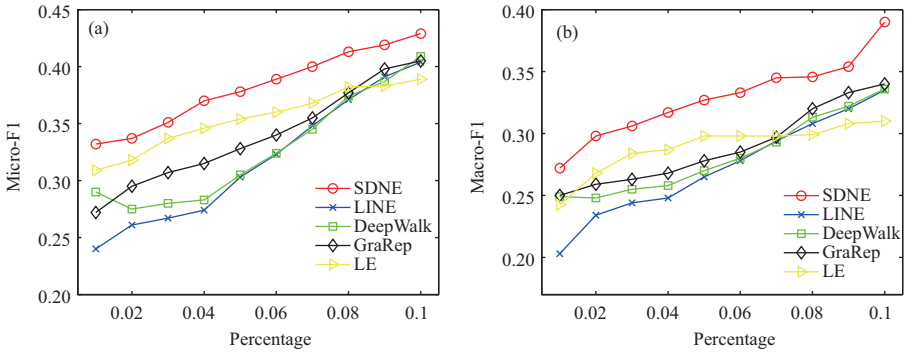

图 8 (网络版彩图) Youtube 上的 (a) Micro-F1 和 (b) Macro-F1

Figure 8 (Color online) (a) Micro-F1 and (b) Macro-F1 on Youtube

从实验结果来看, 我们有如下的观察和分析[7)]:

• 在图 6∼8 中, 本文方法的效果持续优于基准方法的效果. 这证明了本文提出的方法学到的网络表征可以很好地扩展到分类任务.

• 在图 6 中, 当训练数据的比例从 60% 降到 10% 的时候, 本文所提出方法的提升幅度远高于基准方法. 这证明了当标注数据有限的时候, 我们提出的方法相较于基准方法有显著的提升. 这样一个优势对现实应用尤其重要, 因为标注数据在真实场景中往往很稀疏.

• 在大多数情况下, DeepWalk 在所有的网络嵌入表征方法中分类效果是最差的. 我们认为原因主要来自以下两个方面: (i) DeepWalk 缺少一个明确地维持网络结构的目标函数来保证其表征的有效性; (ii) DeepWalk 的核心思想是采用随机游走的方法来拓展各个节点的邻居, 然而由于随机游走方法的随机性, 特别是对于出入度比较高的节点, 其邻居存在很大不确定性, 因此有较大概率引入噪声从而降低表征效果.

2.3.7 链接预测结果

在本小节中, 我们关注链接预测的任务, 并进行了两个实验. 在第 1 个实验中首先评估了整体链接预测的效果; 在第 2 个实验中, 评估不同稀疏的网络对不同方法链接预测的影响. 我们用 Arxiv GR-QC 的数据集来进行实验. 为了在网络中实现链接预测的任务, 随机隐藏了一部分已经存在的节点, 然后用剩余的网络来训练网络嵌入表征模型. 在训练后, 可以得到每一个节点的表征, 然后利用得到的表征来预测没有观测到的链接. 和网络重构任务不同的是链接预测的任务是预测未来的链接而不是重构已经存在的链接. 因此, 该任务展示了不同网络嵌入方法可预测性的能力. 另外, 在该任务中我们添加了 Common neighbor 的方法作为基准方法, 因为它是对链接预测有效的方法 [16]. 对于第 1 个实验, 随机隐藏了 15% 的已经存在的边 (大约 4000 条边), 然后利用 precision@k 作为评测指标来预测隐藏的边. 我们逐渐从 2∼10000 增加 k, 实验结果如表 5 所示. 最好的结果以黑体表示.

一些关于表 5 的分析和讨论如下所示:

• 实验结果表明当 k 增加时, SDNE 的效果持续优于其他的网络嵌入表征模型. 这证明了本文提出的 SDNE 模型学到的表征对于新链接的形成相较于基准方法有更好的预测能力.

• 实验结果表明本文提出的 SDNE 模型优于 Common neighbor, 这证明了我们的算法优于经典的

7) 一些前文相似的结果, 如我们提出的方法效果比 LINE 好等, 已经在前文列举和分析过. 故这里只列举一些独特的观察和分析.

表 5　Arxiv GR-QC 上链接预测的 precision@k

Table 5　precision@k on Arxiv GR-QC for link prediction

Method	P@2	P@10	P@100	P@200	P@300	P@500	P@1000	P@10000
SDNE	**1**	**1**	**1**	**1**	**1***	**0.99****	**0.91****	**0.257****
LINE	1	1	1	1	0.99	0.936	0.79	0.2196
DeepWalk	1	0.8	0.6	0.555	0.443	0.346	0.293	0.1591
GraRep	1	0.2	0.04	0.035	0.033	0.038	0.035	0.019
Common neighbor	1	1	1	0.96	0.9667	0.98	0.798	0.192
LE	1	1	0.93	0.855	0.827	0.66	0.391	0.05

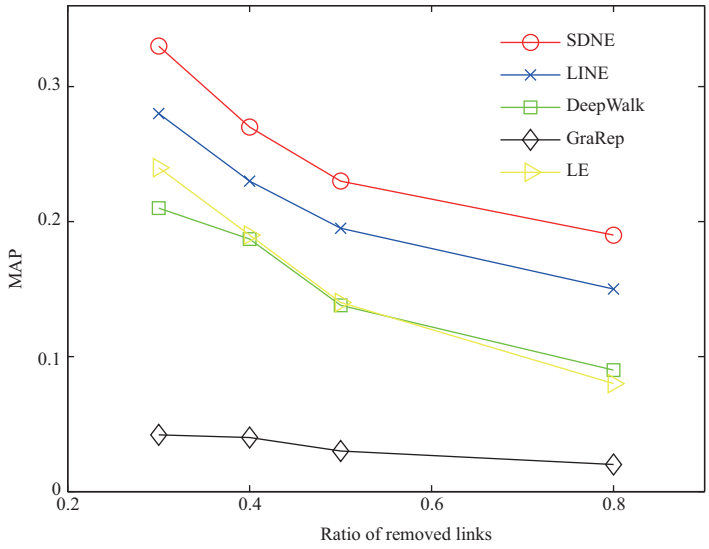

图 9　(网络版彩图) 在不同稀疏度的网络上不同网络嵌入表征模型针对链接预测的效果

Figure 9　(Color online) Link prediction performance of different network embedding algorithms on networks with different sparsities

链接预测的方法.

• 当 $k = 1000$ 时, 本文提出的方法的准确度持续高于 0.9, 但是别的方法迅速跌至 0.8 以下. 这证明了我们的方法对于排在前面的链接可以保持很高的预测准确率. 这样一个优点对于诸如检索和信息检索等实际应用十分重要, 因为用户更加关注排在前面的结果.

• Common neighbor 的结果优于 DeepWalk 和 GraRep, 这证明了二阶相似度对于刻画网络结构十分重要.

在第 2 个实验中, 通过随机去掉原来网络中的一部分链接来改变网络的稀疏度, 接着重复链接预测的步骤来报告不同网络嵌入方法的准确率, 实验结果如图 9 所示.

从图 9 中可以看出随着网络稀疏度的增加, LE 和 SDNE 或者 LE 和 LINE 之间的效果差距越来越大, 这表明引入二阶相似度可以让学到的表征对稀疏网络更加鲁棒. 进一步地, 当移除 80% 的边, 本文方法依然显著优于基准方法的效果. 这也证明了 SDNE 在处理稀疏网络方面的优越性.

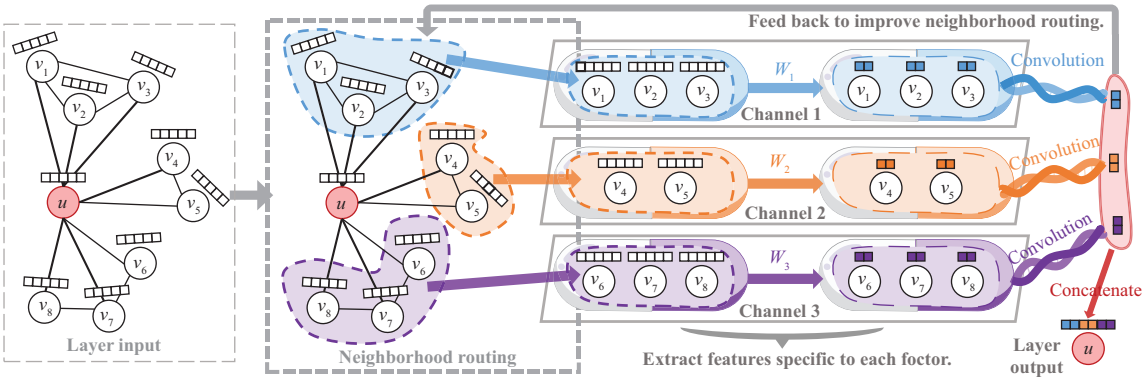

图 10 (网络版彩图) 解离化的图卷积层 (DisenConv). 它接收一个节点及其邻域、包括领域的特征向量 (可以是前一层的输出), 并为该节点输出一个解离化的表示. 邻域路由机制根据底层潜在因素对邻域进行迭代分割. 在每次迭代结束时, 信道的输出被反馈以改进路由的结果. 本例假设存在 3 个潜在因素, 因此是 3 个通道

Figure 10 (Color online) The disentangled convolutional (DisenConv) layer. It receives the feature vector of a node and its neighbors, which can be the output of the previous layer, and outputs a disentangled representation for the node. The neighborhood routing mechanism iteratively segments the neighborhood according to the underlying factors. The outputs of the channels are fed back to improve routing results at the end of each iteration. This example assumes that there are three latent factors, hence three channels

2.4 本部分小结

三元空间大数据之间复杂异构的关联关系会构成网络, 网络中的一阶关系刻画了节点对之间的直接相似度, 是一种局部的相似度, 而二阶相似度刻画了节点对之间邻居的相似度, 是一种全局相似度. 为了保持一阶和二阶相似度, 我们提出了一个多层的半监督深度网络嵌入表征模型来捕捉高度非线性的网络结构. 具体地, 我们通过深度神经网络来重构每个节点邻居向量从而保持二阶相似度, 另外, 我们用一阶相似度作为监督信息约束一对节点的隐层表征, 通过联合优化生成局部 – 全局网络结构保持的网络表征. 在多个数据集的不同任务上, 我们提出的 SDNE 模型均优于基准方法.

3 异构数据网络关联表征

三元空间大数据网络分析面临的另一个重要问题就是如何对节点自身的特征空间与节点之间的拓扑空间进行关联表征与融合分析. 在三元空间组成的现实世界中, 网络 (图) 的形成通常遵循一个复杂、异质的过程, 且这个过程是由许多潜在因素的相互作用驱动进行的. 例如, 社交网络中的一个人通常出于各种原因 (例如工作、学校) 而与他人产生联系, 因此其邻域实际上是由几个十分不同的组成部分构成的.

尽管近年来图神经网络 [29,30], 尤其是图卷积网络 [31~34], 得益于它们的深度架构和端到端学习范式, 在融合网络特征空间与拓扑空间的表征学习方面展现了卓越的性能, 现有的图神经网络在学习表征时通常采用的是整体思想: 即在学习一个节点的表征时将该节点的整个邻域视作一个整体进行操作, 而这个邻域不同部分之间的细微差别却被忽视. 正是因为现有方法采取的整体思想不能意识到, 进而理清这些异质的潜在因素, 它们学习到的表征往往不鲁棒 (例如, 作预测时容易对不相关的因素反应过度), 并且难以解释.

最近, 解离化的表征学习开始获得大量的关注, 特别是在图像表征学习领域 [35~38]. 解离表征学习的目的是学习能够分离并分别捕捉驱动数据形成的潜在因素的表征. 这种表征被证明在复杂的数据变

化过程下能保持更佳的抵抗力[39], 并且能够带来更好的泛化能力以及在面临对抗攻击时的鲁棒性[37]. 此外, 解离化的表征天然具有更好的可解释性, 因而可有助于调试和审计[40,41]. 然而, 在图神经网络的相关研究领域, 如何学习能够解离图背后的潜在因素的表征, 这仍是一个未被探索的重要领域.

三元空间网络 (图) 数据的特性对解离表征学习提出了巨大的挑战. 图的复杂形成过程要求图神经网络具有精心设计的机制, 以基于有限的可用信息, 例如节点属性或图结构, 来推理导致给定的一条边形成的潜在因素. 此外, 该机制需要设计成可导的, 以便支持端到端的训练范式; 且需要能够支持归纳学习, 以便在真实生产环境中部署后能实时地对样本外的新节点进行处理.

我们提出了解离化的图卷积网络, 这是一个端到端的深度神经网络模型, 它解决了上述挑战并能够学习解离化的节点表征. 解离化图卷积网络的关键组成部分是解离化的图卷积层, 该层含有多个解离化的并行卷积通道 (图 10). 我们提出了一种邻域路由算法, 它在解离化图卷积层中执行, 用来识别导致给定中心节点到一邻接节点的链路形成的幕后潜在因素, 并相应地将该邻接节点分发到负责该潜在因素的卷积通道. 该邻域路由算法在将中心节点及其邻居投影到几个潜在子空间后, 通过迭代地分析中心节点及其邻居所在的潜在子空间簇, 来推断出每条边背后的潜在因素. 解离化图卷积层的每个通道[42]8) 之后便从它接收到的邻接节点中提取特定于对应的各潜在因素的特征, 并独立地执行图卷积运算. 邻域路由算法完全由可求导的模块组成. 此外, 它只需要来自局部邻域的信息, 这允许我们将整个解离化图卷积层表示为一个从局部邻域快速转换到节点表征的映射函数, 从而可以支持归纳学习. 通过堆叠多个解离化图卷积层, 我们提出的解离化图卷积网络 (disentangled graph convolutional neural network, DisenGCN) 能够进一步地提取局部邻域以外的更高阶的信息. 在各种实际图数据上进行的大量实验表明, DisenGCN 可以获得可观的性能增益, 在许多情况下大约为 20% 的相对提升.

3.1 解离化图卷积网络表征

我们首先介绍了 DisenConv 层, 然后描述了 DisenGCN 的总体网络结构.

3.1.1 符号和问题表述

我们将主要关注无向图, 尽管将我们的方法扩展到有向图是很简单的. 设 $G=(V,E)$ 是一个图, 由一组节点 V 和一组边 E 组成. 使用 $(u,v)\in G$, 或 $(u,v)\in E$, 来表示节点 u 和 v 之间有一条边. $u\in V$ 都有一个特征向量 $\boldsymbol{x}_u\in\mathbb{R}^{d_{\text{in}}}$.

大多数图卷积网络 (包括我们的网络) 的关键元素是一个层 $f(\cdot)$, 当给定节点及其邻居的特征时, 该层输出节点的表示:

$$\boldsymbol{y}_u=f\left(\boldsymbol{x}_u,\{\boldsymbol{x}_v:(u,v)\in G\}\right).$$

输出 $\boldsymbol{y}_u\in\mathbb{R}^{d_{\text{out}}}$ 被视为节点 u 的表示, 由该层学习得到. 其思想是, 节点的邻域提供了丰富的信息, 可以利用这些信息更全面地描述节点.

我们的目标是导出一个层 $f(\cdot)$, 从而使输出 $\boldsymbol{y}_u$ 是一个解离化的表示. 假设有 K 潜在因素要分离. 我们希望 $\boldsymbol{y}_u$ 是由 K 个独立的成分构成的, 即 $\boldsymbol{y}_u=[\boldsymbol{c}_1,\boldsymbol{c}_2,\ldots,\boldsymbol{c}_K]$, 其中 $\boldsymbol{c}_k\in\mathbb{R}^{\frac{d_{\text{out}}}{K}}$ $(1\leqslant k\leqslant K)$. 第 k 个成分 $\boldsymbol{c}_k$ 用于描述节点 u 与因子 k 相关的方面. 关键的挑战是识别节点 u 由于因子 k 而实际连接的邻居子集, 以便更准确地描述节点 u 的第 k 方面. 为此, 我们提出 3.1.2 小节中介绍的 DisenConv 层.

8) 每个通道的输出可以看作一个胶囊[42], 因此 DisenGCN 可以看作是一种胶囊神经网络.

3.1.2 DisenConv 层

给定 $\boldsymbol{x}_u \in \mathbb{R}^{d_{\text{in}}}$ 和 $\{\boldsymbol{x}_v \in \mathbb{R}^{d_{\text{in}}} : (u,v) \in G\}$ 作为输入, 一个 DisenConv 层输出 $\boldsymbol{y}_u = [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_K] \in \mathbb{R}^{d_{\text{out}}}$, 这里 $\boldsymbol{c}_k \in \mathbb{R}^{\frac{d_{\text{out}}}{K}}$ 描述了节点 u 的第 k 个面.

DisenConv 由 K 个通道构成. 我们把 $\boldsymbol{c}_k$ 看作第 k 个通道的输出. 首先假设 $i \in \{u\} \cup \{v : (u,v) \in G\}$ 时, 通过将节点特征向量 $\boldsymbol{x}_i$ 投影到不同的子空间, K 通道可以提取不同的特性:

$$\boldsymbol{z}_{i,k} = \frac{\sigma(\boldsymbol{W}_k^{\mathrm{T}} \boldsymbol{x}_i + \boldsymbol{b}_k)}{\left\|\sigma(\boldsymbol{W}_k^{\mathrm{T}} \boldsymbol{x}_i + \boldsymbol{b}_k)\right\|_2}, \tag{12}$$

其中 $\boldsymbol{W}_k \in \mathbb{R}^{d_{\text{in}} \times \frac{d_{\text{out}}}{K}}$ 和 $\boldsymbol{b}_k \in \mathbb{R}^{\frac{d_{\text{out}}}{K}}$ 是通道 k 的参数, $\sigma(\cdot)$ 是非线性激活函数. 使用 l_2 – 标准化来确保数值稳定性, 并防止具有过丰富特征 (例如, 长文本) 的邻居扭曲我们的预测. 然后假设 $\boldsymbol{z}_{i,k}$ 大约描述了与第 k 个因子相关的节点 i 的方面, 前提是 $\boldsymbol{x}_i$ 确实包含有关方面的有意义的信息.

然而, 特征向量 $\boldsymbol{x}_i$ 在现实世界中通常是不完整的, 例如, 用户可以阅读, 但从不发布任何内容. 因此, 我们不能直接使用 $\boldsymbol{z}_{u,k}$ 作为输入节点 u 的 $\boldsymbol{c}_k$. 要全面捕获节点 u 的方面 k, 需要从邻居中挖掘信息, 即从 $\boldsymbol{z}_{u,k}$ 和 $\{\boldsymbol{z}_{v,k} : (u,v) \in G\}$ 中构造 $\boldsymbol{c}_k$.

这里的关键见解是, 在构造 $\boldsymbol{c}_k$ 来描述节点 u 的方面 k 时, 不应该使用所有的邻居. 具体地说, 由于因子 k, 我们应该只使用实际连接到节点 u 的邻居. 挑战在于设计一种机制来推断由节点 u 连接的邻居子集 (由于因子 k 导致的).

因此, 我们提出了基于两个合理假设的邻域路由机制. 第 1 个假设着眼于邻居之间的关系:

假设1 因子 k 很可能是节点 u 与其邻居的某个子集相连接的原因, 如果该子集很大, 且子集中的邻居在方面 k 上很相似, 即它们在第 k 个子空间中形成一个簇.

第 1 个假设启发我们在从原始特征空间投影的 K 个子空间中逐个搜索最大的簇. 由于不涉及 $\boldsymbol{x}_u$, 因此该假设在 $\boldsymbol{x}_u$ 有噪声或不完整的情况下是可靠的. 此外, 在寻找大簇时, 缺少因子 k 信息的邻居将被自动剪除, 因为它们的投影特征 $\boldsymbol{z}_{v,k}$ 将是噪声, 并且不会形成足够大的簇.

另一方面, 第 2 个假设侧重于节点 u 与其一个邻居之间的关系:

假设2 如果节点 u 和邻居 v 在方面 k 方面相似, 则因子 k 可能是连接的原因.

第 2 个假设是 $\boldsymbol{z}_{u,k}^{\mathrm{T}} \boldsymbol{z}_{v,k}$ 可以提供 u 和 v 之间的边背后的因子的提示, 这是可以快速计算且十分有效的, 前提是 $\boldsymbol{x}_u$ 和 $\boldsymbol{x}_v$ 包含关于因子 k 的足够信息.

当 $\boldsymbol{x}_u$ 或 $\boldsymbol{x}_v$ 缺少关于因子 k 的信息时, 假说 2 提供的提示 (尽管计算效率高) 可能会产生误导. 因此, 我们需要通过将其与假设结合来缓解这个问题. 同时, 假设 1 需要一个聚类过程, 通常需要多次迭代. 假设 2 可以作为一个强的前向引导聚类, 以实现快速收敛. 因此, 我们提出了基于假设 1 和 2 的邻域路由机制.

设 $p_{v,k}$ 为因子 k 是节点 u 到达邻居 v 的概率, 它应该满足 $p_{v,k} \geqslant 0$ 和 $\sum_{k'=1}^{K} p_{v,k'} = 1$. 那么 $p_{v,k}$ 也是我们应该使用邻居 v 来构造 $\boldsymbol{c}_k$ 的概率. 邻域路由机制将迭代地推断 $p_{v,k}$, 并构造 $\boldsymbol{c}_k$. 它首先根据假设 2 初始化 $p_{v,k}$: $p_{v,k}^{(1)} \propto \exp(\boldsymbol{z}_{v,k}^{\mathrm{T}} \boldsymbol{z}_{u,k})$.

由假设 1 启发, 然后在每个邻域应该近似只属于一个子空间簇的约束下迭代搜索每个子空间中的最大聚类:

$$\boldsymbol{c}_k^{(t)} = \frac{\boldsymbol{z}_{u,k} + \sum_{v:(u,v)\in G} p_{v,k}^{(t-1)} \boldsymbol{z}_{v,k}}{\|\boldsymbol{z}_{u,k} + \sum_{v:(u,v)\in G} p_{v,k}^{(t-1)} \boldsymbol{z}_{v,k}\|_2}, \tag{13}$$

$$p_{v,k}^{(t)} = \frac{\exp(\boldsymbol{z}_{v,k}^{\mathrm{T}} \boldsymbol{c}_k^{(t)})}{\sum_{k'=1}^{K} \exp(\boldsymbol{z}_{v,k'}^{\mathrm{T}} \boldsymbol{c}_{k'}^{(t)})}, \tag{14}$$

对于迭代 $t = 2, \ldots, T$. 最后, 它输出 $\boldsymbol{c}_k = \boldsymbol{c}_k^{(T)}$. 我们可以在这里将 $\boldsymbol{c}_k$ 视为每个子空间簇的中心. 假设 2 不仅用于初始化, 而且在每次迭代过程中也用作先验, 即式 (13) 中的 $\boldsymbol{z}_{u,k}$ 项, 以确保快速收敛.

DisenConv 层的伪代码列在算法 2 中. 它只涉及可微操作, 因此可以反向传播以计算梯度.

Algorithm 2 The proposed DisenConv layer with K channels. It performs T iterations of routing. Typically $T \approx 5$.

Input: $\boldsymbol{x}_u \in \mathbb{R}^{d_{\mathrm{in}}}$ (the feature vector of node u) and $\{\boldsymbol{x}_v \in \mathbb{R}^{d_{\mathrm{in}}} : (u, v) \in G\}$ (its neighbors' features).
Output: $\boldsymbol{y}_u \in \mathbb{R}^{d_{\mathrm{out}}}$ (the representation of node u).
Param: $\boldsymbol{W}_k \in \mathbb{R}^{d_{\mathrm{in}} \times \frac{d_{\mathrm{out}}}{K}}$, $\boldsymbol{b}_k \in \mathbb{R}^{\frac{d_{\mathrm{out}}}{K}}$, $k = 1, \ldots, K$.
for $i \in \{u\} \cup \{v : (u, v) \in G\}$ **do**
  for $k = 1, 2, \ldots, K$ **do**
    $\boldsymbol{z}_{i,k} \leftarrow \sigma(\boldsymbol{W}_k^{\mathrm{T}} \boldsymbol{x}_i + \boldsymbol{b}_k)$;
    $\boldsymbol{z}_{i,k} \leftarrow \boldsymbol{z}_{i,k} / \|\boldsymbol{z}_{i,k}\|_2$; // The k-th aspect of node i
  end for
end for
$\boldsymbol{c}_k \leftarrow \boldsymbol{z}_{u,k}, \forall k = 1, 2, \ldots, K$; // Initialize K channels
for routing iteration $t = 1, 2, \ldots, T$ **do**
  for v that satisfies $(u, v) \in G$ **do**
    $p_{v,k} \leftarrow \boldsymbol{z}_{v,k}^{\mathrm{T}} \boldsymbol{c}_k / \tau, \forall k = 1, 2, \ldots, K$;
    $[p_{v,1} \ldots p_{v,K}] \leftarrow \mathrm{softmax}([p_{v,1} \ldots p_{v,K}])$;
  end for
  for channel $k = 1, 2, \ldots, K$ **do**
    $\boldsymbol{c}_k \leftarrow \boldsymbol{z}_{u,k} + \sum_{v:(u,v)\in G} p_{v,k} \boldsymbol{z}_{v,k}$; // Update
    $\boldsymbol{c}_k \leftarrow \boldsymbol{c}_k / \|\boldsymbol{c}_k\|_2$;
  end for
end for
$\boldsymbol{y}_u \leftarrow$ the concatenation of $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_K$.

3.1.3 网络架构

在本小节中, 我们描述了用于执行节点相关任务的 DisenGCN 的总体网络架构.

设 $G = (V, E)$ 为输入图. 节点 u 与特征向量 $\boldsymbol{x}_u \in \mathbb{R}^D$ 相关联, 与真值标签 $\boldsymbol{y}_u \in \{0, 1\}^C$ 相关联, 其中 C 是类的数目. 有些图数据集不提供节点特征. 在这种情况下, 只需使用 G 的邻接矩阵的 u-th 的行作为特征 $\boldsymbol{x}_u$.

实际中, 我们可能需要堆叠多个 DisenConv 层. 首先, 这允许我们在生成节点表示时挖掘超出本地邻域的信息. 例如, 可以通过堆叠两个不连通层来利用邻居的邻居以及两个邻居之间的边. 其次, 可以通过逐渐减少后期层的通道数来潜在地学习层次表示.

因此, DisenGCN 使用 L DisenConv 层. 设 $f^{(l)}(\cdot)$ 为 l-th 层上的 DisenConv 层, 并将 u 的 $\boldsymbol{y}_u^{(l)} \in \mathbb{R}^{K^{(l)}\Delta d}$ 作为该层的输出. 这里 $K^{(l)}$ 是层 l 使用的通道数. 我们将通道的输出维度 Δd 保持在所有层上相同. 还附加了约束 $K^{(1)} \geqslant K^{(2)} \geqslant \cdots \geqslant K^{(L)}$. ReLU 用作式 (12) 中的激活函数. 然后, 层 l 的输出可以表示为

$$\boldsymbol{y}_u^{(l)} = \mathrm{dropout}(f^{(l)}(\boldsymbol{y}_u^{(l-1)}, \{\boldsymbol{y}_v^{(l-1)} : (u, v) \in G\})),$$

表 6 DisenGCN 模型使用的数据集信息统计

Table 6 Statistics of datasets used by DisenGCN

Dataset	Type	Nodes	Edges	Classes	Features	Multi-label
Citeseer	Citation network	3327	4732	6	3703	No
Cora	Citation network	2708	5429	7	1433	No
Pubmed	Citation network	19717	44338	3	500	No
Blogcatalog	Social network	10312	333983	39	–	Yes
PPI	Biological network	3890	76584	50	–	Yes
POS	Word co-occurrence	4777	184812	40	–	Yes

其中, $1 \leqslant l \leqslant L$, $\boldsymbol{y}_u^{(0)} = \boldsymbol{x}_u$ 和 $u \in V$. dropout 操作 [43] 附加在每个层之后, 仅在训练期间启用. 最后一层是完全连接的层, 即 $\boldsymbol{y}^{(L+1)} = \boldsymbol{W}^{(L+1)^{\mathrm{T}}}\boldsymbol{y}^{(L)} + \boldsymbol{b}^{(L+1)}$, $\boldsymbol{W}^{(L+1)} \in \mathbb{R}^{K^{(L)}\Delta d \times C}$, $\boldsymbol{b}^{(L+1)} \in \mathbb{R}^C$.

我们使用 $-\frac{1}{C}\sum_{c=1}^{C}\boldsymbol{y}_u(c)\ln(\hat{\boldsymbol{y}}_u(c))$, $\hat{\boldsymbol{y}}_u = \mathrm{softmax}(\boldsymbol{y}_u^{(L+1)})$, 作为单标签分类时的损失函数. 对于多标签分类, $\boldsymbol{y}_u$ 可以有多个正值, 我们使用这个损失函数: $-\frac{1}{C}\sum_{c=1}^{C}[\boldsymbol{y}_u(c)\cdot\mathrm{sigmoid}(\boldsymbol{y}_u^{(L+1)}(c)) + (1-\boldsymbol{y}_u(c))\cdot\mathrm{sigmoid}(-\boldsymbol{y}_u^{(L+1)}(c))]$. 我们通过反向传播计算梯度, 并用 Adam [44] 优化参数.

3.2 实验内容

本小节对 DisenGCN 模型在几个与节点相关的任务上的有效性进行了实证评估, 并对其在合成图上的行为进行了分析, 以获得进一步的了解.

3.2.1 实验设置

基准方法. 为了证明本文方法的优越性, 将 DisenGCN 与两种有代表性的图神经网络进行了比较, 包括图卷积网络 (GCN) [34] 和图注意网络 (GAT) [45]. 特别是, GAT 是一种最先进的图形神经网络, 用于与节点相关的任务, 其源代码是公开的. 在进行图卷积时, GCN 根据节点的度对节点的邻域进行加权, GAT 则学习一种参数化的注意机制来剪除不相关的邻域. 我们的模型包含的参数数量与 GCN 相同, 但比 GAT 的要少得多. GCN 和 GAT 的原始实现不支持多标签任务. 因此, 将它们修改为使用与我们相同的多标签损失函数, 以便在多标签任务中进行公平比较.

此外, 在实验中还包括 3 种节点嵌入表征模型, 即 DeepWalk [3], LINE [6] 和 node2vec [46], 因为它们在多标签任务中表现出很强的性能.

数据集. 我们对 6 个真实世界的图进行了实验, 它们的统计数据列在表 6 中. Citeeser, Cora 和 Pubmed [47] 用于半监督节点分类. 这 3 个节点、边和标签分别表示文章、引文和研究领域. BlogCatalog [23], PPI [46,48], POS [46] 用于多标签节点分类. 它们的标签分别是用户兴趣、生物神经状态和 POS 标签. 后 3 个图不提供节点属性. 因此, 使用它们的邻接矩阵的行来代替它们的节点属性特征.

超参数. 设 d 为图神经网络第 1 层的输出维数. 在半监督分类任务中, 遵循 GAT 并使用 $d = 64$. 在多标签分类任务中, 遵循 node2vec 并使用 $d = 128$, 同时将节点嵌入表征算法的节点嵌入维度设置为 128. DisenGCN 第 1 层的输出维度是 $K^{(1)}\Delta d$, 其中 $K^{(1)}$ 是该层使用的通道数, Δd 是每个通道的输出维度. 因此, 当 $d/K^{(1)}$ 不是整数时, 使用 $K^{(1)}\Delta d = K^{(1)}\left\lfloor d/K^{(1)}\right\rfloor$ 而不是 d 作为模型. 我们设置 $T = 7$, 也就是说, 执行 7 次迭代的邻域路由. 然后, 使用 hyperopt [49] 自动调整模型和基线的超参数. 然后利用验证集上的最佳超参数, 我们报告了在每个半监督数据集上平均运行 100 次, 在每个多

表 7 半监督节点分类准确度 (%)

Table 7 Semi-supervised classification accuracy (%)

Method	Dataset		
	Cora	Citeseer	Pubmed
MLP	55.1	46.5	71.4
ManiReg [51]	59.5	60.1	70.7
SemiEmb [52]	59.0	59.6	71.1
LP [53]	68.0	45.3	63.0
DeepWalk [3]	67.2	43.2	65.3
ICA [54]	75.1	69.1	73.9
Planetoid [50]	75.7	64.7	77.2
ChebNet [33]	81.2	69.8	74.4
GCN [34]	81.5	70.3	79.0
MoNet [55]	81.7	–	78.8
GAT [45]	83.0	72.5	79.0
DisenGCN	**83.7**	**73.4**	**80.5**

标签数据集上平均运行 30 次的性能.

3.2.2 半监督节点分类

在此任务中, 每个数据集仅包含每个类的 20 个标记实例. 因此, 在预测其余的标签时, 必须利用图结构. 我们严格遵循前面工作建立的实验设置 [34, 45, 50], 并使用与之相同的数据集分割. 结果列在表 7 [3, 33, 34, 45, 50~55] 中. 这里使用的 3 个数据集没有多标签数据集包含的因素多. 它们中的大多数节点只与同一类的邻居连接. 尽管如此, 我们的模型仍然优于基线. hyperopt 发现 DisenGCN 的最佳层数是 5, 而 GCN 和 GAT 都使用两层. 因此, 改进的性能可能源于 DisenGCN 更好地利用深层架构的能力. 通过采用分离的方法而不是整体的方法, DisenGCN 不会遭受深度 GCN [56] 所面临的过度平滑问题. 与深度 GAT 相比, 它也不容易过度拟合, 因为我们的邻域路由机制与注意机制不同, 不引入额外的参数.

3.2.3 多标签节点分类

我们遵循 node2vec [46] 的设置, 报告每个方法的性能, 同时将标记为训练的节点数从 10%$|V|$ 过渡到为 90%$|V|$, 其中 $|V|$ 是节点总数. 其余的节点被平均分割成一个验证集和一个测试集.

在图 11 的报告结果中. GCN 的宏观 F1 得分较高, 而微观 F1 得分较低, 说明它对类不平衡不具有鲁棒性, 不能很好地处理样本较少的类. 这是因为 GCN 所采用的整体方法往往忽略了少数邻居提供的信息, 这些邻居与样本数较少的类相关联. 另一方面, 由于 GAT 的参数化注意机制, GAT 可能会出现过拟合, 因为我们观察到 GAT 在验证集上的性能远远高于在测试集上的性能. 相比之下, 本文方法在很大程度上始终优于表现最好的基线, 在大多数情况下, 相对提高了约 10%~20% 的相对值. 这表明, 通过对边形成背后的因素进行分离和保留, 本文方法可以有效地解决 GCN 和 GAT 面临的上述问题.

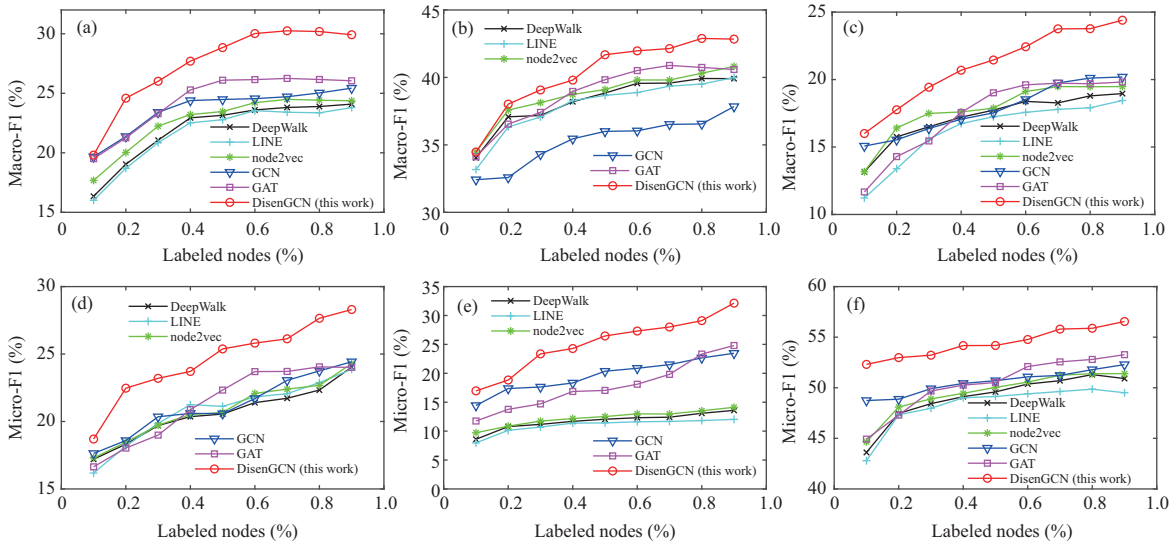

图 11 (网络版彩图) 多标签分类任务的 Macro-F1 和 Micro-F1 得分. 本文方法在很大程度上始终优于表现最好的基线, 在大多数情况下达到 10%∼20% 的相对改善

Figure 11 (Color online) Macro-F1 and Micro-F1 scores on the multi-label classification tasks. Our approach consistently outperforms the best performing baselines by a large margin, reaching 10% to 20% relative improvement in most cases. (a) Macro-F1(%), BlogCatalog; (b) Macro-F1 (%), BlogCatalog; (c) Macro-F1 (%), PPI; (d) Micro-F1(%), PPI; (e) Micro-F1 (%), POS; (f) Micro-F1(%), POS

3.2.4 解离化合成的图数据

为了进一步研究 DisenGCN 的行为, 我们生成了具有多种潜在因素的合成图. 生成一个包含 K 潜在因素的图, 我们首先生成 K Erdős-Renyi 随机图, 每个图有 1000 节点和 16 社区. 随机图中的两个节点如果在同一个社区中, 则连接概率为 p, 否则连接概率为 q. 然后, 通过对随机图的邻接矩阵求和, 生成具有 K 潜在因子的最终合成图. 我们将 q 设置为 $3e^{-5}$, 以生成大约 200 的随机边, 从而确保图是连接的. 对于每个 K 选项, 调整 p, 使平均度数在 39.5 和 40.5 之间. 邻接矩阵的行用作节点特征, 而基真值社区用作标签, 即有 $16K$ 类, 每个节点有 K 标签. 我们在这个任务中使用 $d = 64$. 为了公平比较, 我们没有手动将 DisenGCN 使用的频道数设置为 K, 而是像往常一样进行调整.

我们改变潜在因素的数量, 结果如表 8 所示. 结果表明, 随着潜在因素的数量从 4 个增加到 10 个, DisenGCN 开始有较大的相对改善, 这就强调了对潜在因素进行分离的重要性. 然而, 当 K 非常大, 即 $K > 12$ 时, 合成图变得太具有挑战性, 而 DisenGCN 带来的相对改善开始下降. 在图 12 中, 我们使用 8 个通道, 在具有 8 个因子的合成图上可视化 DisenGCN 所学习的 64 维节点表示的元素之间的相关性的绝对值. DisenGCN 的相关图显示了 8 个清晰的对角线块, 这表明 DisenGCN 的 8 个通道很可能捕获了相互排斥的信息.

3.2.5 超参数敏感性

我们研究两个对 DisenGCN 最重要的超参数: 信道数和路由迭代次数的影响. 在这里使用一个单独的 DisenConv 层, 并在一个包含 8 个潜在因子的合成图上运行实验 (参见 3.2.4 小节). 其他数据集的结果也有类似的趋势. 结果如图 13 所示. 结果表明当信道数约为实际潜在因子数时, DisenGCN 的性能最好, 并且由于其收敛性, 在饱和前进行更多迭代的路由通常会获得更好的性能.

表 8　不同潜在因子维度在合成图上的 Micro-F1 分数

Table 8　Micro-F1 scores on synthetic graphs generated with different numbers of latent factors

Method	Number of latent factors						
	4	6	8	10	12	14	16
GCN	78.78±1.52	65.73±1.94	46.55±1.55	37.37±1.52	24.49±1.03	18.14±1.50	16.43±0.92
GAT	83.77±2.32	60.89±3.75	45.88±3.79	36.72±3.58	24.77±3.47	20.89±3.57	19.53±3.97
DisenGCN	**93.84**±1.12	**74.68**±1.92	**54.57**±1.79	**43.96**±1.45	**28.17**±1.22	**23.57**±1.28	**21.99**±1.34
Relative improvement (%)	+12.02	+13.62	+17.23	+17.63	+13.73	+12.83	+12.6

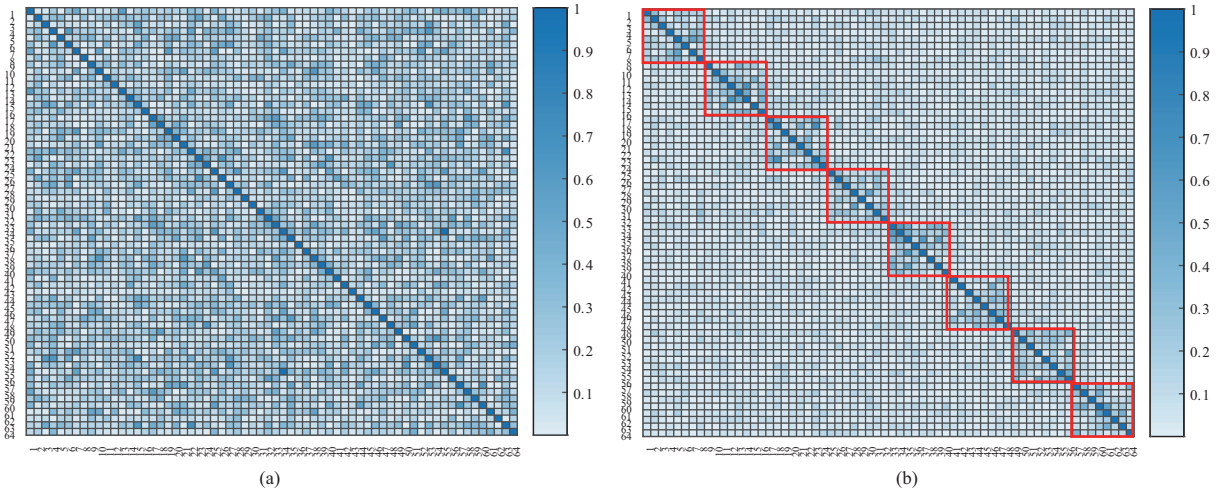

图 12　(网络版彩图) 在一个有 8 个潜在因子的合成图上, 由 GCN 学习 (a) 的 64 维表示元素与 DisenGCN (b) 的 8 个通道分离的 64 维表示元素之间的相关性的绝对值. 可以看到, DisenGCN 的 8 个通道可能捕获互斥信息, 因为其对应的图显示了 8 个对角块 (用红色标记)

Figure 12　(Color online) The absolute values of the correlations between the elements of the 64-dimensional representations learned by GCN (a) and DisenGCN (b) with eight channels, respectively, on a synthetic graph with eight latent factors. We can see that the eight channels of DisenGCN are likely capturing mutually exclusive information, because (b) exhibits eight diagonal blocks (marked in red)

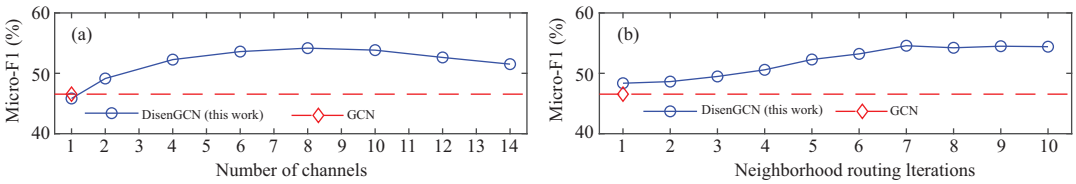

图 13　(网络版彩图) 在具有 8 个潜在因子的合成图上, 使用单个 DisenConv 层的 DisenGCN 的超参数灵敏度

Figure 13　(Color online) Hyper-parameter sensitivity of DisenGCN, using a single DisenConv layer, on synthetic graphs with eight latent factors

3.3　本部分小结

三元空间大数据网络形成的潜在因素错综复杂, 为了更好地对三元空间异构数据进行关联表征, 我们提出了解离图神经网络表征方法, 对异构数据网络形成的隐藏因子进行解离化表达, 从而得到更

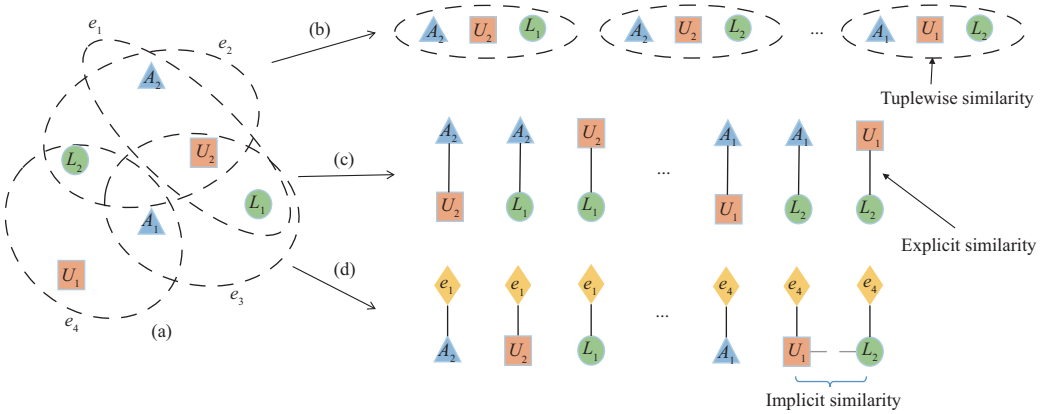

图 14 (网络版彩图) (a) 超图; (b) 本文方法; (c) 团扩展; (d) 星扩展. 本文方法把超边当做一个整体维持多元相似度. 团扩展中, 每条超边展开成一个团 (clique). 每对节点有显式的相似度. 而星扩展中, 每个超边中的节点连接一个新的代表该超边的节点. 由于它们连接相同的节点, 每对节点有隐式的相似度

Figure 14 (Color online) (a) An example of a hyper-network; (b) our method; (c) the clique expansion; (d) the star expansion. Our method models the hyperedge as a whole and the tuplewise similarity is preserved. In clique expansion, each hyperedge is expanded into a clique. Each pair of nodes has explicit similarity. As for the star expansion, each node in one hyperedge links to a new node which stands for the origin hyperedge. Each pair of nodes in the origin hyperedge has implicit similarity because they link to the same node

鲁棒、可解释的异构数据网络表征.

4 跨空间三元结构关联图表征

三元空间数据的一个重要应用就是 “什么人 (人类群体) 在什么地方 (物理地点) 做什么事情 (信息事件)”. 因此, 如何对应运而生的 “人 – 地 – 事” 三元空间网络 (图) 进行关联表征是亟待解决的首要问题. 大多数现在的网络 (图) 表征学习方法都是维持节点之间两两的关系 [57~60], 无法很好地解决 “人 – 地 – 事” 三元空间数据构成的更复杂的三元组关系. 因此, 我们采用超图 (hyper-graph) 来对这种多元关系构成的网络进行建模.

最直接的处理超图的做法就是把超图分解为多个二元关系. 团扩展 [61] (图 14(c)) 和星扩展 [62] (图 14(d)) 是两个经典的做法. 团扩展将一个超边看做一个全连接图. 而星扩展将一个超图看成是一个二部图, 原来超图中的超边也看成是二部图中的点, 如果一个点属于一条超边, 那么二部图中的点与二部图中对应超边的点相连. 这些方法明确地或者间接地假设一个超边时可分割的. 即如果把一个超边看为一些点的集合, 任意一个超边的子集仍然是一条超边. 在同构的超图中, 这个假设也许成立, 因为同构超图中的超边通常是由一个共同的相似度构成的, 比如相同的标签. 然而, 在异构超图中, 我们需要解决以下挑战:

(1) **不可分性.** 异构超图通常是不可分的. 也就是说一些节点之间有一个强的多元关系, 但是这些节点的子集并不一定有强的关系. 比如在推荐系统中, 一个用户给一个电影打了一个标签, 那么 (用户、电影、标签) 之间是一个强的关系但是 (用户、标签) 之间的关系可能就不那么强. 所以不能直接用经典的方法将超图分解为普通的图.

(2) **结构保持.** 超图中的局部结构可以由观测到的关系保持. 但是由于超图的稀疏性, 很多存在的关系并没有被观测到. 只维持局部结构并不有效. 我们需要考虑更高阶的结构比如邻居结构来克服稀疏性问题. 如何同时维持局部和全局的结构仍是一个未解决的问题.

表 9 DHNE 模型相关符号和表示

Table 9 Symbols and definitions for DHNE

Symbol	Definition
T	Number of node types
$\boldsymbol{V}=\{\boldsymbol{V}_t\}_{t=1}^{T}$	Node set
$\boldsymbol{E}=\{(v_1,v_2,\ldots,v_{n_i})\}$	Hyperedge set
$\boldsymbol{A}$	Adjacency matrix of the hyper-network
$\boldsymbol{X}_i^j$	Embedding of node i with type j
$\mathcal{S}(\boldsymbol{X}_1,\boldsymbol{X}_2,\ldots,\boldsymbol{X}_N)$	N-tuplewise similarity function
$\boldsymbol{W}_j^{(i)}$	The i-th layer weight matrix with type j
$\boldsymbol{b}_j^{(i)}$	The i-th layer biases with type j

为了解决不可分性问题, 我们设计了一个不可分的多元相似度度量函数. 这个函数直接定义在超边中的所有节点上并保证超边的子集没有被维持. 我们理论上证明了这个不可分的多元相似度度量函数不能是一个线性函数. 因此我们用一个深度神经网络来使得这个函数高度非线性. 为了解决结构保持问题, 我们设计了一个自编码器来重构节点的邻居结构, 使得有相似邻居的节点有相似的嵌入表征. 我们联合优化多元关系函数以及自编码器来同时解决上面的两个问题.

4.1 问题定义

首先对超图网络嵌入表征学习进行问题定义. 相关符号如表 9 所示. 首先给出超图的定义.

定义5 (超网络) 超网络 hyper-network 定义为一个超图 $\boldsymbol{G}=(\boldsymbol{V},\boldsymbol{E})$, 节点集合为 V, 有 T 种类型 $\boldsymbol{V}=\{\boldsymbol{V}_t\}_{t=1}^{T}$, 以及超边集合 $\boldsymbol{E}$, 其中每条超边有可能超过两个节点 $\boldsymbol{E}=\{E_i=(v_1,v_2,\ldots,v_{n_i})\}$ $(n_i\geqslant 2)$. 如果每条超边的节点数量为 2, 那么这个超网络退化为一个正常的网络. 边的类型 E_i 定义为边内所有点类型的集合. 如果 $T\geqslant 2$, 则超网络定义为异构超网络.

为了得到超边表征, 我们需要维持不可分的多元关系. 我们定义这种不可分的超边结构为超图的一阶相似度.

定义6 (超网络的一阶相似度) 超网络的一阶相似度衡量的是节点之间的多元相似度. 对任意 N 个节点 $v_1,v_2,\ldots,v_N$, 如果这 N 个节点之间有一条超边, 那么这 N 个节点的一阶相似度定义为 1, 但是 N 个节点的任意子集并没有一阶相似.

一阶相似度定义了现实世界中的多元不可分相似度. 但是, 现实的网络通常是稀疏且不全的. 只考虑一阶相似度不足以学到很好的网络表征. 我们需要考虑高阶相似度. 因此引入超网络的二阶相似度来维持全局结构.

定义7 (超网络的二阶相似度) 超网络的二阶相似度两个节点的邻居结构的相似度. 对任意节点 $v_i\in E_i$, E_i/v_i 定义为 v_i 的邻居. 如果 v_i 的邻居 $\{E_i/v_i \text{ for any } v_i\in E_i\}$ 与 v_j 的邻居相似, 那么 v_i 的表征 $\boldsymbol{x}_i$ 与 v_j 的表征 $\boldsymbol{x}_j$ 应该要相似.

比如, 在图 14(a) 中, A_1 的邻居集合是 $\{(L_2,U_1),(L_1,U_2)\}$. A_1, A_2 与它有二阶相似度因为它们有共同的邻居 (L_1,U_2).

4.2 深度超图嵌入表征

本小节介绍我们提出的深度超图嵌入 (deep hyper-network embedding, DHNE) 表征模型. 框架图

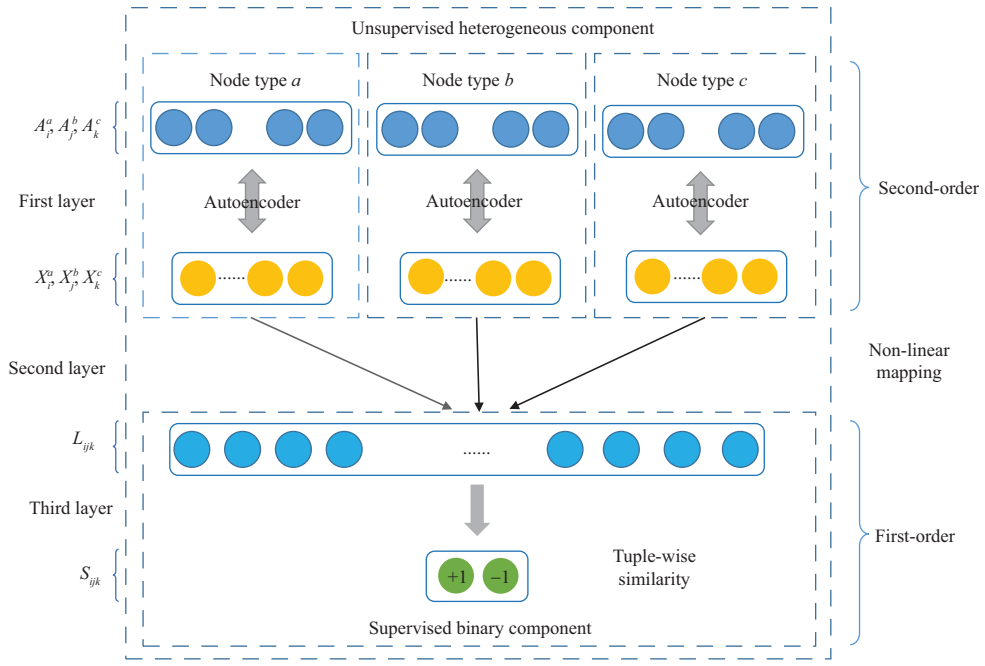

图 15 (网络版彩图) DHNE 模型框架

Figure 15 (Color online) The framework of deep hyper-network embedding

见图 15.

4.2.1 误差函数

为了维持超网络的一阶关系, 需要在表征空间定义 N 元关系度量. 如果 N 个节点构成一条超边, N 元相似度应该大, 否则要小.

性质1 记 $\boldsymbol{X}_i$ 为节点 v_i 的表征, $\mathcal{S}$ 为 N 元相似度函数.

- 如果 $(v_1, v_2, \ldots, v_N) \in \boldsymbol{E}$, 那么 $\mathcal{S}(\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N)$ 应该大 (不是一般性, 大于 l).
- 如果 $(v_1, v_2, \ldots, v_N) \notin \boldsymbol{E}$, 那么 $\mathcal{S}(\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N)$ 应该小 (不失一般性, 小于 s).

本小节中, 我们提出一个数据依赖的 N 元相似度函数, 且研究长 N 为 3 的超边, 实际上, DHNE 模型很容易处理 $N > 3$ 的情况.

这里给出一个定理表明一个线性的多元关系度量函数不能满足上面的性质 1.

定理1 线性函数 $\mathcal{S}(\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N) = \sum_i \boldsymbol{W}_i \boldsymbol{X}_i$ 无法满足性质 1.

证明 用反证法证明这个结论. 假设定理是错的, 也就是存在线性函数 $\mathcal{S}$ 满足性质 1. 我们考虑下面的反例. 假设有 3 种类型的节点, 每种类型的点有两个中心 (0 或 1). 3 个节点形成一条超边当且仅当这 3 个节点来自相同中心且类型不同. 用 $\boldsymbol{Y}_i^j$ 表示中心 i, 类型 j 的节点的表征. 由性质 1, 有

$$\boldsymbol{W}_1\boldsymbol{Y}_0^1 + \boldsymbol{W}_2\boldsymbol{Y}_0^2 + \boldsymbol{W}_3\boldsymbol{Y}_0^3 > l, \tag{15}$$

$$\boldsymbol{W}_1\boldsymbol{Y}_1^1 + \boldsymbol{W}_2\boldsymbol{Y}_0^2 + \boldsymbol{W}_3\boldsymbol{Y}_0^3 < s, \tag{16}$$

$$\boldsymbol{W}_1\boldsymbol{Y}_1^1 + \boldsymbol{W}_2\boldsymbol{Y}_1^2 + \boldsymbol{W}_3\boldsymbol{Y}_1^3 > l, \tag{17}$$

$$\boldsymbol{W}_1\boldsymbol{Y}_1^0 + \boldsymbol{W}_2\boldsymbol{Y}_1^2 + \boldsymbol{W}_3\boldsymbol{Y}_1^3 < s. \tag{18}$$

结合式 (15)∼(18), 得到 $\boldsymbol{W}_1 \times (\boldsymbol{Y}_0^1 - \boldsymbol{Y}_1^1) > l - s$ 和 $\boldsymbol{W}_1 \times (\boldsymbol{Y}_1^1 - \boldsymbol{Y}_0^1) > l - s$, 互为矛盾. 证明完成.

上述定理表明 N 元相似度函数 $\mathcal{S}$ 必须是一个非线性的函数. 这启发我们用一个多层感知机来拟合它. 这个多层感知机由两部分构成, 见图 15 中的第 2 层和第 3 层. 第 2 层是一个有非线性激活函数的全连接层. 输入节点表征 $(\boldsymbol{X}_i^a, \boldsymbol{X}_j^b, \boldsymbol{X}_k^c)$, 将它们拼接并通过一个非线性映射函数将它们映射到空间 $\boldsymbol{L}$. 它们的联合表征如下:

$$\boldsymbol{L}_{ijk} = \sigma(\boldsymbol{W}_a^{(2)} \times \boldsymbol{X}_i^a + \boldsymbol{W}_b^{(2)} \times \boldsymbol{X}_j^b + \boldsymbol{W}_c^{(2)} \times \boldsymbol{X}_k^c + \boldsymbol{b}^{(2)}), \tag{19}$$

其中 σ 是 sigmoid 函数.

得到联合表征 $\boldsymbol{L}_{ijk}$ 之后, 最后将它映射到一个概率空间得到相似度:

$$\boldsymbol{S}_{ijk} \equiv \mathcal{S}(\boldsymbol{X}_i^a, \boldsymbol{X}_j^b, \boldsymbol{X}_k^c) = \sigma(\boldsymbol{W}^{(3)} \times \boldsymbol{L}_{ijk} + \boldsymbol{b}^{(3)}). \tag{20}$$

结合上面的两层, 我们得到了一个非线性的多元相似度函数. 为了使这个相似度函数满足性质 1, 定义误差函数如下:

$$\mathcal{L}_1 = -(\boldsymbol{R}_{ijk} \log \boldsymbol{S}_{ijk} + (1 - \boldsymbol{R}_{ijk}) \log(1 - \boldsymbol{S}_{ijk})), \tag{21}$$

其中如果 v_i, v_j 和 v_k 中有一条超边, 则 $\boldsymbol{R}_{ijk}$ 为 1, 否则为 0. 从误差函数可知, 如果 $\boldsymbol{R}_{ijk}$ 等于 1, 相似度 $\boldsymbol{S}_{ijk}$ 将会大, 否则会小. 也就说, 一阶相似度得以保持.

下一步我们考虑如果维持二阶相似度. 图 15 中第 1 层就是为了保持二阶相似度. 二阶相似度衡量的是邻居的结构相似度. 这里定义超网络的邻接矩阵来刻画邻居结构. 首先, 我们给出一些超图的简单定义. 对于一个超图 $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, 如果 $v \in e$, 则 $|\boldsymbol{V}| * |\boldsymbol{E}|$ 关联矩阵 $\boldsymbol{H}$ 的元素 $\boldsymbol{h}(v, e) = 1$ 否则为 0. 对于节点 $v \in \boldsymbol{V}$, 节点的度数定义为 $d(v) = \sum_{e \in \boldsymbol{E}} \boldsymbol{h}(v, e)$. 记 $\boldsymbol{D}_v$ 表示包含节点度数的对角矩阵. 邻接矩阵 $\boldsymbol{A}$ 定义为 $\boldsymbol{A} = \boldsymbol{H}\boldsymbol{H}^{\mathrm{T}} - \boldsymbol{D}_v$, 其中 $\boldsymbol{H}^{\mathrm{T}}$ 是 $\boldsymbol{H}$ 的转置. 邻接矩阵 $\boldsymbol{A}$ 的元素表示两个节点同时出现的次数, 那么 $\boldsymbol{A}$ 的第 i 行表示节点 v_i 的邻居结构. 我们用邻接矩阵 $\boldsymbol{A}$ 作为输入特征, 一个自编码器 [63] 作为模型来维持邻居结构. 自编码器由一个编码器和一个解码器构成. 编码器是一个输入矩阵 $\boldsymbol{A}$ 到一个隐空间 $\boldsymbol{X}$ 的映射, 解码器是从隐空间 $\boldsymbol{X}$ 逆映射回原始的特征空间 $\hat{\boldsymbol{A}}$:

$$\boldsymbol{X}_i = \sigma(\boldsymbol{W}^{(1)} \times \boldsymbol{A}_i + \boldsymbol{b}^{(1)}), \tag{22}$$

$$\hat{\boldsymbol{A}}_i = \sigma(\hat{\boldsymbol{W}}^{(1)} \times \boldsymbol{X}_i + \hat{\boldsymbol{b}}^{(1)}). \tag{23}$$

自编码器的目标是最小化输入和输出的重构误差. 自编码器的重构过程使得拥有相似邻居结构的节点会有相似的隐式表征, 因此二阶相似度将会保持. 由于输入是超网络的邻接矩阵的一行, 而邻接矩阵通常非常稀疏. 为了加速我们的模型, 只重构非零元素. 重构误差如下:

$$||\mathrm{sign}(\boldsymbol{A}_i) \odot (\boldsymbol{A}_i - \hat{\boldsymbol{A}}_i)||_{\mathrm{F}}^2, \tag{24}$$

其中 sign 是符号函数.

此外, 超网络中节点通常有不同类型, 形成了异构超网络. 考虑到不同类型节点各自的性质, 我们需要对不同类型节点学习不同的表征空间. 在我们的模型中, 每个不同类型的节点使用不同的自编码器. 对所有节点, 误差如下:

$$\mathcal{L}_2 = \sum_t ||\mathrm{sign}(\boldsymbol{A}_i^t) \odot (\boldsymbol{A}_i^t - \hat{\boldsymbol{A}}_i^t)||_{\mathrm{F}}^2, \tag{25}$$

其中 t 节点类型.

为了同时保持异构超网络的一阶和二阶相似度, 联合优化式 (21) 和 (25):

$$\mathcal{L} = \mathcal{L}_1 + \alpha\mathcal{L}_2. \tag{26}$$

4.2.2 优化

用随机梯度下降 (SGD) 来优化本文模型. 关键步骤是计算参数 $\boldsymbol{\theta}$ 的偏导数. 这些导数可以很容易地由后向传播 [63] 计算. 注意到大多数现实网络中只有正边, 所以算法可能收敛到一个所有点都相似的平凡解. 为了解决这个问题, 我们像文献 [64] 一样随机采样多条负边. 整个算法框架见算法 3.

Algorithm 3 The deep hyper-network embedding (DHNE)

Require: the hyper-network $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ with adjacency matrix $\boldsymbol{A}$, the parameter α.
Ensure: hyper-network embeddings $\boldsymbol{E}$ and updated parameters $\boldsymbol{\theta} = \{\boldsymbol{W}^{(i)}, \boldsymbol{b}^{(i)}, \hat{\boldsymbol{W}}^{(i)}, \hat{\boldsymbol{b}}^{(i)}\}_{i=1}^{3}$.
1: Initial parameters $\boldsymbol{\theta}$ by a random process;
2: **while** the value of objective function do not converge **do**
3: Generate next batch form the hyperedge set $\boldsymbol{E}$;
4: Sample negative hyperedge randomly;
5: Calculate partial derivative $\partial\mathcal{L}/\partial\boldsymbol{\theta}$ with backpropagation algorithm to update $\boldsymbol{\theta}$;
6: **end while**

4.2.3 算法分析

接下来分析算法的复杂度和样本外扩展.

样本外扩展. 当新来一个数据集外的节点, 可以很容易通过它与数据集中存在节点的连接得到它的邻接向量. 我们将邻接向量输入到特定的自编码器, 然后利用式 (22) 得到节点表征. 这一步的复杂度是 $\mathcal{O}(d_v d)$, 其中 d_v 是节点 v 的度数, d 是表征空间的维度.

复杂度分析. 训练工程中, 计算梯度的复杂度是 $O((nd + dl + l)bI)$, 其中 n 是节点的数量, l 是隐层的数量, b 是批大小, I 是迭代次数. 参数 l 通常与表征空间维度 d 相关且与节点数量 n 独立. 批大小 b 通常比较小. 迭代次数 I 与节点数量 n 无关. 因此, 训练的复杂度与节点数量成线性关系.

4.3 实验内容

我们在多个现实网络数据和多个应用场景上来衡量我们的模型.

4.3.1 数据集

为了完全衡量我们模型的有效性, 使用 4 个不同类型的数据, 包括 GPS 网络、社交网络、医药网络和语义网络. 详细的信息如下:

• GPS [65]. 数据描述了一个人在特定地方做特定的事情. (用户、地点、事件) 关系用来构建超网络.

• MovieLens [66]. 数据集描述了 MovieLens[9] 上的个性化的标注事件. 每个电影至少有一个标注类别. 超边由 (用户、电影、标签) 构成.

• Drug[10]. 数据从 FDA adverse event reporting system (FAERS) 得到. 它包含报告到 FAERS 上的药物副作用事情. 我们通过 (用户、药物、症状) 关系构成超边.

9) https://movielens.org/.
10) http://www.fda.gov/Drugs/.

表 10 DHNE 模型使用的数据集信息统计

Table 10 Statistics of datasets used by DHNE

Dataset	Node type	#(V)	#(E)
GPS	(user, location, activity)	(146, 70, 5)	1436
MovieLens	(user, movie, tag)	(2113, 5908, 9079)	47957
Drug	(user, drug, reaction)	(12, 1076, 6398)	171756
WordNet	(head, relation, tail)	(40504, 18, 40551)	145966

表 11 网络重构的 AUC 值

Table 11 AUC value for network reconstruction

Method		GPS	MovieLens	Drug	WordNet
DHNE		**0.9598**	**0.9344**	**0.9356**	**0.9073**
Mean	DeepWalk	0.6714	0.8233	0.5750	0.8176
	LINE	0.8058	0.8431	0.6908	0.8365
	node2vec	0.6715	0.9142	0.6694	0.8609
	SHE	0.8596	0.7530	0.5486	0.5618
Min	DeepWalk	0.6034	0.7117	0.5321	0.7423
	LINE	0.7369	0.7910	0.7625	0.7751
	node2vec	0.6578	0.9100	0.6557	0.8387
	SHE	0.7981	0.7972	0.6236	0.5918
Tensor		0.9229	0.8640	0.7025	0.7771
HEBE		0.9337	0.8772	0.8236	0.7391

• WordNet [67]. 数据包括从 WordNet3.0 得到的 (单词、关系类别、单词) 关系.

数据集的详细统计信息见表 10.

4.3.2 参数设置

将本文方法 DHNE 与 6 个广泛使用的算法进行比较: DeepWalk [3], LINE [6], node2vec [46], spectral hypergraph embedding (SHE) [68], Tensor decomposition [69] 和 HyperEdge based embedding (HEBE) [70].

总的来说, DeepWalk, LINE 和 node2vec 是常用的二元关系网络表征学习方法. 在本文实验中, 使用图 14(c) 中的团扩展来使得超网络变为一个传统的网络, 并在这个变换后的传统网络使用这 3 个方法. SHE 是一个同构的异构超图表征学习方法. 张量分解是一个直接的维持多元关系的方法. HEBE 从异构事件数据学习节点表征. 注意到 DeepWalk, LINE, node2vec 和 SHE 只衡量二元关系. 为了使他们能应用于超网络的重构和链接预测, 我们利用超边中任两个节点的相似度的均值和最小值来表示这个多元关系.

我们设置所有方法的表征维度为 64, 对于 deepwalk 和 node2vec, 设置窗口为 10, 路径长度为 40, 每个节点的路径数量为 10. 对于 LINE, 设置负样本数量为正样本数量的 5 倍.

对于本文模型, 使用一层自编码器来维持超网络结构以及一层的全连接网络学习多元相似度函数. 全连接隐层大小设置为 192. 在 $\{0.01, 0.1, 1, 2, 5, 10\}$ 中用网络搜索出最优的参数 α. 设置学习率 ρ_0 为 0.025 并随时间线性下降.

4.3.3 网络重构

一个好的网络嵌入表征模型应该能够在表征空间维持原始网络结构. 首先在网络重构任务上衡量我们提出的方法. 通过学到的表征预测原始网络边的存在性. AUC 值 [71] 用来衡量效果的好坏. 实验结果见表 11. 从结果来看, 有如下发现:

• 本文方法相对于基线方法有明显的提升. 它表明本文方法能有效地保持原始网络结构.

• 相比于基线方法, 本文方法在更稀疏的数据上有更好的表现. 它表明本文方法对于稀疏数据更鲁棒.

• DHNE 的结果表明比假设超网络可分的方法 DeepWalk, LINE 和 SHE. 它表明维持不可分超边的重要性.

4.3.4 链接预测

链接预测广泛应用于现实场景特别是推荐系统. 本小节在所有 4 个数据上完成两个链接预测的实验. 这两个任务衡量了全局的效果与不同稀疏度对效果的影响. 我们用 AUC 作为衡量指标.

对于第 1 个任务, 随机隐藏 20% 的边做测试然后用剩余的边做训练. 训练之后得到每个节点的表征和相似度函数但是用相似度函数来预测隐藏边是否存在. 对于 GPS 这样小量而且稠密的数据集, 可以画出 ROC 曲线, 见图 16(a). 所有数据集上的 AUC 在表 12. 观测结果如下:

• 本文模型在所有数据集都有显著的提升. 它表明我们学到的表征对于未观测到的边有很强的预测能力.

• 观测 LINE, DeepWalk, SHE 和 DHNE 方法, 可以发现将不可分超边变为多条二元关系会损害学到表征的预测能力.

• Tensor 和 HEBE 有一定的维持不可分超边的能力, 本文方法与这两个方法之间的差距表明维持二阶性质的重要性.

对于第 2 个任务, 通过随机隐掉不同比例的边来更改网络的稀疏度然后重复上面的任务. 特别的, 我们在最多超边的数据医药数据上来做这个实验. 剩余边的比例从 10% 到 90%. 结果见图 16(b).

可以观查到在所有稀疏度的网络上本文方法都高于最优的基线方法. 它表明了本文方法在稀疏数据上的有效性.

4.3.5 节点分类

在本小节, 由于只有 MovieLens 和 wordnet 两个数据有标签或者类别数据, 我们在 MovieLens 数据上完成多标签分类以及在 wordnet 数据上完成多类别分类. 学习到节点表征之后, 我们用 SVM 作为分类器. 对于 MovieLens 数据集, 随机采样 10%∼90% 的节点作为训练数据. 而对于 wordnet 数据集, 这个比例选择为 1%∼10%. 平均的微观 F1 和宏观 F1 作为衡量指标. 结果见图 17.

从结果中, 有下列发现:

• 从宏观 F1 和微观 F1 曲线中, 我们的方法都优于基线. 它表明我们的方法在分类上的有效性.

• 当标签数据变丰富之后我们的方法的相对提升更大. 此外, 当标签数据非常非常稀疏的时候, 我们的方法仍然取得了很好的效果. 它表明了本文方法的鲁棒性.

4.3.6 参数敏感性

我们研究参数的不同取值对模型效果的影响以及表征维度和训练时间的关系. 特别地, 我们衡量一阶二阶误差的权重 α 和表征维度 d 取不同值的影响. 为了简单起见, 只报告医药数据上的链接预测

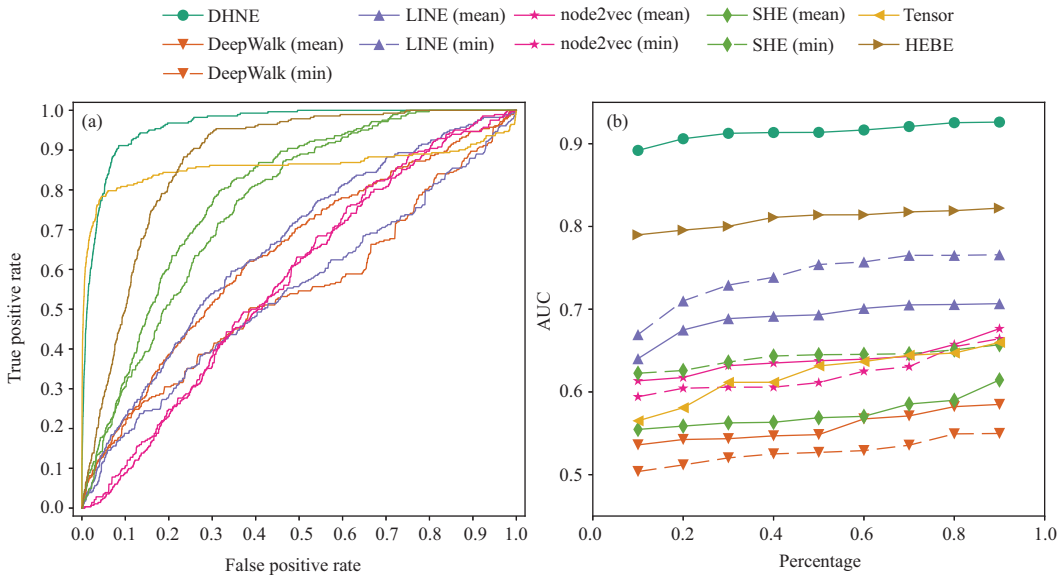

图 16　(网络版彩图) (a) GPS 数据上 ROC 曲线; (b) 不同稀疏度的网络上链接预测的效果

Figure 16　(Color online) (a) ROC curve on GPS; (b) performance for link prediction on networks of different sparsity

表 12　链接预测上的 AUC 值

Table 12　AUC value for link prediction

Method		GPS	MovieLens	Drug	WordNet
DHNE		**0.9166**	**0.8676**	**0.9254**	**0.8268**
Mean	Deepwalk	0.6593	0.7151	0.5822	0.5952
	LINE	0.7795	0.7170	0.7057	0.6819
	Node2vec	0.5835	0.8211	0.6573	0.8003
	SHE	0.8687	0.7459	0.5899	0.5426
Min	Deepwalk	0.5715	0.6307	0.5493	0.5542
	LINE	0.7219	0.6265	0.7651	0.6225
	Node2vec	0.5869	0.7675	0.6546	0.7985
	SHE	0.8078	0.8012	0.6508	0.5507
Tensor		0.8646	0.7201	0.6470	0.6516
HEBE		0.8355	0.7740	0.8191	0.6364

效果.

表征维度的影响. 我们在图 18(b) 展示不用表征维度对效果的影响. 可以看到效果曲线先上升. 这个现象是合理的, 因为更大的表征维度能利用更多的信息. 在表征维度大于 32 之后, 曲线相对平稳, 表明我们的算法对这个参数不太敏感.

参数 α 的影响. 参数 α 影响一阶和二阶之间的平衡. 可以看到结果如图 18(a). 当 α 等于 0 时, 只有一阶被保持. α 在 0.1 到 2 之间时比 α 等于 0 时效果好表征二阶信息的重要性. 而 α 在 0.1 到 2 之间时比 α 等于 5 时效果好表明一阶关系的重要性. 总结来说, 一阶和二阶相似度对于超网络表征学习都是必须的.

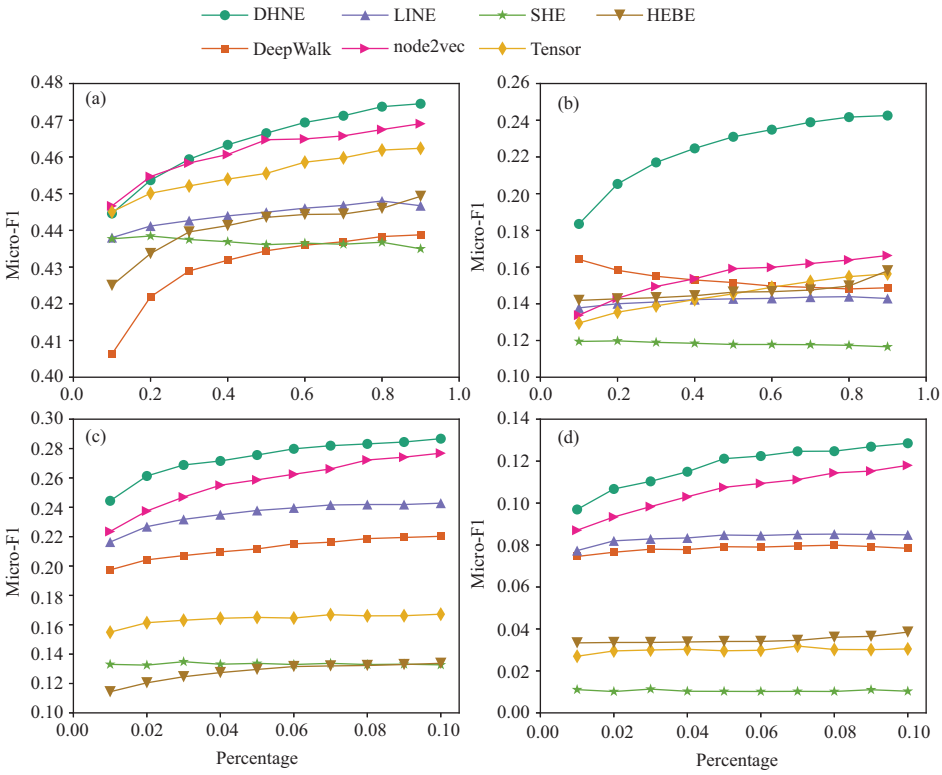

图 17 (网络版彩图) (a), (b) MovieLens 上多标签分类; (c), (d) WordNet 上的多类别分类

Figure 17 (Color online) (a), (b) Multi-label classification on MovieLens dataset; (c), (d) multi-class classification on WordNet dataset

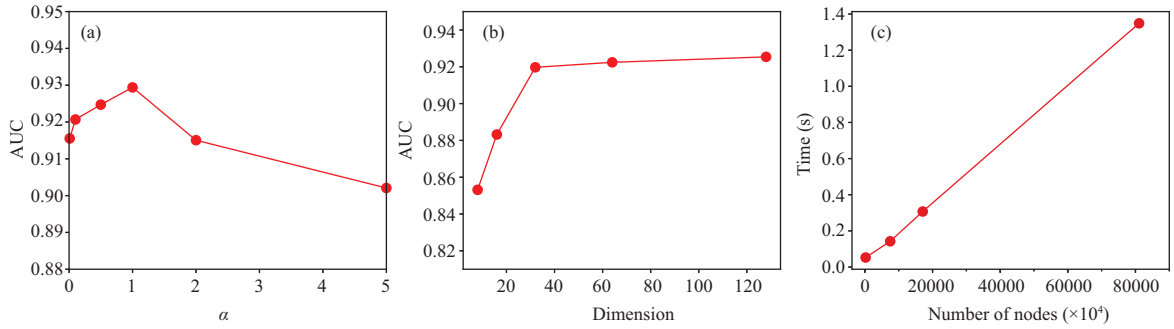

图 18 (网络版彩图) (a), (b) 效果 w.r.t. 表征维度 d, α 值; (c) 每个批次的运行时间 w.r.t. 表征维度

Figure 18 (Color online) (a), (b) Parameter w.r.t. embedding dimensions d, the value of α; (c) training time per batch w.r.t. embedding dimensions

训练时间分析. 为了测试可扩展性, 测试每个批次的训练时间, 结果见图 18(c). 可以看到训练时间随节点数量线性变化. 结果与我们之前的复杂度分析一致表明了我们模型的可扩展性.

4.4 本部分小节

本小节提出了一个新的深度模型来学习跨空间三元数据构成的超网络节点的表征. 特别地, 我们理论上证明了任意线性相似度函数不能维持超图的不可分性. 我们提出一个新的深度框架来实现一个

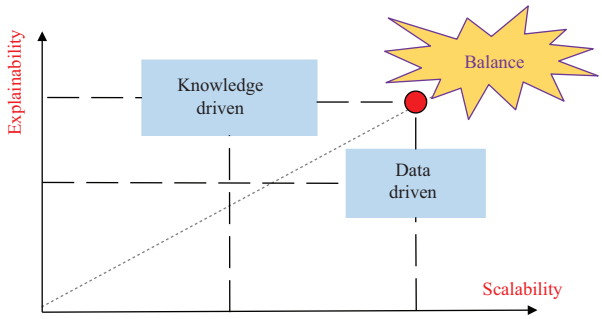

图 19 (网络版彩图) 知识驱动 v.s. 数据驱动

Figure 19 (Color online) Knowledge driven v.s. data driven

非线性的多元相似度函数. 本文方法能同时维持超网络的局部和全局结构相似度. 我们在 4 个不同类型的数据上做了丰富的实验. 实验结果表明本文算法优于现有的最好方法.

5 未来研究方向

本节中, 我们讨论在三元甚至更多元空间数据关联表征与融合分析方面的未来研究方向, 包括: (1) 引入人类的专家知识作为先验知识来指导基于数据驱动的三元空间学习模型; (2) 通过自动化模型超参数学习来减少三元空间数据挖掘对人类专家知识的依赖性; (3) 考虑自适应学习机制学习更加通用化的元知识来解决三元空间下不同任务之间的小样本或不平衡样本训练问题; (4) 辨别三元空间数据之间的因果关系以增强三元空间学习模型的可解释性与可推理性等.

5.1 知识指导与数据驱动协同的多元空间网络表征

数据驱动的方法已经取得了非常丰硕的成果, 可计算性强, 但机器学习的结果和人的学习相比往往具有解释性差或不可解释的特点. 社会科学告诉我们, 人的学习具有可解释性, 但可计算性差. 美国 IT 巨头公司谷歌 (Google) 在这方面已经做出了一些大胆而有效的尝试, 其子公司深度大脑 (DeepMind) 在 *Nature* 杂志上发表了题为 "Human-level control through deep reinforcement learning" 论文 [72], 该论文的核心算法是将深度学习与强化学习相结合, 也就是数据和人的学习机制相结合, 在深度学习中结合了人对外部数据的理解, 从而实现打游戏中与人类表现相当甚至比人厉害的算法.

因此, 将知识指导与数据驱动的协同的指导思想 (如图 19 所示) 引入三元甚至更多元空间网络表征, 也就是将知识图谱融入三元或更多元空间网络表征 [73, 74], 是一个十分有意义且值得继续深入探索的未来研究方向.

5.2 自动化三元空间网络表征

现实的人类社会 – 物理世界 – 信息空间三元空间数据中, 每个空间域都有各自独立的特性和相互之间复杂的交互关系. 三元空间关联表征学习的关键就在于如何提取得到保持这些各自独立的特性的表征以及维持它们之间复杂交互关系的融合表征. 这一步需要对三元空间数据有很深入的了解的专家知识. 未来研究的关健是能够通过自动机器学习减少人类专家的依赖, 来达到自动信息提取与融合的目的.

自动机器学习 [75~77] 的目标就是试图在有限的资源和没有人类的领域知识下自动找到尽可能优的解决方案. 自动机器学习主要可分为 3 个部分, 定义搜索空间、更快地评估配置的性能, 以及高效地

搜索最优的配置. 在三元空间表征学习中, 如何找到一个统一的框架来定义搜索空间, 更快地评估性能以及搜索效率以便更好地进行自动信息提取与融合是一个很有挑战的问题.

5.3 基于自适应元学习的三元空间大数据网络表征

一方面, 现有的方法大多数只考虑静态的图结构, 而现实中三元空间数据通常随环境动态变化, 比如交通网络不同时间段不同地区的车流量变化等. 三元空间的时空复杂性以及环境动态性给研究带来了很大的难题. 我们需要考虑如何动态高效自适应地学习三元空间大数据的表征, 提出能够进行自适应计算与在线动态学习的模型方法.

另一方面, 我们在直接处理涉及三元空间的复杂任务时, 面临的一大困难就是承载这些复杂任务的系统 (如城市交通系统、健康医疗系统等) 通常极其庞大, 会产生大规模的数据用于模型训练. 当前大多数机器学习对于新任务都只能从头开始训练. 为了利用过去已有的知识经验, 元学习 [78, 79] 利用过去的知识来调整模型. 元学习可以解决极少样本下训练问题, 被认为是通向一般人工智能的重要一步. 而在三元空间中, 不同的任务可能归属于不同的学科, 它们之间的知识融合难度非常大, 给元学习的引入带来了很大的困难.

5.4 可解释可推理的三元空间大数据网络表征

机器学习的方法通常不具备可解释性与可推理性, 这使得它们在包括医学、自动驾驶等高风险任务的应用方面受到了极大限制 [80]. 而目前的三元空间关联表征学习, 只能学到实体之间的关联关系, 而不能学到可解释可推理的因果关系. 近年来, 可解释可推理人工智能因其在高风险任务上的巨大应用价值得到了越来越多计算机领域学者的注意 [81].

建立具有因果约束的可解释、可推理三元空间大数据分析理论方法体系, 发展新一代以人为本的三元空间大数据智能计算范式, 势在必行. 因此, 在三元空间大数据网络表征中引入可解释性与可推理性, 使得我们能在三元空间中知其然也知其所以然也是一个很重要的未来研究方向.

6 总结

本文研究三元空间大数据异构信息关联表征难题, 指出由信息空间、物理世界, 以及人类社会组成的三元空间产生的海量数据呈现出了跨时空、多尺度、动态关联等特性, 已经无法采用现有的信息特征表达方法对其进行高效处理. 为了解决上述难题, 本文提出三元空间下的大数据网络关联表征方法, 通过拓扑图理论将三元空间大数据关联关系表示成网络或图的形式, 采用深度计算理论对三元空间异构大数据进行弱先验关联表征, 借助解离化理论对承载特征信息与拓扑信息的三元空间异构网络进行深度可解释表征, 利用度量学习等统计学手段与拓扑学习相结合的思想实现三元空间大数据异构关系的关联结构构建与统一空间上的精准信息表征. 最后, 本文对未来研究方向进行展望, 指出包括知识与数据双驱动、自动机器学习与元学习, 以及可解释与可推理机器学习理论在三元空间大数据分析方面的潜在应用.

致谢 我们感谢崔鹏、王岱鑫、涂珂、马坚鑫对本文的贡献.

参考文献

1 Luo D J, Ding C H Q, Nie F P, et al. Cauchy graph embedding. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011. 553–560

2 Shaw B, Jebara T. Structure preserving embedding. In: Proceedings of the 26th Annual International Conference on Machine Learning, 2009. 937–944
3 Perozzi B, Al-Rfou R, Skiena S. DeepWalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014. 701–710
4 Tenenbaum J B. A global geometric framework for nonlinear dimensionality reduction. Science, 2000, 290: 2319–2323
5 Belkin M, Niyogi P. Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput, 2003, 15: 1373–1396
6 Tang J, Qu M, Wang M Z, et al. LINE: large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web, 2015. 1067–1077
7 Tian F, Gao B, Cui Q, et al. Learning deep representations for graph clustering. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence, 2014. 1293–1299
8 Vishwanathan S V N, Schraudolph N N, Kondor R, et al. Graph kernels. J Mach Learn Res, 2010, 11: 1201–1242
9 Zhuang J, Tsang I W, Hoi S. Two-layer multiple kernel learning. In: Proceedings of International Conference on Artificial Intelligence and Statistics, 2011. 909–917
10 Bengio Y. Learning deep architectures for AI. FNT Mach Learn, 2009, 2: 1–127
11 Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks. In: Proceedings of Advances in Neural Information Processing Systems, 2012. 1097–1105
12 Socher R, Perelygin A, Wu J Y, et al. Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP), 2013
13 Hinton G, Deng L, Yu D, et al. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process Mag, 2012, 29: 82–97
14 Dash N S. Context and contextual word meaning. SKASE J Theor Linguist, 2008, 5: 21–31
15 Jin E M, Girvan M, Newman M E J. Structure of growing social networks. Phys Rev E, 2001, 64: 046132
16 Liben-Nowell D, Kleinberg J. The link-prediction problem for social networks. J Am Soc Inf Sci, 2007, 58: 1019–1031
17 Salakhutdinov R, Hinton G. Semantic hashing. Int J Approx Reason, 2009, 50: 969–978
18 Jamali M, Ester M. A matrix factorization technique with trust propagation for recommendation in social networks. In: Proceedings of the 4th ACM Conference on Recommender Systems, 2010. 135–142
19 Wang X, Hoi S C, Ester M, et al. Learning personalized preference of strong and weak ties for social recommendation. In: Proceedings of the 26th International Conference on World Wide Web, 2017. 1601–1610
20 Wang X, Zhu W W, Liu C H. Social recommendation with optimal limited attention. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019. 1518–1527
21 Hinton G E, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets. Neural Comput, 2006, 18: 1527–1554
22 Erhan D, Bengio Y, Courville A, et al. Why does unsupervised pre-training help deep learning? J Mach Learn Res, 2010, 11: 625–660
23 Tang L, Liu H. Relational learning via latent social dimensions. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009
24 Tang L, Liu H. Scalable learning of collective behavior based on sparse social dimensions. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, 2009. 1107–1116
25 Leskovec J, Kleinberg J, Faloutsos C. Graph evolution: densification and shrinking diameters. ACM Trans Knowl Discov Data, 2007, 1: 2
26 Cao S S, Lu W, Xu Q K. Grarep: learning graph representations with global structural information. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, 2015. 891–900
27 Liu T L, Tao D C. Classification with noisy labels by importance reweighting. IEEE Trans Pattern Anal Mach Intell, 2016, 38: 447–461
28 Fan R E, Chang K W, Hsieh C J, et al. Liblinear: a library for large linear classification. J Mach Learn Res, 2008, 9: 1871–1874
29 Gori M, Monfardini G, Scarselli F. A new model for learning in graph domains. In: Proceedings of IEEE International Joint Conference on Neural Networks, 2005
30 Scarselli F, Gori M, Tsoi A C, et al. The graph neural network model. IEEE Trans Neural Netw, 2009, 20: 61–80
31 Bruna J, Zaremba W, Szlam A, et al. Spectral networks and locally connected networks on graphs. In: Proceedings

of the 2nd International Conference on Learning Representations, 2014
32 Henaff M, Bruna J, LeCun Y. Deep convolutional networks on graph-structured data. 2015. ArXiv:1506.05163
33 Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. In: Proceedings of Conference on Neural Information Processing Systems, 2016
34 Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks. In: Proceedings of the 5th International Conference on Learning Representations, 2017
35 Higgins I, Matthey L, Pal A, et al. beta-VAE: Learning basic visual concepts with a constrained variational framework. In: Proceedings of the 4th International Conference on Learning Representations, 2016
36 Chen X, Kingma D P, Salimans T, et al. Variational lossy autoencoder. In: Proceedings of the 5th International Conference on Learning Representations, 2017
37 Alemi A A, Fischer I, Dillon J V, et al. Deep variational information bottleneck. In: Proceedings of the 5th International Conference on Learning Representations, 2017
38 Kim H, Mnih A. Disentangling by factorising. 2018. ArXiv:1802.05983
39 Bengio Y, Courville A, Vincent P. Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell, 2013, 35: 1798–1828
40 Doshi-Velez F, Kim B. Towards a rigorous science of interpretable machine learning. 2017. ArXiv:1702.08608
41 Lipton Z C. The mythos of model interpretability. Commun ACM, 2018, 61: 36–43
42 Hinton G E, Krizhevsky A, Wang S D. Transforming auto-encoders. In: Proceedings of the 21st International Conference on Artificial Neural Networks, 2011
43 Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res, 2014, 15: 1929–1958
44 Kingma D P, Ba J. Adam: a method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations, 2015
45 Veličković P, Cucurull G, Casanova A, et al. Graph attention networks. In: Proceedings of the 6th International Conference on Learning Representations, 2018
46 Grover A, Leskovec J. node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2016
47 Sen P, Namata G, Bilgic M, et al. Collective classification in network data. AI magazine, 2008. http://www.cs.iit.edu/~mbilgic/pdfs/umtr08.pdf
48 Breitkreutz B J, Stark C, Reguly T, et al. The BioGRID interaction database: 2008 update. Nucleic Acids Res, 2007, 36: 637–640
49 Bergstra J, Yamins D, Cox D D. Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures. In: Proceedings of the 30th International Conference on Machine Learning, 2013
50 Yang Z, Cohen W W, Salakhutdinov R. Revisiting semi-supervised learning with graph embeddings. In: Proceedings of the 33rd International Conference on Machine Learning, 2016
51 Belkin M, Niyogi P, Sindhwani V. Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. J Mach Learn Res, 2006, 7: 2399–2434
52 Weston J, Ratle F, Mobahi H, et al. Deep learning via semi-supervised embedding. In: Neural Networks: Tricks of the Trade. Berlin: Springer, 2012
53 Zhu X, Ghahramani Z, Lafferty J D. Semi-supervised learning using Gaussian fields and harmonic functions. In: Proceedings of International Conference on Machine Learning, 2003
54 Lu Q, Getoor L. Link-based classification. In: Proceedings of International Conference on Machine Learning, 2003
55 Monti F, Boscaini D, Masci J, et al. Geometric deep learning on graphs and manifolds using mixture model CNNs. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2017
56 Li Q M, Han Z C, Wu X M. Deeper insights into graph convolutional networks for semi-supervised learning. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, 2018
57 Ou M D, Cui P, Pei J, et al. Asymmetric transitivity preserving graph embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016. 1105–1114
58 Zhang Z W, Cui P, Wang X, et al. Arbitrary-order proximity preserved network embedding. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018. 2778–2786

59 Zhu D Y, Cui P, Wang D X, et al. Deep variational network embedding in wasserstein space. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018. 2827–2836

60 Tu K, Cui P, Wang X, et al. Deep recursive network embedding with regular equivalence. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018. 2357–2366

61 Sun L, Ji S W, Ye J P. Hypergraph spectral learning for multi-label classification. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008. 668–676

62 Agarwal S, Branson K, Belongie S. Higher order learning with graphs. In: Proceedings of the 23rd International Conference on Machine Learning, 2006. 17–24

63 LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015, 521: 436–444

64 Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality. In: Proceedings of Advances in Neural Information Processing Systems, 2013. 3111–3119

65 Zheng V W, Cao B, Zheng Y, et al. Collaborative filtering meets mobile recommendation: a user-centered approach. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence, 2010. 236–241

66 Harper F M, Konstan J A. The movielens datasets: history and context. ACM Trans Interact Intell Syst, 2016, 5: 19

67 Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data. In: Proceedings of Advances in Neural Information Processing Systems, 2013. 2787–2795

68 Zhou D Y, Huang J Y, Schölkopf B. Learning with hypergraphs: clustering, classification, and embedding. In: Proceedings of Advances in Neural Information Processing Systems, 2006. 1633–1640

69 Kolda T G, Bader B W. Tensor decompositions and applications. SIAM Rev, 2009, 51: 455–500

70 Gui H, Liu J L, Tao F B, et al. Large-scale embedding learning in heterogeneous event data. In: Proceedings of IEEE International Conference on Data Mining, 2016

71 Fawcett T. An introduction to ROC analysis. Pattern Recogn Lett, 2006, 27: 861–874

72 Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. Nature, 2015, 518: 529–533

73 Wu F, Song J, Yang Y, et al. Structured embedding via pairwise relations and long-range interactions in knowledge base. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence, 2015

74 Zhuang Y T, Wu F, Chen C, et al. Challenges and opportunities: from big data to knowledge in ai 2.0. Front Inform Technol Electron Eng, 2017, 18: 3–14

75 Feurer M, Klein A, Eggensperger K, et al. Efficient and robust automated machine learning. In: Proceedings of Advances in Neural Information Processing Systems, 2015. 2962–2970

76 Zhu W W, Wang X, Zhang W P. Automl and meta-learning for multimedia. In: Proceedings of the 27th ACM International Conference on Multimedia, 2019. 2699–2700

77 Hutter F, Kotthoff L, Vanschoren J. Automated Machine Learning. Berlin: Springer, 2019

78 Balte A, Pise N, Kulkarni P. Meta-learning with landmarking: a survey. Inte J Comput Appl, 2014, 105: 49–51

79 Lemke C, Budka M, Gabrys B. Metalearning: a survey of trends and technologies. Artif Intell Rev, 2015, 44: 117–130

80 Lundberg S M, Nair B, Vavilala M S, et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. Nat Biomed Eng, 2018, 2: 749–760

81 Holzinger A. From machine learning to explainable AI. In: Proceedings of World Symposium on Digital Intelligence for Systems and Machines (DISA), 2018. 55–66

Cyber-physical-human big data correlational representation

Wenwu ZHU* & Xin WANG

Department of Computer Science and Technology, Tsinghua University, Beijing 100084*, China*
* Corresponding author. E-mail: wwzhu@tsinghua.edu.cn

Abstract Cyber-physical-human ternary space is composed of information space, physical world and human society. Cyber-physical-human ternary space big data consists of the Internet data generated by the information space, the IoT data generated in physical space and data generated in human social space. This paper introduces the associated complexity of ternary spatial big data. Moreover, in view of the inherent difficulties lying in the correlation complexity of big data in ternary space, we propose to solve the key scientific problem of correlational representation for heterogeneous ternary space data. We take full advantages of deep representations obtained by deep architectures to model the heterogeneous correlational relationship of ternary space data, and establish a joint data-knowledge dual driven ternary space big data analysis theory. Last but not least, we present our insights on future research directions that deserve further investigation in the area of cyber-physical-human ternary space big data analysis.

Keywords cyber-physical-human space, big data, correlational representation, deep representation, network representation

Wenwu ZHU is currently a professor and deputy head of Computer Science Department of Tsinghua University and vice dean of National Research Center on Information Science and Technology. Prior to his current post, he was a senior researcher and research manager at Microsoft Research Asia. He was the chief scientist and director at Intel Research China from 2004 to 2008. He worked at Bell Labs New Jersey as a member of technical staff during 1996–1999. His current research interests are in the areas of cross-media big data and intelligence, and multimedia edge computing.

Xin WANG is currently an assistant professor at the Department of Computer Science and Technology, Tsinghua University. He got both of his Ph.D. and B.E degrees in computer science and technology from Zhejiang University, China. He also holds his Ph.D. degree in computing science from Simon Fraser University, Canada. His research interests include cross-modal multimedia intelligence and inferable recommendation in social media.