软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2021,32(3):636–649 [doi: 10.13328/j.cnki.jos.006186]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

图嵌入算法的分布式优化与实现*

张文涛[1,2], 苑 斌[1], 张智鹏[1], 崔 斌[1]

[1](高可信软件技术教育部重点实验室(北京大学),北京 100871)

[2](腾讯科技(北京)有限公司数据平台部,北京 100193)

通讯作者: 崔斌, E-mail: bin.cui@pku.edu.cn

摘 要: 随着人工智能时代的到来,图嵌入技术被越来越多地用来挖掘图中的信息.然而,现实生活中的图通常很大,因此,分布式图嵌入技术得到了广泛的关注.分布式图嵌入算法面临着两大难点:(1) 图嵌入算法多种多样,没有一个通用的框架能够描述大部分的算法;(2) 现在的分布式图嵌入算法扩展性不足,当处理大图时性能较低.针对以上两个挑战,首先提出一个通用的分布式图嵌入框架,具体地,将图嵌入算法中的采样流程和训练流程进行解耦,使得框架能够较好地表达多种不同的算法;其次,提出一种基于参数服务器的模型切分嵌入策略,具体地,将模型分别切分到计算节点和参数服务器上,同时使用数据洗牌的操作保证计算节点之间没有模型交互,从而减少了分布式计算中的通信开销.基于参数服务器实现了一种原型系统,并且用充分的实验证明了在不损失精度的前提下,基于模型切分的策略能够比基线系统取得更好的性能.

关键词: 分布式机器学习;图嵌入;网络优化

中图法分类号: TP181

中文引用格式: 张文涛,苑斌,张智鹏,崔斌.图嵌入算法的分布式优化与实现.软件学报,2021,32(3):636–649. http://www.jos.org.cn/1000-9825/6186.htm

英文引用格式: Zhang WT, Yuan B, Zhang ZP, Cui B. Distributed optimization and implementation of graph embedding algorithms. Ruan Jian Xue Bao/Journal of Software, 2021,32(3):636–649 (in Chinese). http://www.jos.org.cn/1000-9825/6186.htm

Distributed Optimization and Implementation of Graph Embedding Algorithms

ZHANG Wen-Tao[1,2], YUAN Bin[1], ZHANG Zhi-Peng[1], CUI Bin[1]

[1](Key Laboratory of High Confidence Software Technologies of Ministry of Education (Peking University), Beijing 100871, China)

[2](Department of Data Platform, Tencent Technology (Beijing) Co., Ltd., Beijing 100193, China)

Abstract: With the advent of artificial intelligence, graph embedding techniques are more and more used to mine the information from graphs. However, graphs in real world are usually large and distributed graph embedding is needed. There are two main challenges in distributed graph embedding. (1) There exist many graph embedding methods and there is not a general framework for most of the embedding algorithms. (2) Existing distributed implementations of graph embedding suffer from poor scalability and perform bad on large graphs. To tackle the above two challenges, a general framework is firstly presented for distributed graph embedding. In detail, the process of sampling and training is separated in graph embedding such that the framework can describe different graph embedding methods. Second, a parameter server-based model partitioning strategy is proposed—the model is partitioned to both workers and servers and shuffling is used to ensure that there is no model exchange among workers. A prototype system is implemented on parameter server

* 基金项目: 国家重点研发计划(2018YFB1004403); 国家自然科学基金(61832001); 北京大学-腾讯协同创新实验室项目

Foundation item: National Key Research and Development Program of China (2018YFB1004403); National Natural Science Foundation of China (61832001); PKU-Tencent Joint Research Lab

本文由“支撑人工智能的数据管理与分析技术”专刊特约编辑陈雷教授、王宏志教授、童咏昕教授、高宏教授推荐.

收稿时间: 2020-08-23; 修改时间: 2020-09-03; 采用时间: 2020-11-06; jos 在线出版时间: 2021-01-21

and solid experiments are conducted to show that partitioning-based strategy can get better performance than all baseline systems without loss of accuracy.

Key words: distributed machine learning; graph embedding; network optimization

图数据在现实生活中普遍存在,例如社交网络、交通网络、电商网络等等.随着人工智能时代的到来,研究人员开始越来越多地使用机器学习技术挖掘图上的信息[1−3],例如图嵌入(graph embedding)技术.另外,随着数据来源的增加,例如电商、传媒、社区、论坛等,人们所面临的数据规模也在不断增加.据统计,谷歌与脸书的数据已经达到 PB 量级,每日数据增加量可达几十到几千 TB.因此,在这种环境下,对大数据的处理与迭代能力显得极为重要.

在这种大数据和机器学习的双重背景下,大规模的图嵌入算法也越来越受到人们的关注与重视.伴随着人们对业务场景时间要求的紧迫性与互联网提供服务的及时性,研究分布式图嵌入算法的性能优化也是当下的趋势.开发高效、大规模的分布式图嵌入系统需要考虑通用性和高效性两个方面.

- 通用性:目前,研究人员针对不同的场景开发了各种各样的图嵌入算法.在进行图嵌入算法的选择时,往往需要互相比较,互相借鉴,选择最适用当前场景的算法进行使用.另外,在开发新算法的同时,需要更多地考虑如何进行改进采样的部分,以适应更好的业务需求.针对以上需求,需要对现有的算法进行总结与归纳,通过抽象,将采样模块暴露给用户,并透明化后续训练模块,需要支持后续算法更好的可扩展性,需要让算法的更新与维护变得更加方便快捷;
- 高效性:互联网线上场景的需求需要算法的及时更新与快速迭代,往往对训练时间提出了更紧迫的要求,因此需要更好的分布式策略作为支撑[4,5].此外,工业界的集群数目众多,机器数目多达几千台,在这种集群规模下,需要更好的可扩展性.目前,在工业界针对图嵌入算法有两种主流的分布式策略:一种为图简易嵌入分布式策略,另一种为图点积嵌入分布式策略.其中:图点积嵌入分布式策略缺乏良好的可扩展性(scalability),而图简易嵌入分布式策略具有性能瓶颈,需要进一步地优化方案.

为了解决以上两个挑战,本文首先总结了目前基于随机游走的算法和其他类型的一部分算法,包括深度游走算法(DeepWalk)[6]、大规模信息网络嵌入算法(LINE)[7]、非对称性临近可扩展图嵌入算法(APP)[8]、图节点向量算法(Node2Vec)[9]等.笔者发现:利用点对描述进行图嵌入表示的输入是一种良好的抽象,这种抽象能够与目前的主流词向量[10]与负采样[11]模型完美兼容,并基于此提出了基于高性能高可复用性的分布式框架.该框架目前支持 DeepWalk,LINE,APP 这 3 种采样算法,并且在对后续算法的更新上起到很好的支持作用.另外,笔者针对参数服务器上的分布式实现提供了性能优化的策略,对已有的两种在 Angel[12,13]上实现的工业界级别的分布式策略进行了详细的分析(此外,高维度图计算平台还有 AliGraph[14]).提出了一种模型划分的策略,支持更好的可扩展性.

本文的贡献可以概括如下:对目前多种图嵌入算法进行了抽象,通过将采样和训练两部分解耦,提出了一套高可复用性的图嵌入框架;通过对已有两种分布式策略实现的分析,发现这两种策略在扩展性上表现不足,进而提出了一种基于图分割嵌入分布式策略;实现了一套分布式图嵌入算法的原型系统,并用充分的实验证明了图分割嵌入分布式策略的高效性.

本文第 1 节介绍相关工作.第 2 节介绍高可复用性的图嵌入框架.第 3 节介绍图分割嵌入式策略.第 4 节介绍实验结果.最后,第 5 节总结全文.

1 相关工作

深度游走算法是图嵌入算法中较早提出来的基于随机游走的算法,其借鉴了词向量算法.首先选择某一特定点为起始顶点,通过深度优先搜索或者宽度优先搜索得到图的序列;再将该序列送入词向量算法进行学习,得到该点的表示向量.DeepWalk 算法解决了两大问题:(1) 随机游走采样节点的序列,形成图节点序列;(2) 利用 Skip-gram[15]算法结合负采样进行图节点向量的计算.

非对称临近可扩展性图嵌入(scalable graph embedding for asymmetric proximity)提出一种保留图的非对称结构的基于随机游走的图嵌入方法,并通过理论推导证明了该嵌入式方法保留了根网络排名(rooted pagerank)值,DeepWalk 算法和节点向量算法都无法保留图的非对称性结构(有向图和无向图).

LINE 是图嵌入算法中较常用的算法,其适用任意类型的信息网络、无向图、有向图、无权图、有权图等,优化了目标函数,使得其能够保留局部和全局网络结构.LINE 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,提高了算法的效率和有效性.

分布式的图嵌入算法利用参数服务器[16−19]进行实现,目前在工业界的应用有两种:一种为图简易嵌入分布式策略,另一种为图点积嵌入分布式策略.两种策略将在第 3 节详细说明.

2 高可复用的图嵌入框架

2.1 已有图嵌入算法分析

图嵌入算法很多,例如 DeepWalk、LINE、APP、节点向量算法等.其原始训练流程如图 1 所示,以 3 种算法为例,其采样算法模块与后续训练模块相互耦合,从而导致代码复用性差,重复度高.

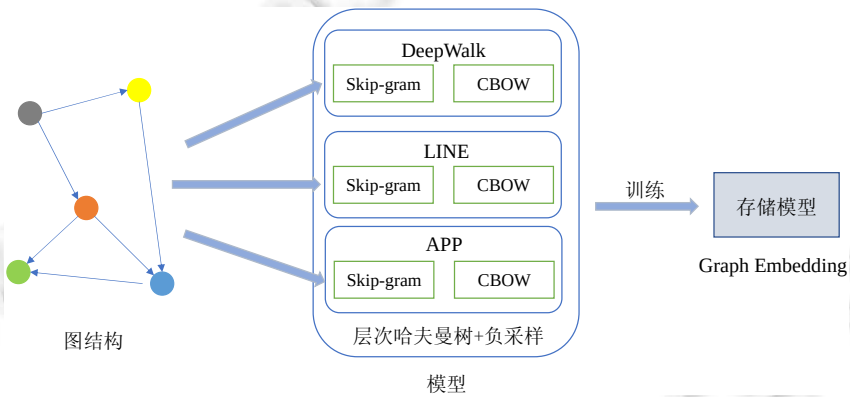


Fig.1 The original training process

图 1 原始训练流程

DeepWalk 基于图上的随机游走进行采样,生成一系列的序列信息,利用采样得到的序列信息进行图嵌入算法的训练.LINE 采用的采样算法与 DeepWalk 截然不同,其收集边信息,利用每条边直接生成点对信息,利用边信息进行训练.而 APP 的计算过程来源于网页排名值,在网页排名值中会设定一个停留率(stoprate),通过制造随机数与停留率进行比较,来决定当前页面是否停留,收集起始点与终点从而生成点对信息.

得到点对信息后,进入模型训练的流程.在训练时,采用对损失函数进行求导.通用的损失函数结合负采样算法,通过随机梯度下降法对模型进行更新,其模型主要包括两个部分:第一部分为图节点向量,第二部分为负采样节点向量.

我们发现:在上述这 3 个算法过程中,均能够生成点对的信息,接下来可以统一利用点对的信息进行计算.因此,可以将基于词向量与负采样模型的图嵌入算法归纳出先采样后训练的模式,从而将采样逻辑协同训练逻辑相互独立,对训练的逻辑进行单独的优化,形成单独的模块.一般新算法的提出往往是在采样的模块进行更改,优化的模块对于用户是透明的,用户只需要修改采样的代码即可完成新算法的修改.

2.2 高可复用的分布式图嵌入框架

基于上文中的分析,笔者构建了一种基于采样的框架,将各种图嵌入的算法进行统一,提出一个高可复用的分布式图嵌入框架.

如图 2 所示,算法的流程是:首先输入为原始图,以边为单位,通过处理原始图,利用不同算法进行采样,将这

些采样信息处理为 Skip-gram 能够接受的格式,即点对信息.在 Skip-gram 与负采样训练过程中,其输入是处理完毕的点对信息,将这些点对分成批次进行训练,每一次训练均通过随机梯度下降算法收敛得到最后模型.结合图 3 介绍高可用的分布式图嵌入框架.

Fig.2 The training process of single-machine

图 2 单机训练流程

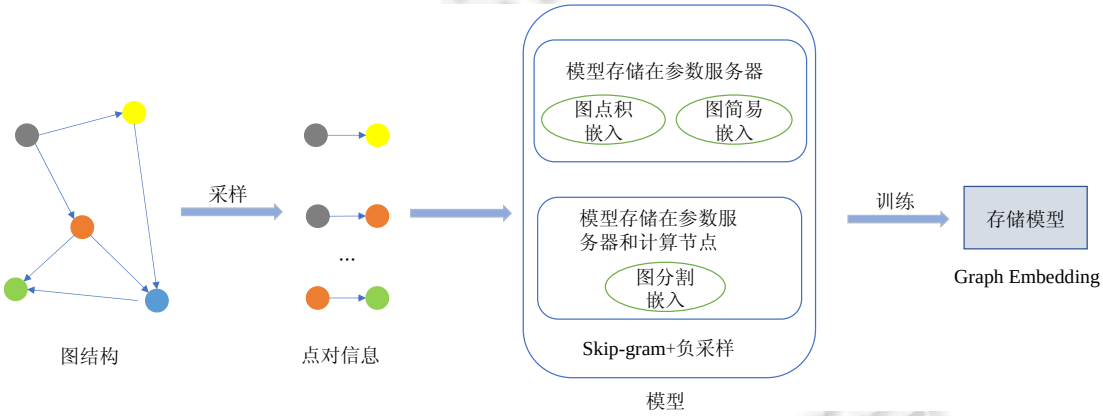


Fig.3 The distributed training framework

图 3 分布式训练框架

分布式的训练流程主要分为两个部分:第一部分为采样流程,第二部分为训练部分.采样流程和训练流程均与单机不同,接下来介绍这两个部分.

- 采样流程:数据输入为一张图,在图上会运行一些采样算法,通过分布式的机制进行采样.实际原理是将其转化为 Spark[20]上的 RDD 算子运算,通过将数据分发给不同的运算节点来处理得到结果,减少采样时间;
- 训练流程:在分布式训练模块中,输入为点对信息,不同策略对点对信息进行分组,放置在不同计算节点上.计算节点拿取到点对信息后,会分批次进行分布式训练.通过在参数服务器上放置模型位置的不同,可将分布式策略分为两种:一种在参数服务器上存储所有模型,另一种在参数服务器与计算节点同时存储模型.在参数服务器上存储所有模型有两种:图简易嵌入分布式策略和图点积嵌入分布式策略,这两种策略是目前业界常见的实现策略.

以深度游走为例,其采用深度优先遍历的方式,从某个节点出发,通过多次深度优先游走记录下节点遍历的路径,从而将路径转化为一条序列.这种做法称之为采样.采样得到的序列再经过 SKIP-GRAM 的格式变化之后,可以处理为点对信息.在输入采样算法具有相似的目标函数的作用下,分布式训练可以独立于采样算法本身,性能优化交给分布式训练的模块,训练模块只需顾忌自身的训练性能以及收敛精度即可.完成最终的模型训练,与原先的算法得到同样好的收敛结果.

最终的框架具有良好的可扩展性,解除了算法采样与训练之间的耦合关系,使得该框架成为一种高度内聚

的实现.当有新的算法加入时,只需要将其采样的部分改写为框架所支持的代码即可.大多数的算法往往会在采样上做出新的调整,而训练的损失函数变动不大.当目标函数与框架不同时,只需要对分布式的训练模块进行简单的处理,通过继承或者接口注入的方式改写目标函数相对应的类即可,用户无需对新的算法进行大量的调参,也无需对其进行分布式上网络通信的优化(如图 4 所示).

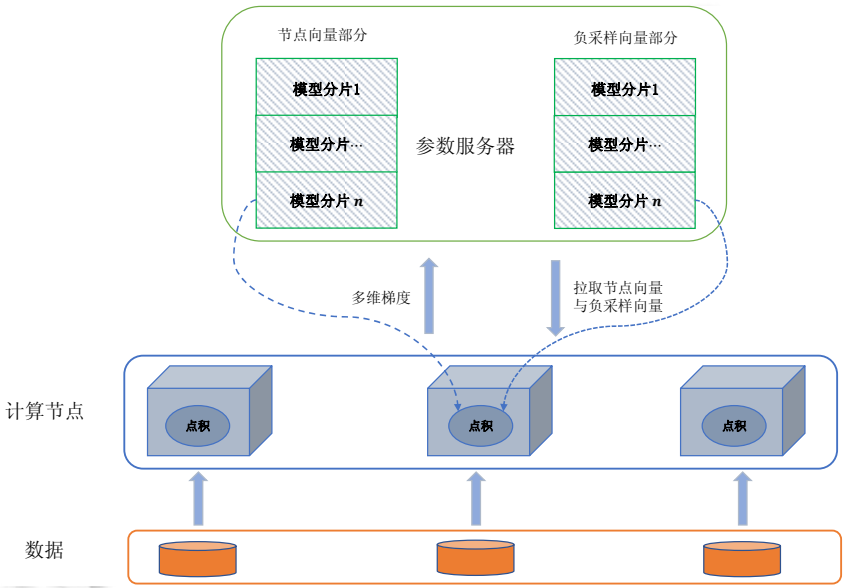


Fig.4 The distributed execution process of the simple graph embedding strategy

图 4 图简易嵌入分布式策略的执行流程

3 基于高性能的分布式解决方案

3.1 已有的分布式策略分析

目前,工业界已有两种很常见的分布式实现策略,它们都是在参数服务器上存储所有模型.根据在计算节点和参数服务器上点积的不同运算方式,可以分为图简易嵌入分布式策略与图点积嵌入分布式策略:图点积嵌入分布式策略缺乏良好的可扩展性;而图简易嵌入分布式策略具有性能瓶,需要进一步的优化方案.本节简要介绍这两种常用的分布式策略.

3.1.1 图简易嵌入分布式策略

- 数据与模型存储

图简易嵌入分布式策略将点对数据独立划分在分布式文件系统[21]上,利用参数服务器进行节点向量与负采样向量模型的存储.每个点的嵌入信息完全存放在一台机器上,不存在单点嵌入的切割,这里与图点积嵌入分布式策略加以区分.图简易嵌入分布式策略与图点积嵌入分布式策略的执行流程分别如图 4 和图 5 所示.

- 训练流程

步骤 1. 图简易嵌入分布式策略进行本地的数据读取,得到需要参与计算的模型索引,模型索引即为每个图顶点的索引值.预先对模型索引进行挑选,选取需要用到嵌入信息的顶点.在第 2 步中,拉取只需拉取需要的模型向量;

步骤 2. 根据这些模型索引,从参数服务器上拉取需要的模型.只有部分模型需要参与一个批次的计算,参与计算的模型索引在步骤 1 中计算完毕,需要拉取的模型参数为节点向量模型与负采样向量模型的部分模型,通信开销为 $O(2mD)$,m 为每一个批次下参与计算的独立节点个数(去除重复节

点之后的结果);

步骤 3. 图简易嵌入分布式策略利用 Skip-gram 中的损失函数进行求导,进行梯度的计算,这里需要涉及到节点向量和负采样向量两个模型,采用的算法统一为负采样.

步骤 4. 计算完毕之后,将梯度传递回参数服务器.因为节点向量和负采样向量均需要传递 D 维度的参数,因此这里的通信开销是 $O(2mD)$.模型在参数服务器的更新采用的是累加的模式.

总通信开销为拉取与推送这两个操作带来的模型传输和梯度传输,一个批次下需要的通信量为 $O(4mD)$.

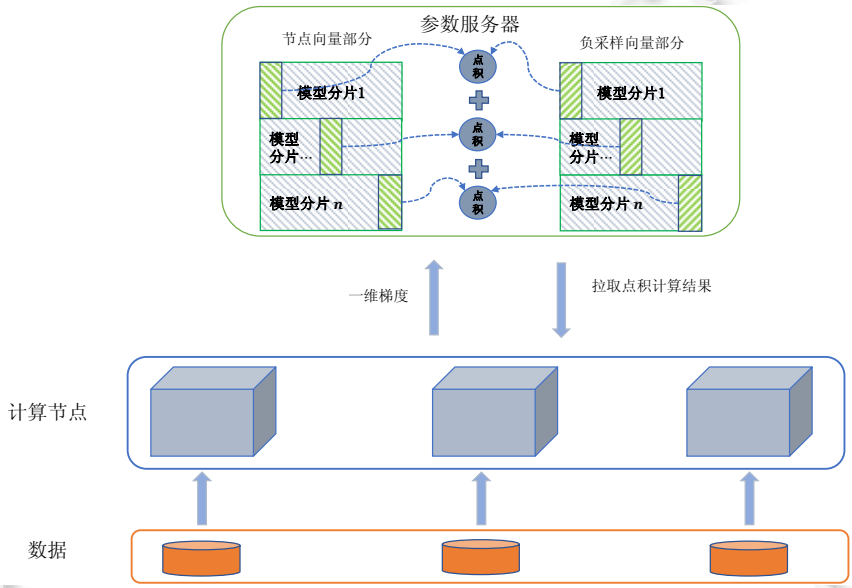


Fig.5 The distributed execution process of the dotted graph embedding strategy

图 5 图点积嵌入分布式策略执行流程

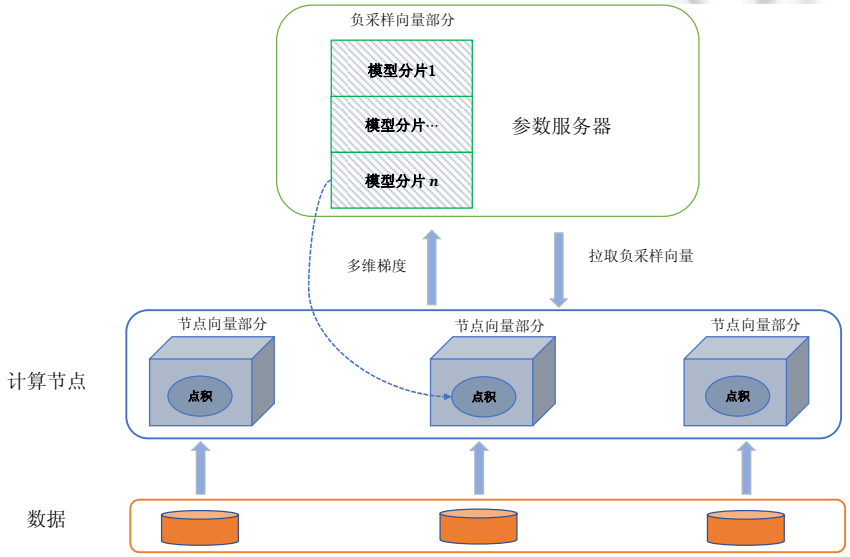


Fig.6 The distributed execution process of the partitioned graph embedding strategy

图 6 图分割嵌入分布式策略分布式训练执行流程

3.1.2 图点积嵌入分布式策略

- 数据与模型存储

数据划分同图简易策略,模型分为节点向量与负采样向量.一个节点向量的不同列存放在不同的机器上,与图简易嵌入分布式策略不同.

- 训练流程

步骤 1. 计算节点进行本地的数据读取,得到需要参与计算的模型索引;

步骤 2. 计算节点根据获取到的模型索引,从参数服务器上拉取需要的模型.需要拉取的结果为点积计算之后的结果,点积计算在参数服务器上完成,无需进行通信,只需将点积计算结果传递给计算节点.传输的维度也从 D 维变为了 1 维度.因此,通信开销为 $O(2m)$;

步骤 3. 计算节点进行梯度计算,计算一维梯度,将一维梯度推送回参数服务器,参数服务器可利用一维梯度更新每个维度;

步骤 4. 计算完毕后,将一维梯度传递回参数服务器,节点向量和负采样向量均需要传递 1 维度参数,参数服务器将会利用一维梯度去更新其他维度梯度,更新结果同原来一致,故通信开销是 $O(2m)$.

总体通信开销为拉取与推送两个操作带来模型传输和梯度传输之和,一个批次下,通信量为 $O(4m)$.

3.2 图分割嵌入分布式策略

- 数据存储与模型存储

节点向量模型按照节点的 id 分布在计算节点的每台机器上,在参数服务器上存储全部的负采样向量模型,同一节点向量的不同维度存放在相同的机器上,它的训练执行流程如图 6 所示.

- 训练流程

步骤 1. 每台机器进行本地的数据读取,得到需要参与计算的模型行索引,数据读取按照图上节点 id 进行哈希划分,分散在每台机器上;

步骤 2. 需要拉取的模型为负采样向量的部分模型,由于图节点向量在每个计算节点本地,通信开销为 $O(mD)$;

步骤 3. 每台机器需要进行梯度的计算.节点向量的模型放置在计算节点上,数据为点对信息,根据节点哈希到不同机器上,节点向量模型直接在本地更新;

步骤 4. 计算完毕之后,将负采样向量模型的梯度传递回参数服务器,并且在本地更新相应节点向量模型.因为每个负采样向量模型节点需要传递 D 维度的参数,通信开销为 $O(mD)$.

总体的通信开销为拉取与推送这两个操作带来的模型传输和梯度传输之和,一个批次下需要的通信量为 $O(2mD)$,对比图简易嵌入分布式策略,减少了一半通信量.

4 实验与分析

4.1 图分割嵌入分布式策略

- 数据集

我们采用 3 个公开的数据集(下载地址:https://snap.stanford.edu/data/),即维基百科(WIKI)、油管(YOUTUBE)、现场博客(LIVEJOURNAL)数据集.数据集的基本信息概括在表 1 中.

Table 1 Dataset size

表 1 数据集规模

数据集	#节点个数	#边个数	有向图/无向图
维基百科	7 115	103 689	有向图
油管	1 134 890	2 986 7624	无向图
现场博客	4 847 571	68 993 773	有向图

- 实验环境

目前的实验在一个实验室集群下进行,该集群包含了 8 台物理机,每个机器包含 2 个 CPU,32GB 内存,机器之间的网络为 1Gb/s.笔者在开源系统 Spark 上统一实现了 3 种图嵌入分布式策略.

- 实验设计

我们从以下 3 个角度设计实验:(1) 选用 3 种常见的图嵌入算法进行实验,以证明算法的通用性;(2) 采用常见的图嵌入算法对 3 种不同分布式嵌入策略在不同数据集上进行对比,以展示图分割嵌入式策略的高效性;(3) 对 3 种图嵌入分布式收敛的情况分析,以证明各种实现策略均不影响算法的正常收敛.在所有实验中,采用的图节点嵌入向量维度均为 128 维,均使用负采样算法进行训练.

4.2 各分布式策略的性能与可扩展性分析

图点积嵌入分布式策略在较少计算节点的数目下具有性能优势,在 2 个或者 4 个计算节点时,其训练的时间开销往往低于图简易嵌入分布式策略和图分割嵌入分布式策略.随着计算节点数目的增加,图点积嵌入分布式策略的性能开始下降.下降原因来自计算节点数目增加后,点积运算带来的提升有限,并且点积运算所需要的加法次数变多.此外,机器数目增多引入额外协调开销,一台机器需要等待另外的机器点积运算完毕才能拿到结果,因此拿到结果最终取决于最慢的机器.因此,其核心运算优势减少,通信开销增加,最终导致性能大幅下降(如图 7~图 15 所示).

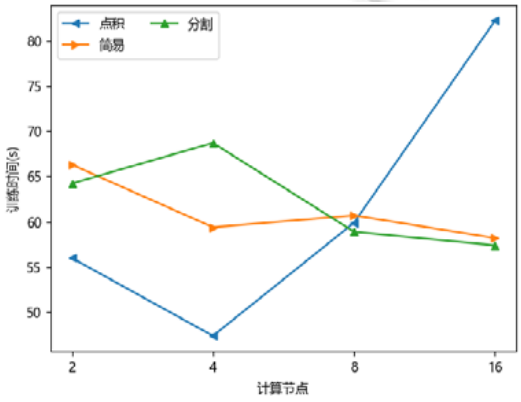


Fig.7 WIKI-LINE

图 7 维基百科-大规模信息网络嵌入

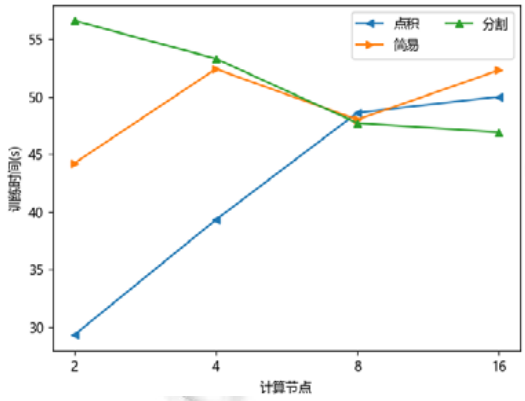


Fig.8 WIKI-APP

图 8 维基百科-非对称临近可扩展性图嵌入

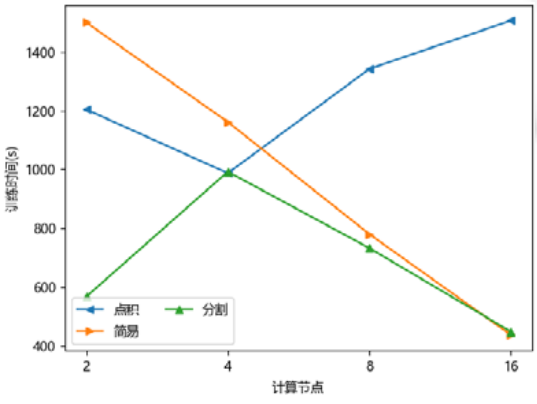


Fig.9 WIKI-DeepWalk

图 9 维基百科-深度游走

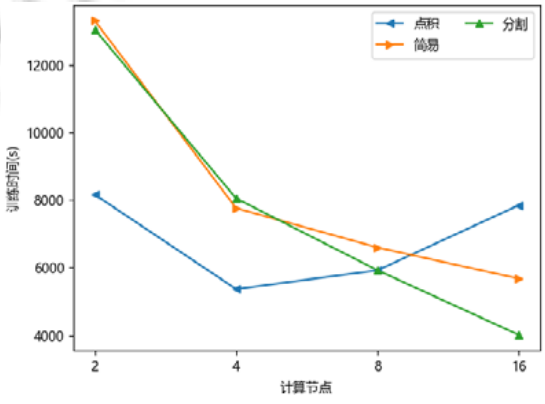


Fig.10 YOUTUBE-LINE

图 10 油管-大规模信息网络嵌入

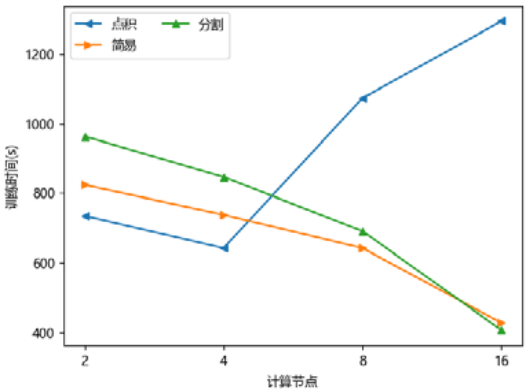


Fig.11 YOUTUBE-APP

图 11 油管-非对称临近可扩展性图嵌入

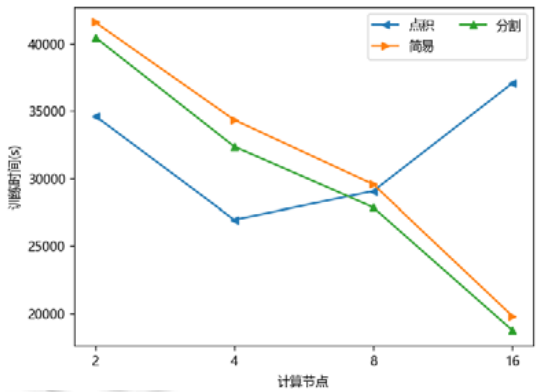


Fig.12 YOUTUBE-DeepWalk

图 12 油管-深度游走

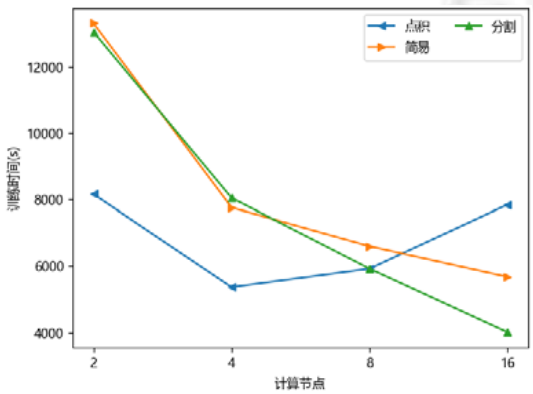


Fig.13 LIVEJOURNAL-LINE

图 13 现场博客-大规模信息网络嵌入

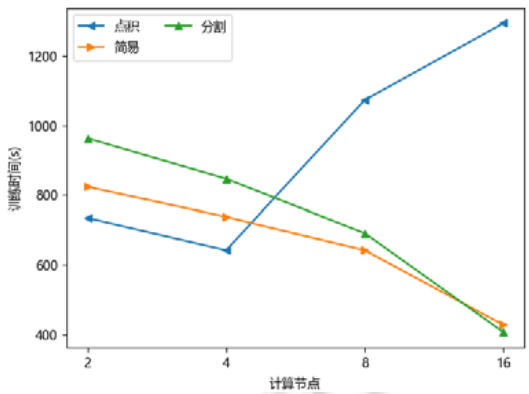


Fig.14 LIVEJOURNAL-APP

图 14 现场博客-非对称临近可扩展性图嵌入

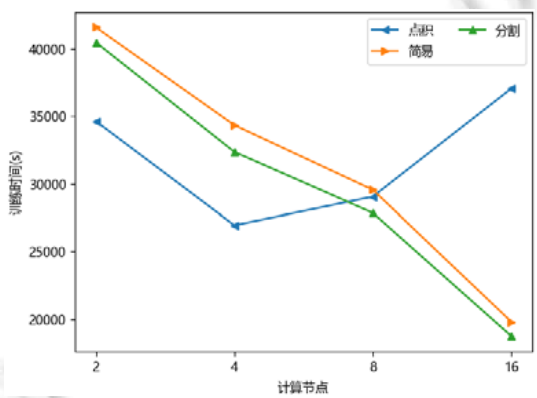


Fig.15 LIVEJOURNAL-DeepWalk

图 15 现场博客-深度游走

对于图简易嵌入分布式策略而言,当计算节点和模型切片数目增加,其点积操作并不与模型切片数目相关,其具有较好的加速比;另一方面,每一台机器的运算均可以独立进行,无需和其他机器同步,因此机器的协调时间大大减少.虽然机器数目增加会带来一定的额外时间开销,但增大量远不如图点积嵌入分布式策略,其可扩展

性较好,性能总体而言不如图分割嵌入分布式策略.

对于图分割嵌入分布式策略,其具有不错的性能优势.当计算节点和模型切片数目增加时,其计算模式同图简易嵌入分布式策略近似;但网络开销比其要少,其节点向量矩阵的更新与拉取在每台机器本地完成,理论上将节省一半的网络通信.同样,与图点积嵌入分布式策略不同,其额外开销与机器数目影响不大,内积运算单独在每台机器上完成,并未累加.因此,其性能具有良好的可扩展性,并且其往往在多个计算节点的情况下,性能比图点积嵌入分布式策略要好.由于其拉取操作的通信量比图简易嵌入分布式策略要少,其实验表现比图简易嵌入分布式策略要好.而对于少许性能震荡的情况而言,可能出于数据划分的倾斜问题,导致每台机器分配的算力不均衡,最终导致个别结果不如图简易嵌入分布式策略与图点积嵌入分布式策略.总体而言,对比图简易嵌入分布式策略与图点积嵌入分布式策略的分布式训练策略,图分割嵌入分布式策略在三者数据集上体现出了极好的性能优势,并且表现出了优异的可扩展性.

4.3 各分布式策略收敛结果的对比

本文对计算节点、图点积嵌入分布式策略、图分割嵌入分布式策略这 3 种采样策略均进行了 LOSS 收敛曲线的实验.实验中固定嵌入的维度为 128,并且采用 20 轮次的迭代.出于不同分布式策略的学习率与模型适应程度不同,并且不同分布式策略具有不同的计算模式,梯度更新顺序与模式不完全一致.因此,为了让每个算法取得更好的收敛结果,本文对于不同的算法采用了不同的学习率初始值.

本文以现场博客数据集为例,分别绘制了非对称临近可扩展性图嵌入、深度游走、大规模信息网络嵌入这 3 种采样策略对应的 2,4,8,16 个计算节点和模型切片数目的 LOSS 收敛曲线图.接下来将通过图表对结果做进一步的分析.

APP 基于根网页排名,根据停留率(stoprate)对图节点序列进行采样,其中,图节点序列采用随机游走的方式得到,包括深度优先遍历或宽度优先遍历.因此,在输入序列数据规模相同的情况下,其采样点对会远远少于 DeepWalk 的点对.由于图嵌入为无监督学习,其 LOSS 损失很大程度上取决于输入数据规模:当数据规模扩大后,其总体数据复杂程度会有更大机率扩大;当输入数据规模减少时,其数据复杂程度会减少.数据复杂程度影响 LOSS 损失函数:当 LOSS 损失很大时,表明模型对数据的认知较弱;当 LOSS 损失较小时,表明此时模型对数据的认知较强.当数据复杂程度较小时,模型会很快收敛.因此,由图中可以观察得出:对于非对称临近可扩展性图嵌入采样算法,图点积嵌入分布式策略、图简易嵌入分布式策略、图分割嵌入分布式策略三者均会稳定收敛.以图 16(a)为例:当采用两个计算节点时,3 种分布式策略均在第一轮迭代后很快收敛,并在后续的迭代中缓慢收敛.图 16(b)~图 16(d)分别表现了采用更多计算节点时,LOSS 的收敛情况.从 4 张图中可以发现:扩大计算节点个数后,图点积嵌入分布式策略、图简易嵌入分布式策略、图分割嵌入分布式策略三者均会稳定收敛.

深度游走采样算法是最初提出用以建模图节点向量,流程为从某一节点出发,根据深度优先遍历结果或宽度优先遍历统计节点邻域序列,再根据节点邻域序列学习得到模型.深度游走采样对模型的刻画较为粗糙,数据复杂程度较非对称临近可扩展性图嵌入更为复杂.数据规模比非对称临近可扩展性图嵌入大,模型学习难度较大.由图 17(a)可以看出:当采用 2 个计算节点时,模型在第一轮迭代损失会下降,但下降程度并不明显,随后的几轮迭代模型进入缓慢收敛期,出于模型的复杂程度较大,3 类算法对其学习程度略微陡峭,收敛速度整体不如非对称临近可扩展性图嵌入.其中,图点积嵌入分布式策略在第一轮次的收敛值相比其他两者稍显优势,在后来几轮迭代中进入缓慢收敛期,这一收敛结果与学习率的设置也有一定关系.总体三者最终收敛结果相同,趋于一致,表明 3 类算法均可学习得到不错的模型结果,取得优异的收敛值.

通过观察发现:在采用大规模信息网络嵌入采样算法时(如图 18 所示),图简易嵌入分布式策略与图分割嵌入分布式策略的 LOSS 曲线较为一致.这一现象是由于图简易嵌入分布式策略与图分割嵌入分布式策略具有较为近似的计算模式,其梯度更新顺序较为接近.两者不同之处仅仅在于模型存储位置以及其存储带来的梯度传输差异,例如:在图分割嵌入分布式策略中,其梯度将有一部分在本地更新;而图简易嵌入分布式策略则全于参数服务器中更新.两者还具有相应的数据划分引入的差异性:在图简易嵌入分布式策略中,其数据划分为随机进行;而图分割嵌入分布式策略则是预先进行划分,不含有梯度写回的误差.最终,3 种分布式策略在大规模信息网

络嵌入采样算法中达到同样的收敛结果.

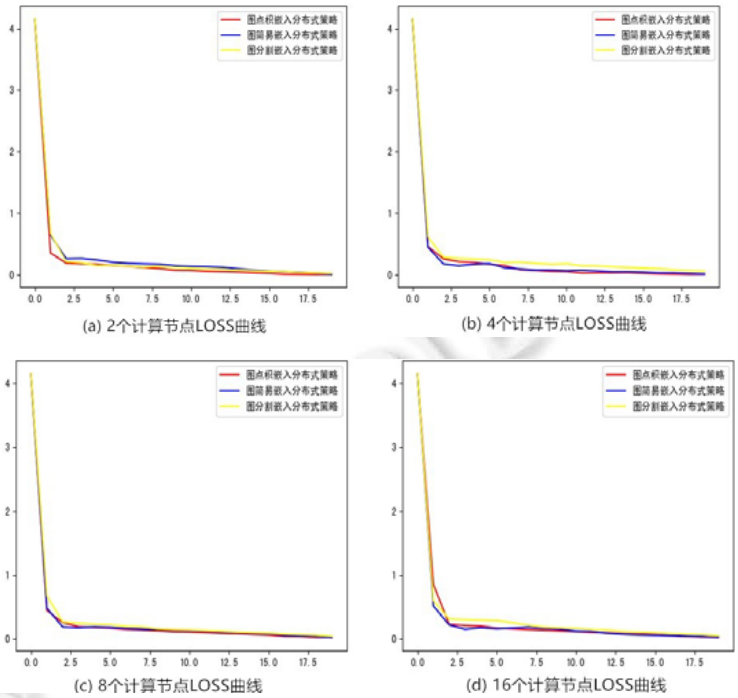


Fig.16 The LOSS curve of APP

图 16 非对称临近可扩展性图嵌入算法 LOSS 曲线

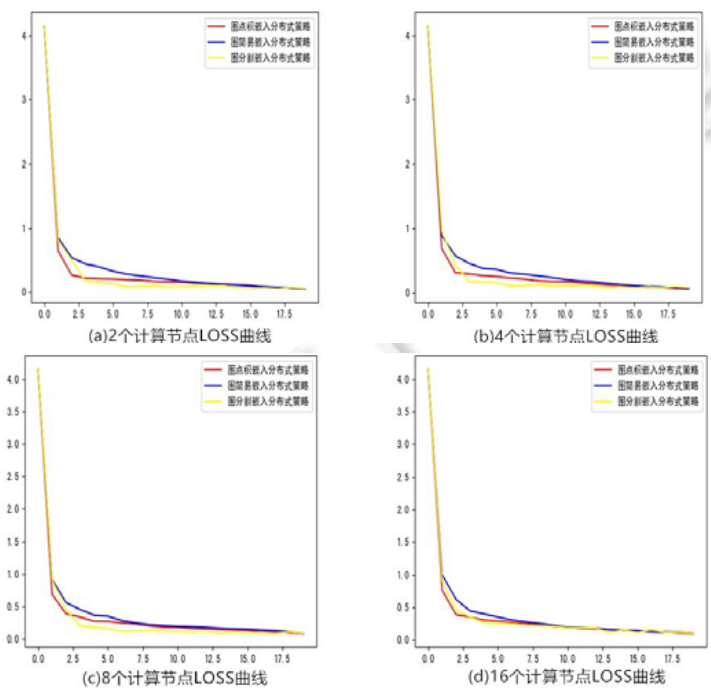


Fig.17 The LOSS curve of DeepWalk

图 17 深度游走 LOSS 曲线

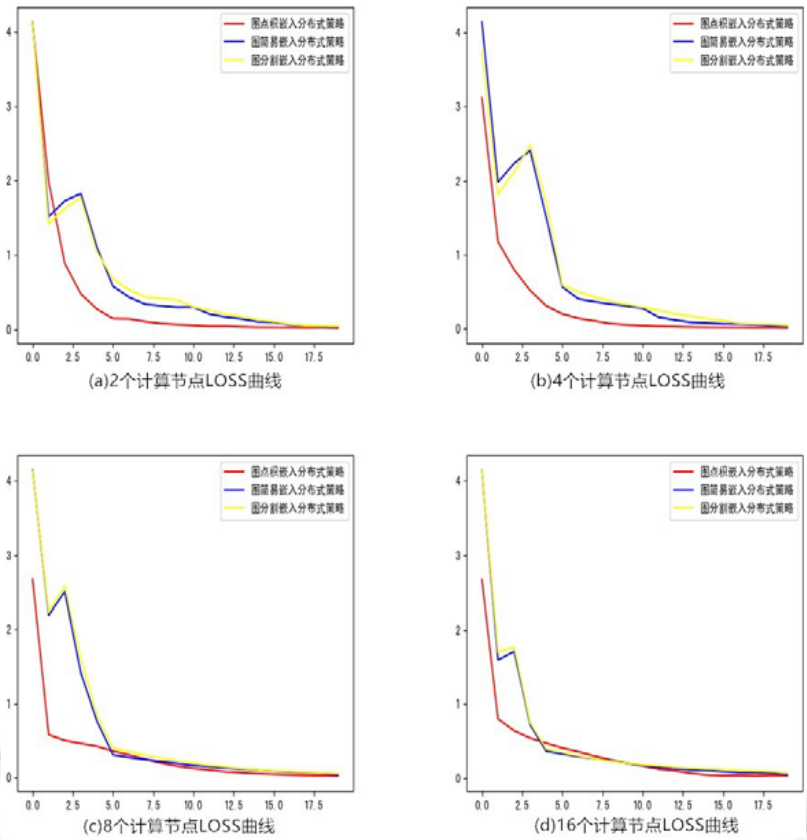


Fig.18 The LOSS curve of LINE

图 18 大规模信息网络嵌入 LOSS 曲线

结合上述图表与分析,3 种分布式策略的收敛结果差异不大,有些算法在收敛中间略有震荡.这一结果可能与不同机器的数据分布不同,以及分布式策略对数据的依赖性、数据打乱的结果以及初始学习率的设置有关.最终,在 3 种图嵌入采样算法中,3 类分布式训练策略得到的损失(LOSS)收敛结果均为一致,表明三者策略对于数据的学习程度较为接近.

4.4 各分布式策略收敛速度的对比

通过多次实验发现:机器学习模型的收敛速度与其初始的学习率具有较大的关联,并且与选择的优化器有直接关系.其中,优化器可以分为有动量的与随机梯度下降法,出于对分布式多机器收敛影响的考虑,在此选择为最基础的随机梯度下降法.若初始的学习率较大,那么模型将会很快地进入较优值,并且 LOSS 收敛曲线会随着迭代轮次数目的增加时而震荡.如果初始的学习率较小,模型收敛到达最优解的可能性会增加,通常情况下,LOSS 收敛曲线会在最后迭代次数下轻微震荡.不同的分布式训练策略往往有着不同的数据依赖,模型向量之间的依赖,每一种训练策略,其并行运算的流程与常见的串行算法并不一致,而基于 HOGWILD[22]理论,每个机器之间可以无序的写回梯度,使得最终结果收敛.

5 总 结

本文首先通过对多种现有的图嵌入方法进行调研,提出了一个高可用的分布式训练框架.该框架通过将采样、训练过程解耦,从而能够表达多种图嵌入算法.用户可以只关心采样的流程或只关心训练的流程,从而使得其中的算法具有高度的可复用性.另外,笔者针对业界现有的分布式图嵌入策略进行分析,发现了这些算法在训

练性能以及扩展性上的不足,进而提出了一种图分割嵌入分布式策略.笔者实现了一个原型系统,并通过充分的实验分析了 3 种分布式策略.笔者发现:机器数目增加后,图点积嵌入分布式策略的可扩展性变差,在性能对比上处于劣势地位;而图简易嵌入分布式策略和图分割嵌入分布式策略具有较为不错的可扩展性,本文提出的图分割嵌入分布式策略具有更好的性能优势.

References:

[1] Zhang W, Miao X, Shao Y, Jiang J, Chen L, Ruas O, Cui B. Reliable data distillation on graph convolutional network. In: Proc. of the 2020 ACM SIGMOD Int'l Conf. on Management of Data. 2020. 1399−1414.

[2] Wu S, Zhang Y, Gao C, Bian K, Cui B. GARG: Anonymous recommendation of point-of-interest in mobile networks by graph convolution network. Data Science and Engineering, 2020,5(4):433−447.

[3] He J, Liu HY, Zheng YQ, Shu T, He W, Du XY. Bi-labeled LDA: Inferring interest tags for non-famous users in social network. Data Science and Engineering, 2020,5(1):27−47.

[4] Shao Y, Chen L, Cui B. Efficient cohesive subgraphs detection in parallel. In: Proc. of the 2014 ACM SIGMOD Int'l Conf. on Management of Data. 2014. 613−624.

[5] Sikos LF, Philp D. Provenance-aware knowledge representation: A survey of data models and contextualized knowledge graphs. Data Science and Engineering, 2020,5(3):293−316.

[6] Perozzi B, Al-Rfou R, Skiena S. Deepwalk: Online learning of social representations. In: Proc. of the 20th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. 2014. 701−710.

[7] Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q. Line: Large-scale information network embedding. In: Proc. of the 24th Int'l Conf. on World Wide Web. 2015. 1067−1077.

[8] Zhou C, Liu Y, Liu X, Liu Z, Gao J. Scalable graph embedding for asymmetric proximity. In: Proc. of the 31st AAAI Conf. on Artificial Intelligence. 2017. 2942−2948.

[9] Grover A, Leskovec J. node2vec: Scalable feature learning for networks. In: Proc. of the 22nd ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. 2016. 855−864.

[10] Mikolov T, Sutskever I, Chen K, *et al*. Distributed representations of words and phrases and their compositionality. In: Proc. of the Advances in Neural Information Processing Systems. 2013. 3111−3119.

[11] Goldberg Y, Levy O. word2vec explained: Deriving Mikolov *et al*.'s negative-sampling word-embedding method. arXiv preprint arXiv:1402.3722, 2014.

[12] Jiang J, Yu L, Jiang J, Liu Y, Cui B. Angel: A new large-scale machine learning system. National Science Review, 2018,5(2): 216−236.

[13] Jiang J, Xiao P, Yu L, Li X, Cheng J, Miao X, Cui B. PSGraph: How tencent trains extremely large-scale graphs with spark? In: Proc. of the 2020 IEEE 36th Int'l Conf. on Data Engineering (ICDE). IEEE, 2020. 1549−1557.

[14] Yang H. Aligraph: A comprehensive graph neural network platform. In: Proc. of the 25th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining. 2019. 3165−3166.

[15] Guthrie D, Allison B, Liu W, Guthrie L, Wilks Y. A closer look at skip-gram modelling. In: Proc. of LREC. 2006. 1222−1225.

[16] Smola A, Narayanamurthy S. An architecture for parallel topic models. Proc. of the VLDB Endowment, 2010,3(1-2):703−710.

[17] Dean J, Corrado G, Monga R, Chen K, Devin M, Mao M, Le QV. Large scale distributed deep networks. In: Proc. of the Advances in Neural Information Processing Systems. 2012. 1223−1231.

[18] Xing EP, Ho Q, Dai W, Kim JK, Wei J, Lee S, Yu Y. Petuum: A new platform for distributed machine learning on big data. IEEE Trans. on Big Data, 2015,1(2):49−67.

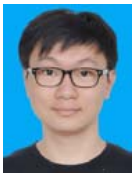
[19] Li M, Andersen DG, Park JW, Smola AJ, Ahmed A, Josifovski V, Su BY. Scaling distributed machine learning with the parameter server. In: Proc. of the 11th USENIX Symp. on Operating Systems Design and Implementation (OSDI 2014). 2014. 583−598.

[20] Zaharia M, Chowdhury M, Franklin MJ, Shenker S, Stoica I. Spark: Cluster computing with working sets. HotCloud, 2010, 10(10-10):95.

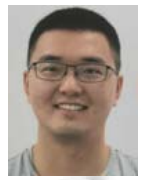
[21] Shvachko K, Kuang H, Radia S, Chansler R. The hadoop distributed file system. In: Proc. of the 26th IEEE Symp. on Mass Storage Systems and Technologies (MSST). IEEE, 2010. 1−10.

[22] Recht B, Re C, Wright S, Niu F. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In: Proc. of the Advances in Neural Information Processing Systems. 2011. 693−701.

张文涛(1994－),男,博士,主要研究领域为图计算,自动化机器学习,分布式系统.



苑斌(1993－),男,硕士,主要研究领域为数据库,大数据管理分析,机器学习.



张智鹏(1993－),男,博士,主要研究领域为大数据处理,分布式机器学习.

崔斌(1975－),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为数据库,大数据管理分析.