

图数据流的模型、算法和系统

李友焕, 邹磊

北京大学计算机科学技术研究所, 北京 100080

摘要

在应用数据高速增长的场景下,已有的静态图计算的模型和方法难以应对数据高速更新的挑战,图数据流模型应运而生。首先讨论当前大规模复杂数据流的产生及其管理需求,分析静态图模型以及已有数据流算法、系统在应对这一数据流场景的固有缺陷,阐述图数据流模型产生的重要背景。然后通过总结分析早期图的流式计算以及已有的少量图数据流的研究工作,给出图数据流模型的一般定义。最后,从方法和问题两个角度探讨图数据流的研究前景,并简要介绍图数据流管理系统相关技术架构。

关键词

图模型;数据流系统;图数据流;数据管理系统

中图分类号:TP315

文献标识码:A

doi: 10.11959/j.issn.2096-0271.2018039

Graph stream: model, algorithm and system

LI Youhuan, ZOU Lei

Institute of Computer Science & Technology of Peking University, Beijing 100080, China

Abstract

In the scenario where data of real-world applications is in high-speed growth, existing methods for static graph computation are hard to approach the challenges from the rapidly updated data, and graph stream model arise at the historic moment. The inherent defects of static graph model and exiting data stream algorithms/systems over high-speed graph-structured data were discussed, and then the formal definition of graph stream combining with previous similar models was presented. Some promising research problems and applications over graph stream were probed, and then the possible requirements and technique issues on graph stream management system (GSMS) were looked into the future.

Key words

graph model, data stream system, graph stream, data management system

1 大规模流数据的机遇与挑战

图数据流是最近几年才受到广泛关注的前沿科研领域，其兴起主要是源于新时代下移动应用实时产生的大规模复杂数据。

过去十几年，随着智能手机的普及以及移动互联网的发展，移动应用层出不穷。这些应用涉及即时通信、社交网络以及网络购物等各个方面，并实时地产生大量的数据。这些数据本质上是现实世界人、事、物及其交互的一种深入量化。对这些数据的及时分析与挖掘能够产生高价值的信息，进而改进人们生活的多个方面。例如，微信、微博等社交网络上有庞大的活跃用户，这些用户对社交网络而言更像是分布在各地的“传感器”，将各自的活动区域内的热点见闻“报告”在社交网络上。如在地震等自然灾害发生时，人们可以通过社交网络实时传递和获取相关的灾情^[2]。因此，这些应用数据具有极大的分析研究价值。

尽管移动应用数据蕴含着高价值的信息，但这些数据却具有结构复杂、规模庞大、高速增长等特点。人们对不同应用有不同的需求，这决定了移动应用数据是复杂多样的，而针对同一应用产生的数据，不同的数据分析方也会有不同的数据需求。例如，针对社交网络的数据，研究社交心理的人更关注用户以及用户间的好友关系与交互行为，而广告媒体的从业人员则更关心平台上发文内容中的产品或话题信息。人们对数据的多样化的需求决定了移动应用数据的复杂性。数量众多的软件及其庞大的用户量决定了相关数据的海量规模。例如，微信的月活跃用户数已超过了10亿，而用户之间的交互则会带来更大规

模的数据，包括语音、视频、图片以及相关的文本等。这些大规模的复杂数据还在实时地高速增长，如社交网络每天以亿级别的发文、轨道交通应用形成的大规模定位与轨迹信息以及网络通信中的数据传播等。

传统的关系型数据管理模型虽然已有众多标准规范和技术积淀，但仍难以管理复杂多变的数据。一方面，数据的关系框架的设计成本较高，既定的数据框架结构很难适应数据种类、格式的频繁变化；另一方面，关系型数据库中，基于关联信息的计算代价很高，如表格的联结操作等，这使得在大规模数据场景下关系型数据库管理模型难以满足数据分析处理的需求。

图模型的点、边元素非常适用于建模复杂数据中的对象以及对象间的关联和交互，点和边上的属性、标签以及相关数据等的自由定义使得图模型能够很容易地以统一的形式表达不同的对象及其间的交互行为。例如，在社交网络上，基于用户好友关系建模的图和以文本关键字共现关联建模的图可以很容易通过增加用户与文本的发表关系快速融合成一个图。因此，图模型非常适合用来建模大规模复杂数据。然而，图模型上的计算却很难应对图数据高速更新的场景。图数据上的计算往往通过构建复杂的索引来加速查询。在静态图数据上，因为索引只需要离线构建一次，所以高构建代价对整体性能的影响有限。而在图数据高速更新的场景下，索引也需要频繁更新，越是复杂的索引往往更新越困难，甚至需要完全重新构建。尽管索引能够加速查询，但在流场景下的频繁索引更新也会严重影响整体性能。

数据流模型及其相关研究虽然都有针对数据更新的设计，但已有的数据流模型中缺少对图结构数据的支持。数据流中的

元素往往具有统一简单的格式,并且元素之间相对独立,缺少对对象关联的建模。因此,数据流模型的相关算法也很难扩展到需要图模型建模的复杂数据上。

在大规模复杂数据流的场景下,已有的图与数据流相关的模型和算法均有明显缺陷。尽管大规模实时更新的复杂数据给人们带来了获取高价值信息的重大机遇,但也带来了数据管理和计算上的巨大挑战。人们急需一种既能够为复杂数据建模,又能够应对更新挑战的新的数据模型、技术来满足相应的信息管理需求。

2 流计算系统

在图数据流模型受到广泛关注之前,有几种流式计算工具兴起,并在一定程度上弥补了传统技术应对大规模复杂数据流的不足,其中包括Storm^[3]、Spark Streaming、Samza、Flink以及Kafka Stream。本章将分别简要介绍和分析这几种系统,而后总结分析这些系统对大规模复杂数据流处理的相对优势及其仍然存在的重要不足。

(1) Storm

Storm是Twitter开源的一个流系统。在Storm初始化时,需要用户定义一个实时计算的框架,这个框架的结构其实是一个有向图,图中点是集群中的计算节点,而点与点之间的边则对应整体计算逻辑中数据的传输。这个图框架也被称为拓扑。在一个拓扑中,传输的数据单元是一系列不可修改的键值对(tuple),键值对从消息源(spout)点中输出,形成流数据,并传输到消息处理者(bolt)点中进行计算,进而产出新的输出流,bolt输出流也可以传输给其他bolt节点,形成流水线式的计算处

理流。

(2) Spark Streaming

流式Spark其实是Spark核心应用程序编程接口(application programming interface, API)的一个扩展,其对流式处理的支持其实是将流数据分割成离散的多个小批量的RDD(RDD是Spark的数据单元)数据,然后再进行处理。这些小批量的数据被称为DStream(D为discretized,即离散化的意思)。DStream数据既可以被自定义的函数并发地处理,也可以在移动窗口的数据模型下进行计算。

(3) Samza

Samza系统处理的流数据单元实质是类型相同或相近的消息,这些消息在产生之后是不可修改的。新产生的消息将被追加到流中,而流中的消息也不断地被读取。每条消息可以有相应的键值,这些键值可以用来对流中的所有消息进行分割。Samza的工作过程是对一组输入流中的消息数据进行按需计算转换(或过滤),并将计算结果以消息数据的形式附加到输出流中。因此, Samza的运行工作负载可能包含多个工作组,工作组之间可能有流数据的依赖关系,进而将整体的Samza计算抽象成一个数据流图框架,其中点都是流式的消息,而消息之间的边则为完成消息逻辑转化的计算任务的工作组。

(4) Flink

Flink是跟Storm比较相似的系统,其整个数据处理过程称为Stream Dataflow,既定的数据流动框架类似于Storm的拓扑。此外,Flink提取数据流的操作(source operator)、数据转换(map, aggregate)的操作(transformation operator)以及数据流输出的操作(sink operator)跟Storm架构中spout与bolt之间、bolt和bolt之间的数据流是高度对应的。两者比较大的区别

在于容错机制。Flink的容错机制是检查点(checkpoint),通过异步实现,并不会打断数据流,因此Flink的检查点的开启与关闭对吞吐量的影响很小。而Storm采用的是ACK机制,开销更大,而且对吞吐量的影响更明显。另外,Flink提供的API相对Storm的API来说更高级,而Storm的API则更基础。Storm的优势主要是具有比较成熟的社区支撑和经过较长期迭代之后的稳定性,目前Flink成熟度较低,仍有部分功能需要完善,如在线的动态资源调整等。

(5) Kafka Stream

Kafka Stream是Apache Kafka中的一个轻量级流式处理类库,用来支撑Kafka中存储数据的流式计算与分析,利用这个类库进行计算的结果既可以写回Kafka,也可以作为数据源向外输出。目前的流式计算系统基本都支持以Kafka Stream的输出作为数据源,如Storm的Kafka Spout。作为一个Java类库,Kafka Stream并不是一个系统,但却是讨论流计算的工具中不得不提的对象。它可以非常方便地嵌入任意Java应用中,也可以以任意方式打包和部署,除了Kafka之外,并没有任何外部依赖。同流计算系统相比,使用Kafka受到的逻辑限制较少,开发者能够更好地控制和使用Kafka Stream上的应用。

这些流系统和流计算库的核心思想都是针对流式的输入利用集群进行协作计算,而整体的数据依赖所形成的框架则为一个有向无环图,因此集群的整体协作更像是流水线的协作,计算框架中的数据依赖所形成的数据流动方向基本上是单一既定的。除了数据处理的先后关系之外,这些流系统对不同工作组之间的数据的独立性要求往往更高,进而能实现较好的并行效果。然而对于很多图计算而言,计算的逻辑并不是流水线式的流系统能容易处

理的。图具有很强的表达能力,数据与数据之间的关联紧密,路径、连通分量等图计算使数据间的独立性比较低,单条边的变化可能对整个图的结构特征产生全局的影响。而且,图数据计算中的中间结果较多(如子图查询中的部分解),在分布式集群下的传输代价很高。因此,流式系统往往难以支撑大规模图数据流的计算。

3 图数据流模型

已有的相关模型、算法和流式计算系统均无法满足大规模复杂数据流的管理需求,图数据流应运而生。目前的相关工作缺乏针对图数据流的统一的形式化定义。后文首先介绍早期图的流式计算以及已有的少量图数据流的研究工作,然后再结合图数据流研究现状以及现实应用的背景,给出图数据流一般性的形式化定义。

3.1 图的流式计算

图的流式计算是指给定一个静态的图,在一遍或多遍顺序地读取对应的图数据的过程中,获得预期数据信息或完成既定操作的计算问题。图的流式计算的研究初衷在于图的规模远大于内存容量时,无法一次性将整个大图载入内存中计算,只能每次定量地载入一部分图数据到内存中,直到读完整个大图,在不断顺序读的过程中完成相关计算。在一次遍历图数据无法获取相应计算结果的情况下,可以多次进行完整的顺序读操作,也就是流式计算。随着存储的发展,计算空间开销带来的挑战明显降低,因此当前很少有图的流式计算的研究工作,已有研究工作主要出现在20世纪90年代末到21世纪初内存资源相对有限的时期。

流式计算涉及很多经典的图计算问题,包括多种图的属性特征计算和相关的图结构操作等。这些图的基本问题的计算方法往往是解决其他应用问题的重要基础,故这类问题的研究也比其他图数据流上的应用问题要早得多。这类问题的形式化模型往往很明确,同具体应用并没有直接的联系。其关注的重点主要是在空间有限无法载入整个大图时,如何多次遍历图数据完成相应图计算。图流式访问的研究问题包含众多图计算的经典问题,本文主要从其中的三角形计算、可达性和最短路径等问题做出简要介绍。

Bar-Yossef Z等人^[4]结合relative-近似和ratio-近似,提出了一个基于归约(reduction)的流式访问模型下的流算法,其中就以三角形计数流算法作为应用来分析,主要提到了两种流访问的形式:一种是以边为项的邻接边流(adjacency流),边的前后顺序任意;另一种是以点及其邻居的星状点流(incidence流),每个项是一个顶点及其邻居形成的星状结构。其后的聚焦在相同问题的两个工作^[5-6]基本沿着近似的思路以及adjacency流和incidence流的模型研究流式访问的三角形计数问题,并给出了更优的时间和空间上的相应算法的复杂度。

Unel G等人^[7]使用区间标记解决流式访问的图可达性问题。首先在必要时对图进行可达性等价的去环转化,转化过程是线性的。其次,根据图的去环转化的过程将顶点从前到后打上时间标签,然后流式访问时就按这个时间标签从前到后进行,也就是说这个图数据流模型是设定了流的顺序进行访问的,因为目标在于对大图的可达性的计算,所以这种假设除了离线的时间开销外并没有实际限制。这种情况下,去环的图可以转化为树,从而将图数据流上的可达性转化为树上的可达性问

题,而显然后者是有高效算法的,而且树的空间效率相比图也有很大的提高。

有向图流式访问上的最短路径问题最早是在Demetrescu C等人^[8]的一个工作中提出的,是图数据流上空间复杂度和计算所需的遍历次数的权衡问题下的一个图算法的应用。研究的结论是在给定比特的空间下,有向图的单源最短路径问题可以在 $O(n \cdot \log n)$ 次遍历过程中解决。

有学者将这类图的流式计算称为图数据流^[9],但考虑到数据仍是已知的静态图,只是读取的方式是流式的,因此本文将这类能够多遍顺序读取图的计算模型称为图的流式计算模型。显然,图的流式计算并不能满足当前大规模复杂数据流的计算需求。一方面,大规模复杂数据流很难进行多遍读取。图的流式计算中,全局的数据是给定的,而当前应用中实时产生的大量图结构数据则是无限增长的。另一方面,流式计算很难应对数据的过期更新操作。高速增长的数据往往具有鲜明的时效性,人们的信息获取也常常聚焦在近期的数据,因此当部分数据失去时效性时,需要结合既定的计算逻辑删除与这部分数据相关的计算结果,避免这部分过期数据的影响,而图的流式计算显然并不支持数据的删除更新。以节约内存为初衷而出现的图的流式计算,尽管结合了图模型和类似于数据流模型的流式计算,但显然并不适用于大规模图结构数据流的场景。

3.2 图数据流算法

目前有少量关于图数据流模型的研究工作,但这些工作并没有给出一个统一的图数据流形式化的定义。本节通过总结分析已有的图数据流工作以及存在的各种图数据流模型定义,给出同这些图数据流模型基本相融的、更具一般性的图数据流的

定义。本文提出的图数据流模型同时还能够很好地为现实应用中的大规模复杂数据流进行建模。

已有的图数据流模型可以分为3种,第一种是基于边流,即边的增删的图数据流。Song C Y等人^[10]提出图数据流下的强仿真计算,对应的图数据流定义为一个包含顶点集、边集、标签集以及时间戳函数的有向图,其中流动的元素为按时间戳先后顺序排列的边,并且在边流上引入了时间窗的概念。Angel A等人^[1]最先研究了在实时更新的边流上维护密集子图的算法,其边流定义为一个无限序列,序列中的每个元素为给定边及其边权的变化值(增/减)。边本身没有时间戳的概念,但是边权更新的操作带有时间戳。此外,FAN W F等人^[11]以及Choudhury S等人^[12]各自的图数据流子图同构中的模型均属于第一种模型。第二种是基于子图流,即数据流中的元素为一个个小的子图,边本身没有时间戳的概念,而流中的子图元素才有。IBM Watson研究中心的Aggarwal C C等人^[13]利用Min-Hash的相似度衡量方法,将以子图流为形式的图数据流采样成一个静态的抽样集,进而把研究图数据流上的子图模式挖掘转换成了静态的抽样集的频繁模式挖掘的问题。这个工作中的图数据流模型不支持删除操作,只考虑不断新增的小图,这些小图是作为整个图数据流对应的子图而定义的,彼此之间并不是独立的图。第三种则是图数据流中,元素是定义在不同点或边集上的独立图数据。Valari E等人^[14]提出了一个给定动态的大图集合的模型,即大图的数量是一个固定的常数,每次集合都将按增加一个新大图、删除一个最旧的大图的方式更新,进而形成动态的大图集合,然后在大图集合上研究Top k 的密集子图的计算算法。值得注意的是,这个图数据流模型中,不同的大图是定

义在不同的顶点集之上的,并不存在这些大图的公共超图的语义。

显然,目前的图数据流模型虽然都是针对大规模高速生成的图数据的,但缺乏统一的形式化定义以及普适的一般性含义。

3.3 图数据流模型定义

已有的相关图数据流工作中的模型定义虽然有所区别,但核心在于图模型与数据流模型的融合,即数据流模型下的数据元素为图模型定义的元素,图数据中点的意义主要通过边来体现,孤立点在现实中的意义有限,给定一个图的边集就能明确地获得非孤立点的点集。因此,本文以无限边序列作为图数据流的一般定义。

定义1 图数据流:图数据流是一个无限地随时间不断变长的边序列,其中每条边在序列中都有一个对应的时间戳,序列中边的时间戳从前往后是非降序的。

图1给出了图数据流的示例。其中边两端的顶点字母为顶点标签,而字母上标为对应的顶点标识符(ID)。虽然也有相关工作的图数据流模型是由无限的小图序列定义的,但小图均可以分解为一系列的边(孤立点除外),因此以无限边序列定义的图数据流更具一般性和统一性。

现实应用高速生成数据的同时往往伴随已有数据的高速失效。例如利用社交网络数据进行广告投放时,过时数据的参考价值显然没有新近数据高,而大量的过时数据又会带来高的处理开销,因此,往往可以利用时间窗的概念来聚焦更有时效性的数据。时间窗主要分为两种:基于数据规模的时间窗^[15]和基于时间跨度的时间窗^[16]。

定义2 时间窗:给定一个图数据流,基于数据规模的时间窗定义为包含最近的给定 N 条数据边的边序列区间,时间窗内

的数据则是最最近的 N 条边；基于时间跨度的时间窗定义为以当前时间为结尾的一个给定时间段(T_1, T_2)内的数据边所形成的边序列区间，而窗口内的边即为过去 $T_2 - T_1$ 时间内的所有数据边。

图2给出了带时间窗口的图数据流的模型示例。给定时间窗口后，在时间窗口内的边所生成的图称为当前时间下的图数据流的快照，如图3所示。

3.4 图数据流与动态图

图数据流与动态图^[17]既有联系，又有明显区别。带有时间窗口和快照概念的图数据流跟动态图的概念非常接近，甚至是动态图的细分概念。动态图是指给定一个大图，在大图上出现数据增删的动态行为的模型。因此，快照可以看作一个动态图，区别在于快照内的边带有时间戳，并且有时序先后的概念。同时快照新增的边

往往带有最大的时间戳，而删除的边则是带有最旧的时间戳。动态图不需要具有时间戳，但针对图的更新操作（增边或删边的操作，不是数据边本身）是具有时间的概念的。而图数据流中，组成的数据元素不一定都定义在同一个点集和边集，例如之前提到的大图流^[14]。此外，图数据流具有的时间信息更贴合现实世界的交互与联系，而动态图强调图结构随时间的变化，并不一定强调数据自身的时间信息。因此图数据流跟动态图的概念互有交集，但作为带有明确时间信息的图数据流，与动态图又有明显不同。

4 图数据流的研究前景：算法和系统

基于大规模图数据流的管理需求，针对图数据流模型的研究有广大的前景。传统的图计算的问题在这一模型下仍然会有

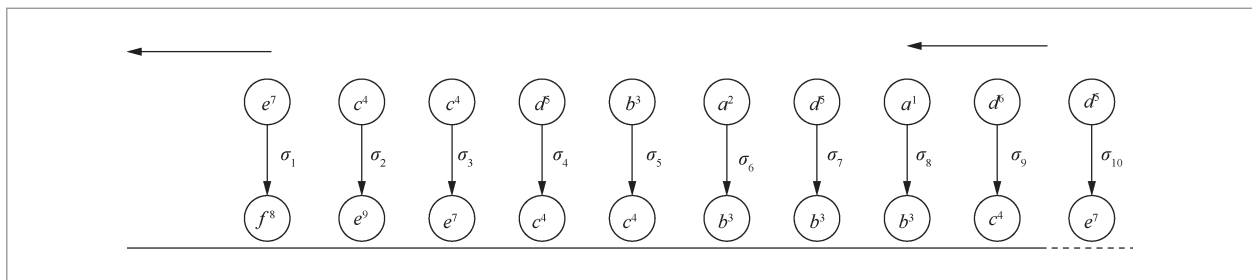


图1 图数据流示例

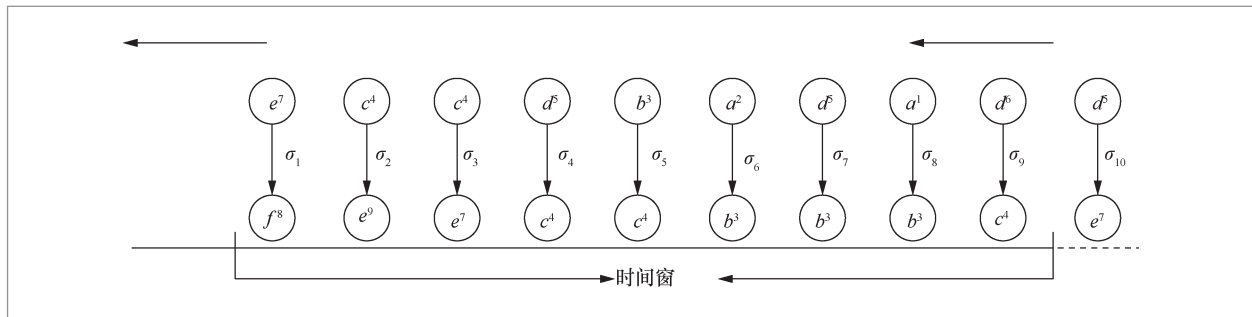


图2 带时间窗的图数据流示例

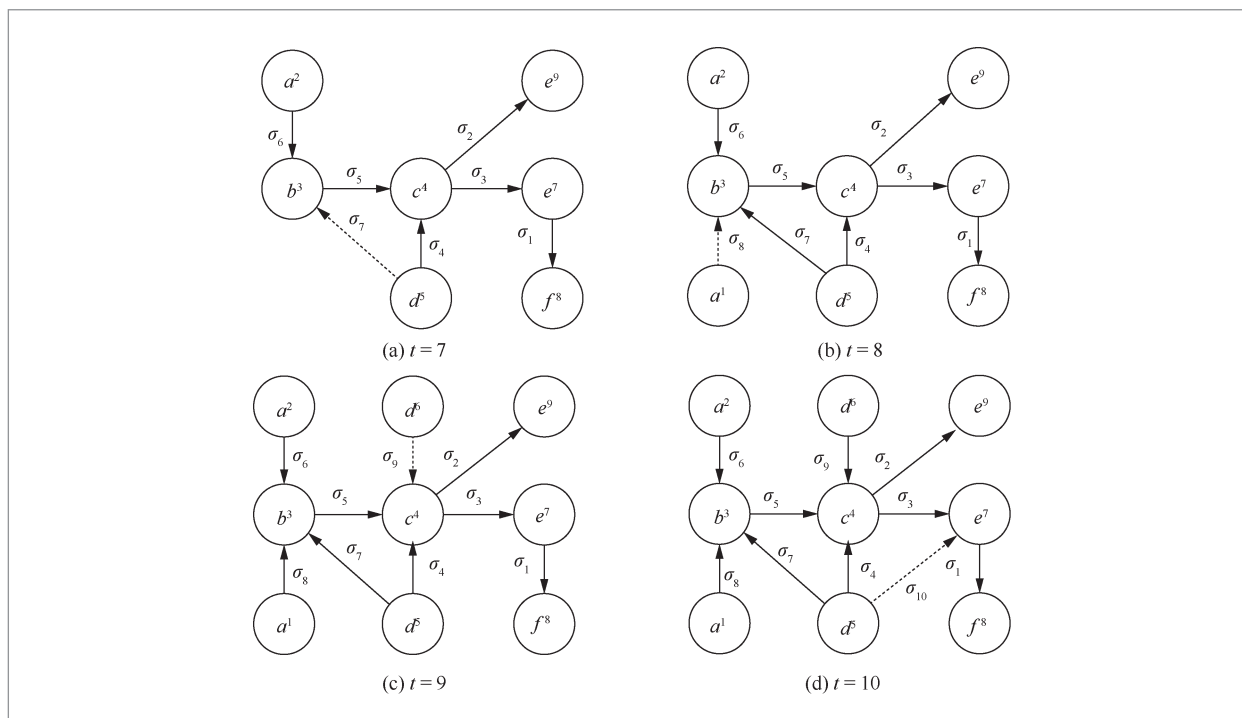


图3 图数据流中的快照示例

研究价值,包括三角形计算^[18-19]、最短路径^[20-21]、子图查询^[22]、子图挖掘^[23]以及图分类^[24]、图聚类^[25]等。鉴于这部分图计算问题已有充足的讨论和总结,本文聚焦在基于图数据流自身独特特点的相关研究前景,主要从研究问题和研究方法两个角度探讨,之后讨论图数据流管理系统需要的相关技术和框架。

4.1 问题的角度

从研究问题的角度分析图数据流的研究前景需要结合图数据流的重要特征。图数据流的一个非常本质的特征就是边的时间信息,这将为很多图计算问题赋予新的语义。首先是基于边的时序信息丰富查询语义。现实世界对象间的关联关系和交互行为往往有明确的先后关系,例如好友关系的传播、交通路径的递增时间戳等。因此,图数据流的查询问题可以引入时序

的限制。例如,在子图查询中引入对边的时序限制^[110]。除了时序限制,还可以引入基于边的时间间隔限制等,以丰富图数据流相关查询的语义。其次是利用图数据流的时间维度分析和挖掘数据的变化行为。对于两个顶点而言,在给定的时间段内可能出现多条相连的边,在时间维度上会有不同的行为表现。以社交网络的好友关系分析为例,假设将用户建模成点,而用户间的交互为边,那么在给定的一段时间内,两个用户的交互频率在时间维度上是上升还是下降对客户的好友关系有明显不同的意义。此外,两个用户之间交互内容的主题或者关键字等在时间维度上的变化也包含描述两者关系的潜在信息。因此,流场景使边有了行为的概念,相应的行为分析对图数据流的分析有很大的价值。还有就是充满挑战但又有极大研究价值的图数据流上的行为预测,即根据已有的图数据流上的数据分布、行为变化,

综合关联的结构信息,预测未来一段时间可能出现的图数据特征,包括分布、关联等。例如,在交通网络上将交通站点建模成顶点,站点间的车流建模成边,这个模型下一个值得研究的问题就是如何根据过去几个小时的车流信息预测未来一个时间段内各条道路可能的车流密度。在网络信息传输管道上,如何根据当前的网络状况预测接下来的网络流量拥堵情况,进而进行更合理的路由调度等,也是值得研究的问题。

4.2 方法的角度

从研究方法的角度看图数据流的研究前景,则要结合目前技术上处理图数据流遇到的核心挑战:加速计算带来的更新开销。以图数据流上的子图查询为例,如果通过构建复杂的索引来加速子图查询,那么在数据更新时必然需要更新相应的索引。如果不构建索引,则更新发生时,为了处理子图查询需要重新进行计算,而实际上一次更新往往涉及的只是局部的数据,完全重新计算将会有大量计算结果同上一个时间点的计算结果重叠,造成冗余计算。因此,处理图数据流上的计算的关键是如何避免冗余计算,同时还能加速查询的响应。首先,需要考虑保存哪些已有的计算结果。尽量避免或删除对之后的计算缺乏利用价值的计算结果,优先考虑能够多次重复利用的计算结果,进而能够更大限度地提高性能。其次,在确定需要保存的计算结果上,要相应地保存组织的策略,即如何优化空间开销。对于有重叠的中间结果,需要尽量避免重复占用存储开销。因此,图数据流计算的首要研究思路就是加速计算的中间结果的选取与保留数据的组织形式。其次则是利用并行技

术加速图数据流的更新与计算。在高速更新的图数据上,单个更新往往涉及的只有部分图数据,多个更新可能彼此之间不涉及读写冲突,通过并行技术加速这些无冲突的更新显然能够大幅提高系统的吞吐量。

4.3 图数据流管理系统

目前还没有针对图数据流模型的管理系统,结合图数据流管理的需求和研究前景,本节设计了图数据流管理系统的大致框架,如图4所示。框架主要分为3层,分别是图数据流的基本数据层、算法和索引的逻辑层以及向上支撑的各种应用的应用层。首先是基本数据层,该层负责管理图数据流的最基本的边序列的存储、增删和基本的访问。其中访问操作包括图数据的一些基本操作,如访问节点度数、节点的邻居等。其次是包含核心算法和相应的索引数据的逻辑层。其包含的索引数据能够根据数据层的增删而进行更新,从而保持与数据层的逻辑一致性。而核心算法则涉及图相关的基本算法,包括路径、子图等经典计算以及图数据流下边的行为分析等。同时,还需要提供图数据流一般性的聚集查询与相关数据统计等逻辑接口。逻辑层之上则是需要支撑的应用层,针对现实应用数据的分析需求,定制更为复杂的计算接口。例如进行基于子图挖掘的事件侦测、城市交通的智能分析与管理和信息或话题的传播跟踪等。

5 结束语

高度数字化的社会生活带来了大量高速生成的应用数据,分析挖掘这些应用数

据能给人们的生产生活带来极大的正反馈。然而目前主流的数据管理模型和技术(包括传统的关系模型和图模型等)难以应对大规模复杂数据流的管理需求,图数据流应运而生。图数据流是图模型和数据流模型相融合形成的数据模型,一方面具备图模型对复杂数据的强表达能力,另一方面又结合数据流的更新场景来建模复杂数据的高速生成行为。因此,图数据流模型及其研究具有非常重要的现实意义。目前,关于图数据流的研究工作较少,已有的工作重点分散并且不系统,因此图数据流的研究前景广阔。未来针对图数据流的研究工作有3个较为重要的路线:一是基于已有的大量图算法/算子等,研究图数据流场景下相关计算操作的实现和优化;二是基于图数据流场景独自的特点和现实应用需求,提出新的图数据流研究问题及其解决方案;三是针对图数据流模型本身的数据管理系统,研究其存储、索引、数据的增删改查的基本操作、并发管理以及一致性保证等问题,进而设计结合现实需求的图数据流管理系统。图数据流的研究具有重要的现实意义和广阔的研究前景。

参考文献:

- [1] ANGEL A, KOUZAS N, SARKAS N, et al. Dense subgraph maintenance under streaming edge weight updates for real-time story identification[J]. VLDB Journal, 2014, 23(2): 175-199.
- [2] SAKAKI T. Earthquake shakes Twitter users: real-time event detection by social sensors[C]//The 19th International Conference on World Wide Web, April 26-30, 2010, Raleigh, USA. New York: ACM Press, 2010: 851-860.
- [3] TOSHNIWAL A, TANEJA S, SHUKLA A, et al. Storm@Twitter[C]//The 2014

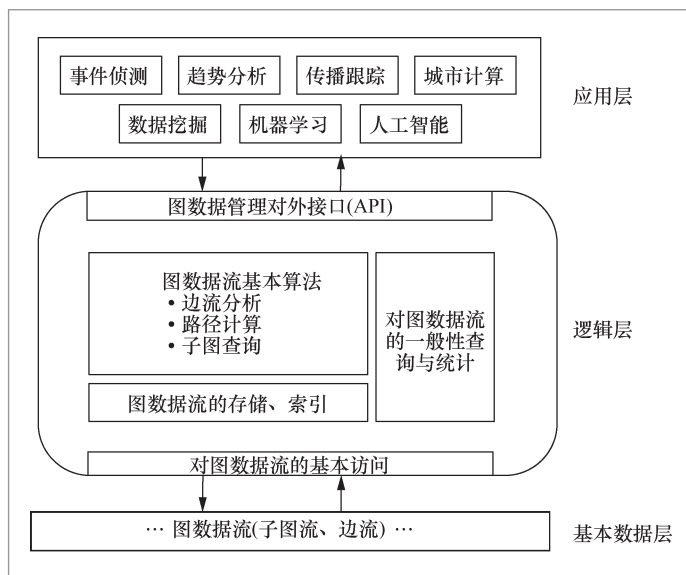


图4 图数据流管理系统框架

- ACM SIGMOD International Conference on Management of Data, June 22-27, 2014, Snowbird, USA. New York: ACM Press, 2014: 147-156.
- [4] BAR-YOSSEFF Z, KUMART R, SIVAKUMAR D. Reductions in streaming algorithms, with an application to counting triangles in graphs[C]//The 13th Annual ACM-SIAM Symposium on Discrete Algorithms, January 6-8, 2002, San Francisco, USA. Philadelphia: Society for Industrial and Applied Mathematics, 2002: 623-632.
 - [5] JOWHARI H, GHODSI M. New streaming algorithms for counting triangles in graphs[C]//The 11th Annual International Conference, August 16-19, 2005, Kunming, China. [S.l.:s.n.], 2005: 710-716.
 - [6] BURIOL L S, FRAHLING G, LEONARDI S, et al. Counting triangles in data streams[C]//The 25th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, June 26-28, 2006, Chicago, USA. New York: ACM Press, 2006: 253-262.

- [7] UNEL G, FISCHER F, BISHOP B. Answering reachability queries on streaming graphs[C]//The 1st International Workshop Stream Reasoning, May 31, 2009, Heraklion, Greece. [S.l.:s.n.], 2009.
- [8] DEMETRESCU C, FINOCCHI I, RIBICHINI A. Trading off space for passes in graph streaming problems[J]. ACM Transactions on Algorithms, 2006, 6(1): 1–17.
- [9] MCGREGOR A. Graph stream algorithms: a survey[J]. ACM SIGMOD Record, 2014, 43(1): 9–20.
- [10] SONG C Y, GE T, CHEN C, et al. Event pattern matching over graph streams[J]. Proceedings of the VLDB Endowment, 2014, 8(4): 413–424.
- [11] FAN W F, WANG X, WU Y, et al. Incremental graph pattern matching[J]. ACM Transactions on Database Systems, 2013, 38(3): 925–936.
- [12] CHOUDHURY S, HOLDER L, CHIN G, et al. A selectivity based approach to continuous pattern detection in streaming graphs[J]. Computer Science, 2015, 93(8): 939–945.
- [13] AGGARWAL C C, LI Y, YU P S, et al. On dense pattern mining in graph streams[J]. Proceedings of the VLDB Endowment, 2010, 3(1–2): 975–984.
- [14] VALARIE, KONTAKI M, PAPADOPOULOS A N. Discovery of top-k dense subgraphs in dynamic graph collections[C]//The 24th International Conference on Scientific and Statistical Database Management, June 25–27, 2012, Chania, Greece. Heidelberg: Springer, 2012: 213–230.
- [15] CHI Y, WANG H, YU P S, et al. Moment: Maintaining closed frequent itemsets over a stream sliding window[C]//The 4th IEEE International Conference on Data Mining, November 1–4, 2004, Brighton, UK. Piscataway: IEEE Press, 2004: 59–66.
- [16] SOLOMON M M. Algorithms for the vehicle routing and scheduling problems with time window constraints[J]. Operations Research, 1987, 35(2): 254–265.
- [17] RODITTY L, ZWICK U. A fully dynamic reachability algorithm for directed graphs with an almost linear update time[J]. SIAM Journal on Computing, 2016, 45(3): 712–733.
- [18] SCHANK T, WAGNER D. Finding, counting and listing all triangles in large graphs, an experimental study[C]//The 36th Annual ACM Symposium on Theory of Computing, June 13–16, 2004, Chicago, USA. Heidelberg: Springer, 2005: 606–609.
- [19] KOLOUNTZAKIS M N, MILLER G L, PENG R, et al. Efficient triangle counting in large graphs via degree-based vertex partitioning[J]. Internet Mathematics, 2012, 8(1–2): 161–185.
- [20] ZILIASKOPOULOS A K, MAHMASSANI H S. Time dependent, shortest-path algorithm for real-time intelligent vehicle highway system applications[J]. Transportation Research Record, 1993(1408): 94–100.
- [21] CLIMACO J C N, MARTINS E Q V. A bicriterion shortest path algorithm[J]. European Journal of Operational Research, 1982, 11(4): 399–404.
- [22] YUAN Y, WANG G, WANG H, et al. Efficient subgraph search over large uncertain graphs[J]. Proceedings of the VLDB Endowment, 2011, 4(11): 876–886.
- [23] YAN X, HAN J. gSpan: graph-based substructure pattern mining[C]//The 2002 IEEE International Conference on Data Mining, December 9–12, 2002, Maebashi City, Japan. Piscataway: IEEE Press, 2002: 721–724.

- [24] KASHIMA H, INOKUCHI A. Kernels for graph classification[C]// The 2002 IEEE International Conference on Data Mining, December 9-12, 2002, Maebashi City, Japan. Piscataway: IEEE Press, 2002.
- [25] SCHAEFFER S E. Graph clustering[J]. Computer Science Review, 2007, 1(1): 27-64.

作者简介



李友焕 (1991-), 男, 北京大学计算机科学技术研究所博士生, 主要研究方向为图数据流的管理、数据流算法、社交网络分析等。



邹磊 (1981-), 男, 北京大学计算机科学技术研究所教授、博士生导师, 主要研究方向为海量图数据的管理、基于图的RDF知识库数据管理、图数据库、知识图谱构建与应用等。

收稿日期: 2018-05-15