



文章编号:1001-9081(2018)04-0978-09

DOI:10.11772/j.issn.1001-9081.2017092202

# 大数据相似性连接查询技术研究进展

马友忠<sup>1,2\*</sup>, 张智辉<sup>3</sup>, 林春杰<sup>1,2</sup>

(1. 洛阳师范学院 信息技术学院, 河南 洛阳 471934;  
2. 河南省电子商务大数据处理与分析重点实验室(洛阳师范学院), 河南 洛阳 471934;  
3. 洛阳铁路信息工程学校 计算机教研室, 河南 洛阳 471900)  
(\*通信作者电子邮箱 ma\_youzhong@163.com)

**摘要:**为了深入理解和全面把握大数据相似性连接查询技术的研究进展,更好地促进其在图片聚类、实体解析、相似文档检测、相似轨迹检索等领域的广泛应用,对大数据相似性连接查询技术相关研究工作进行了深入调研和分析。首先对相似性连接查询的基本概念进行了介绍,然后分别对集合、向量、空间数据、概率数据、字符串等不同类型大数据的相似性连接查询相关研究工作进行了深入研究,对其优缺点进行了分析和总结。最后,指出了大数据相似性连接查询面临的若干挑战性问题及未来的研究重点。

**关键词:**大数据;相似性连接查询;MapReduce 框架;K 最近邻

**中图分类号:**TP311.13    **文献标志码:**A

## Research progress in similarity join query of big data

MA Youzhong<sup>1,2\*</sup>, ZHANG Zhihui<sup>3</sup>, LIN Chunjie<sup>1,2</sup>

(1. School of Information Technology, Luoyang Normal University, Luoyang Henan 471934, China;  
2. Henan Key Laboratory for Big Data Processing and Analytics of Electronic Commerce  
(Luoyang Normal University), Luoyang Henan 471934, China;  
3. Department of Computer, Luoyang Railway Information Engineering School, Luoyang Henan 471900, China)

**Abstract:** In order to deeply understand and fully grasp the research progress of similarity join query technology of big data and to promote its wide application in image clustering, entity resolution, similar document detection, similar trajectory retrieval, a comprehensive survey was conducted on similarity join query technology of big data. Firstly, the basic concepts of similarity join query were introduced; then intensive study on the big data similarity join research works for different data types, such as set, vector, spatial data, probabilistic data, string and graph was elaborated, their advantages and disadvantages were analyzed and summarized. Finally, some challenging research problems and future research priorities in big data similarity join query were pointed out.

**Key words:** big data; similarity join query; MapReduce framework; K-Nearest Neighbors (KNN)

## 0 引言

相似性连接查询是指给定两个数据对象集合  $R$  和  $S$ (集合、向量、空间数据、字符串等),一个度量函数  $sim$  和一个相似度阈值  $\varepsilon$ ,找出所有分别来自  $R$  和  $S$  的相似度大于阈值  $\varepsilon$  的数据对象对。相似性连接查询是一种应用广泛且非常重要的操作,是很多数据分析及数据挖掘任务的基本操作,如分类、聚类、异常检测、重复文档检测等。

相似性连接查询存在两大挑战:一是相似度计算代价大,尤其是当数据类型比较复杂或者是数据维度比较高时,两个数据对象之间的相似度的计算将耗费很多时间。二是扩展性问题。随着大数据时代的到来,数据的规模日益增长,传统的集中式算法或串行算法已经不能在可接受的时间内完成大规模数据集的相似性连接查询任务。因此,如何借助最新的计算技术,设计具有良好扩展性的相似性连接查询算法,成为目

前大数据相似性连接查询的重要研究内容。MapReduce 是一个由 Google 最先提出的分布式计算软件框架,具有高可扩展性、高可用性、高容错性等特点,可以支持大规模数据的分布式处理,目前已经成为大数据处理的首选平台。为了解决大规模复杂数据类型的相似性连接查询问题,很多学者针对基于 MapReduce 框架的相似性连接查询算法进行了深入研究,提出了很多创新的解决方案。

根据分类标准的不同,相似性连接查询可以有不同的分类方法。可以按照数据对象类型、对象集合的动态性,以及查询结果和处理方式的不同等标准来进行划分。不同分类标准得到的类别之间可能会有重叠,并且不同的数据类型,对象之间的相似度量函数往往不同,因此所采用的方法也不相同,本文将主要按照不同数据对象类型的连接查询方法进行分别描述。表 1 描述了不同数据类型对应的常用相似度度量函数,其中,集合、向量和字符串的相似度度量有所重叠,部分方

收稿日期:2017-09-11;修回日期:2017-11-27。    基金项目:国家自然科学基金资助项目(61602231);国家重点研发计划项目(2016YFE0104600);河南省科技开放合作项目(172106000077,152106000048);河南省高等学校重点科研项目(16A520022)。

作者简介:马友忠(1981—),男,河南项城人,副教授,博士,CCF 会员,主要研究方向:大数据、Web 数据管理; 张智辉(1979—),男,河南洛阳人,讲师,硕士,主要研究方向:数据挖掘; 林春杰(1981—),男(朝鲜族),吉林吉林人,讲师,硕士,主要研究方向:数据挖掘、粗糙集。



法可以通用。为便于描述,本文中集合主要采用杰卡德相似度和余弦相似度,向量主要采用欧氏距离,字符串主要采用编辑距离。

表1 相似度度量函数  
Tab. 1 Similarity measurement functions

| 序号 | 数据类型 | 相似度度量函数                         |
|----|------|---------------------------------|
| 1  | 集合   | 杰卡德相似度、余弦相似度                    |
| 2  | 向量   | 欧氏距离、余弦相似度                      |
| 3  | 空间数据 | 空间交叠数、特定空间关系                    |
| 4  | 概率数据 | EMD 距离 (Earth Mover's Distance) |
| 5  | 字符串  | 编辑距离、杰卡德相似度                     |
| 6  | 图    | 图编辑距离                           |

另外,关于集中式的相似性连接查询方法,文献[1~3]已经进行了综述,本文不再详细描述。庞俊等<sup>[4]</sup>对基于

MapReduce 框架的海量数据相似性连接工作进行了综述,但是主要包含的是 2013 年及以前的部分研究工作。还有部分学者<sup>[5~6]</sup>对典型的基于 MapReduce 的相似性连接查询算法进行了实验验证和比较。本文将针对截至 2017 年的云计算环境下的大数据相似性连接查询技术进行更全面、深入的分析。

## 1 集合相似性连接查询

集合相似性连接查询广泛应用于文本分类、聚类、重复网页检测等应用中,文本、网页都可以表示成单词的集合。集合的相似性度量主要包括杰卡德相似度、余弦相似度、重叠相似度和 Dice 相似度等。随着数据规模的不断扩大,单机环境下的相似性连接算法已经不能满足性能要求,有很多学者基于 MapReduce 框架,提出了若干种针对大规模集合数据的相似性连接查询处理方案,如表 2 所示。

表2 集合相似性连接查询方案  
Tab. 2 Related works on set similarity join query

| 方案名称            | 代表算法  | 优点                                       | 缺点   |
|-----------------|---|--|--|
| 穷举方案            | Brute force 算法、索引算法 <sup>[7]</sup>                          | 可以充分利用 MapReduce 的并行特点,加快计算速度            | Brute force 算法没有进行任何过滤,任意两个集合都需要比较一次;<br>基于索引的算法重复比较次数会比较多   |
| 前缀过滤            | 单前缀过滤 <sup>[8~9]</sup> 、多前缀过滤 <sup>[10]</sup>               | 充分利用了前缀的过滤功能                             | 每个记录要复制 $ prefix $ 次,对于较长的集合来讲,数据复制率较高;<br>重复比较:两个记录如果有 $k$ 个公共的项,则会重复比较 $k$ 次                                 |
| Word-Count-Like | Pairwise Join <sup>[11]</sup> 、V-SMART-Join <sup>[12]</sup> | 不复制记录本身,只复制了权重,网络传输代价低;<br>无重复计算,每一对计算一次 | 没有利用前缀的过滤功能;<br>会产生很多不必要的候选对   |
| 混合方案            | Self-Join <sup>[13]</sup>                                   | 将前缀方案与 Word-Count-Like 方进行了结合;<br>没有重复计算 | 因候选对中没有集合的全部信息,所以需要在候选对的基础上进行进一步的处理,进行数据的传输等;<br>候选对的数量与 ALL Pairs 方法中候选对的数量是一样的,即只要两个集合的前缀中至少含有一个公共项,就成为一个候选对 |
| 基于划分的方法         | FS-Join <sup>[14]</sup> 、Greedy + <sup>[15]</sup>           | 避免了重复计算;<br>可以确保 Map 端和 Reduce 端的负载均衡    | 算法性能对枢轴 (Pivot) 的选择以及枢轴个数有较大依赖   |
| 基于位置敏感哈希的方法     | PLSH <sup>[16]</sup>  | 可以减少伪正例和伪反例的个数;<br>可以实现负载均衡              | 是一种近似计算,不能返回所有结果   |

表 2 描述了已有的大规模集合相似性连接查询方案,并按照采用的技术不同,将其分为六类:穷举方案、前缀过滤、Word-Count-Like、混合方案、基于划分的方法和基于位置敏感哈希的方法。各类方案的主要优缺点和代表算法详细描述如下。

### 1) 穷举方案。

文献[7]较早利用 MapReduce 框架对相似性连接查询问题进行了研究,并提出了 Brute force 算法和基于索引的算法。其中,Brute force 算法通过精确匹配找出所有满足条件的文本对。该算法在 Map 阶段使用暴力法,计算出每一个文本和其他所有文本的相似度,然后输出所有相似度不为零的文本对;在 Reduce 阶段,针对每一个文本求出与其相似度最大的  $k$  个文本。该算法可以求出精确的结果,但是计算代价比较大。索引算法可以在一定程度上降低计算代价,提升性能,但是结果的精度会有所下降。总体来看,Brute force 算法没有进行任何过滤,任意两个集合都需要比较一次;基于索引的算法虽然可以进行一定程度的过滤,但是过滤效果不理想,重复比较次数仍然会比较多。

### 2) 前缀过滤。

文献[8]在 MapReduce 框架下,提出了一种基于前缀过滤(prefix filtering)技术的大规模集合数据相似性连接查询方案。文献[8]中处理的数据对象是一些记录(record)的集合,每一条记录由记录标识和其他若干个属性组成,其中连接属性是项(token)的集合。前缀过滤是一种很有效的过滤方法,其基本思想是:假设数据集中所有的项(token)按照其出现频率由低到高进行排序,得到一个序列  $O$ ,数据集中的每一个集合都按照序列  $O$  进行排列,集合  $A$  的  $p$ -前缀就是集合  $A$  的前  $p$  个项。假定相似性度量采用杰卡德相似度,阈值为  $t$ ,则如果  $J(A, B) \geq t$ ,那么  $A$  的  $(\lfloor A \rfloor - \lceil t \cdot |A| \rceil + 1)$ -前缀和  $B$  的  $(\lfloor B \rfloor - \lceil t \cdot |B| \rceil + 1)$ -前缀至少含有一个公共的项(token)。即如果两个集合的前缀没有公共的项,则这两个集合肯定不相似。根据上述基本思想,文献[8]在 MapReduce 框架下实现了前缀过滤方案。该方案主要分为三个阶段,分别为项排序(Token Ordering)、相似对生成(RID-Pair Generation)和最终结果生成(Record Join),每个阶段由一个或多个 MapReduce Job 来完成。项排序阶段的主要目的是通过计算数据集合的



相关统计信息,得到具有较好过滤效果的集合前缀,具体方法是计算出集合中所有项(token)出现的频率,然后按照出现频率将所有项进行升序排列,作者分别提出了基本排序法(basic token ordering)和一阶段排序法(using one phase to order tokens)来完成排序任务。相似对生成阶段主要是根据连接的属性,找出相似度大于给定阈值的记录对,该阶段由一个MapReduce Job 完成。在 Map 阶段,针对每一个输入记录,首先抽取出其 ID 和连接属性,然后根据前一阶段得到的项序列,计算出连接属性的前缀,接着根据连接属性的前缀构建倒排索引。针对每一个记录,会生成若干个 $\langle key, value \rangle$ 对的集合,其中: $key$ 是记录前缀中的每一个项; $value$ 包括该记录的 ID 和连接属性。在 Reduce 阶段,具有相同  $key$  值的  $value$  组合在一起,发送到同一个 reducer 中。同一个  $key$  就对应着包含该  $key$  的记录列表,这些记录有可能相似,针对这些记录,作者提出了嵌套循环链接和 PPJoin + 两种方法来计算相似度大于阈值  $t$  的记录对,包括记录 ID 和相似度。在最终结果生成阶段,将前一阶段得到的相似对集合与原始数据集合进行连接,获取记录的完整信息,得到最终结果。文献[9]的解决思路与文献[8]类似,改进之处在于文献[9]除了前缀过滤之外,又增加了长度过滤,这样就可以进一步提高性能。

为了进一步提高过滤效果,减少候选对的数量,文献[10]中提出了一种基于多前缀的集合相似性连接方案。文献[8]中所使用的前缀过滤技术虽然在一定程度上减少了候选对的数量,但是在候选对中仍然有很多是不相似的,因为即使两个对象的前缀有公共的项,这两个对象也不一定相似。文献[10]通过分析、观察发现,集合的前缀是按照某个项的排序生成的,不同的排序,所产生的前缀就不一样。基于该思想,文献[10]为每个对象按照不同的排序生成多个不同的前缀,其中用第一个前缀来建立倒排索引,用其他的前缀来进行进一步的过滤,最终通过过滤的候选对才需要进行最后的验证,从而大大减少了计算的代价。同时,文献[10]还提出了一个代价模型,来确定序列(即前缀)的数量,并提出了多种排序方案,主要包括随机选择(random selection)、基于反序列的随机选择(random selection with reverse ordering)和基于词频的全局排序(TF as the first global ordering)。作者还在单机环境及 MapReduce 框架下分别对该算法进行了实现及验证,实验结果表明,基于多前缀的集合相似性连接方法具有较好的性能。

但是基于前缀过滤技术的方案也存在一些不足之处:a)网络传输代价较大。每个集合都要复制多次,复制的次数等于前缀的长度,因此对于较长的集合来讲,数据复制率太高,会导致网络传输代价比较大。b)重复比较次数比较多。对于任意两个集合,如果它们的前缀中有  $k$  个公共项,则会重复比较  $k$  次。

### 3) Word-Count-Like 方案。

为了克服前缀方案中存在的缺点,文献[11]中提出了一种“Word-Count-Like”的解决方案 Pairwise Join,之所以取名为“Word-Count-Like”,是因为该方案充分利用了 MapReduce 框架的特性,基本思路类似于 Word-Count 程序。文献[11]主要解决的是文档的相似性连接问题,每一个文档可以被表示成一个词的权重的向量,相似性度量采用的是权重向量的内积来表示。该方案主要分两个阶段来完成,第一阶段主要生成倒排索引,第二阶段用来计算任意两个文档的相似度,每个阶

段均由一个 MapReduce Job 来实现。在 Job 1 中:Map 阶段针对每一个输入的文档向量 $\langle i, d_i \rangle$ ,输出 $\langle key, value \rangle$ 的集合 $\langle t, \langle i, d_i[t] \rangle \rangle$ ,其中: $i$ 是文档标识; $d_i$ 是权重向量; $t$ 是单词; $d_i[t]$ 是 $t$ 在文档 $i$ 中的权重。在 Reduce 端,同一个单词  $t$  对应的  $value$  被组合在一起,按照 $\langle t, [\langle i, d_i[t] \rangle, \langle j, d_j[t] \rangle, \dots] \rangle$ 的形式输出。在 Job 2 中,Map 阶段针对每一个 $\langle t, [\langle i, d_i[t] \rangle, \langle j, d_j[t] \rangle, \dots] \rangle$ 进行处理,输出 $\langle \langle i, j \rangle, w = d_i[t] \cdot d_j[t] \rangle$ 的集合,其中: $\langle i, j \rangle$ 是两个文档的 ID 对; $w$ 是 $t$ 在两个文档中的权重的乘积。在 Reduce 端,与同一个 $\langle i, j \rangle$ 对对应的  $w$  被组合在一起,对这些权重求和,得到的最终结果就是文档对 $\langle i, j \rangle$ 的相似度。

文献[12]的基本思路与文献[7]类似,提出了 V-SMART-Join 方案。不同之处在于文献[12]中对处理方案进行了一般化,处理的数据对象类型进行了扩充,可以处理集合、多重集合以及向量,相似性度量可以采用内积、余弦相似度以及杰卡德相似度等。此外,文献[12]中还提出了一些解决方案,用来高效处理倾斜数据,并在大规模真实数据集上进行了充分的实验,实验结果表明,文献[12]方案的性能比文献[8]方案的性能要高 30 倍以上。

Word-Count-Like 方案的优点是避免了重复比较,任何一对文档只需要比较一次。另外,文档本身没有被重复复制,传输的仅仅是各个单词对应的权重,所以大大降低了网络传输的代价。但是该类方案也存在一些局限性,任何两个文档只要包含一个公共元素,都需要比较一次,没有利用到相似度阈值和前缀的过滤功能,因此会产生很多不必要的候选对。

4) 混合方案。前缀过滤方案和 Word-Count-Like 方案各有优缺点,文献[13]中结合上述两种方案的优点,提出了一种混合方案 Self-Join,从而进一步提高了集合相似性连接的效率。文献[13]对文献[11]进行了扩充,在建立倒排索引时候,利用前缀过滤技术来缩短索引列表的长度,从而可以大幅减少中间结果的数量。由于采用了前缀进行过滤,对于任意一个候选对中的文档来说,可能只包含一部分权重信息,因此还需要两次远程操作来获取其相应的权重信息,从而计算出最终的相似度。由于远程操作的代价一般比较大,为了解决这一问题,文献[13]又提出了一种优化方案——SSJ-2R(Double-Pass MapReduce Prefix-Filtering with Remainder File)。

该方案的优点是将前缀过滤技术与 Word-Count-Like 方案进行了结合,从而减少了候选对的数量;没有重复计算,每一对只计算一次。不足之处在于,因候选对中没有集合的全部信息,所以需要在候选对的基础上,进行进一步的处理,进行数据的传输等;候选对的数量与 ALL Pairs 方法中候选对的数量是一样的,即只要两个集合的前缀中至少含有一个公共项,就成为一个候选对。

5) 基于划分的方法。文献[14]中提出了一种基于垂直划分的相似性连接查询算法 FS-Join,可以有效避免重复计算,并可以实现 Map 端和 Reduce 端的负载均衡。但是,枢轴(Pivot)的选择以及枢轴个数的确定对该算法的性能影响较大,需要花费较大的代价确定枢轴及其个数。为了减少重复计算的代价,Greedy +<sup>[15]</sup>设计了一种新的划分机制,将集合划分成若干子集,并确保:如果两个集合相似,那么它们一定有相同的子集。

6) 基于位置敏感哈希的方法。个性化位置敏感哈希(Personalized Locality Sensitive Hashing, PLSH)方法<sup>[16]</sup>在传



统位置敏感哈希技术的基础上,提出了一种新的采用灵活阈值的分块技术(banding technique),从而可以大幅减少伪正例的个数,提高计算效率,并能够实现伪正例和伪反例之间的平衡。

## 2 向量相似性连接查询

向量数据相似性连接查询针对的数据类型是向量,包括低维向量和高维向量,如图形、图像、Web文档、基因表达数据等多种数据经过处理后,都可以用向量来表示。向量的相似性度量也有很多种,包括余弦相似度、杰卡德相似度、欧氏距离和闵可夫斯基距离等,本文将主要研究基于欧氏距离的解决方案。根据返回结果的不同,向量相似性连接查询可以分为基于阈值的连接查询、Top- $k$ 连接查询和KNN连接查询,具体信息如表3所示。

**阈值相似性连接查询:**文献[17]中提出了一种称为DAA(Dimension Aggregation Approximation)技术的降维方法,并在此基础上提出了基于MapReduce框架的并行相似性连接查询算法。DAA是受分段累积近似法(Piecewise Aggregate Approximation, PAA)技术的启发,PAA是时间序列中一种比较有效的数据降维方法,其核心思想是:将一个长度为 $n$ 的时间序列,划分成 $m$ 段,形成一个长度为 $m$ 的新序列,用每一段

内的所有元素的均值作为新序列的元素值,基于新序列,可以构建一个新的距离函数,该距离是原始序列距离的下界,因此可以基于降维后的序列来对原始向量进行过滤。

在DAA技术的基础上,结合MapReduce框架,作者以嵌套循环连接算法为基础,提出了两种新的并行化算法:一阶段过滤验证算法OSFR(One-Stage Filtering-and-Refinement)和两阶段过滤验证算法TSFR(Two-Stage Filtering-and-Refinement)。在进行数据传输的过程中,OSFR除了传输降维后的向量之外,原始向量也要一起传输,在得到候选对以后,在验证阶段就可以直接利用原始向量的信息计算原始距离。但是当向量维度特别高时,这种方案的网络传输代价比较大。TSFR算法分为两个阶段来完成,在第一个阶段仅仅使用降维后的低维向量进行计算,得到候选对以后,再利用第二个阶段去获取原始向量信息,进行最后的验证。该方案可以减少网络传输的代价,但是在第二个阶段需要远程获取原始向量的信息,也需要一定的额外开销。最后,作者提出了一个代价模型,用来进行算法选择,通过分析和实验验证表明,OSFR算法比较适合于较低维向量,TSFR算法比较适合于超高维向量。文献[17]方案可以以较低的代价进行过滤,从而减少不必要的比较。但是,该方案的时间复杂度仍然是 $O(n^2)$ ,即任意两个向量在低维空间上都需要比较一次。

表3 向量相似性连接查询技术  
Tab. 3 Related works on vector similarity join query

| 查询类型        | 代表工作                             | 维度 | 是否精确  | 主要技术                                  |
|-------------|----------------------------------|----|-------|---------------------------------------|
| 阈值连接        | OSFR/TSFR <sup>[17]</sup>        | 高维 | 是     | 基于DAA的降维,基于DAA的并行算法                   |
|             | MR-DSJ <sup>[18]</sup>           | 低维 | 是     | 基于网格划分的过滤                             |
|             | PHiDJ <sup>[19]</sup>            | 低维 | 是     | 基于网格划分的过滤                             |
|             | MRSimJoin <sup>[20-21]</sup>     | 高维 | 是     | 基于中枢的数据划分法                            |
|             | MRSJ_IDS <sup>[22]</sup>         | 高维 | 是     | 增量式相似性连接                              |
|             | FACET <sup>[23]</sup>            | 高维 | 是     | 基于前缀过滤和长度过滤                           |
|             | SAX-Based HDSJ <sup>[24]</sup>   | 高维 | 是     | 基于符号累积近似法(SAX)的过滤技术                   |
|             | MP-V-SJQ <sup>[25]</sup>         | 高维 | 是     | 基于多PAA的过滤技术                           |
| KNN连接       | Grid-Based SJ <sup>[26]</sup>    | 低维 | 是     | 基于网格的数据划分方法                           |
|             | SJT <sup>[27]</sup>              | 高维 | 是     | 基于相似性连接树(SJT)的数据划分方法                  |
| Top- $k$ 连接 | H-zkNNJ <sup>[28]</sup>          | 高维 | 否     | 基于空间填充曲线技术的近似KNN连接                    |
|             | PGBJ <sup>[29]</sup>             | 低维 | 是     | 基于维诺图的数据划分及KNN连接查询                    |
|             | DSGMP-J/DSGMP-J <sup>[30]</sup>  | 低维 | 精确/近似 | 网格划分与维诺图划分两种方案                        |
| Top- $k$ 连接 | Top- $k$ Join <sup>[31-32]</sup> | 低维 | 是     | Divide-and-conquer和branch-and-bound算法 |
|             | Spark <sup>[33]</sup>            | 高维 | 是     | 基于LSH的距离和基于Spark的并行算法                 |

文献[18]中提出了一种基于网格划分方法的大规模向量相似性自连接处理方案,该方案的核心思想是利用网格对数据对象进行划分,并按照相应的过滤技术,减少计算比较的次数,降低网络传输的代价。以二维向量为例来进行介绍:假设距离阈值为 $\varepsilon$ ,首先将空间进行网格划分,格子的宽度为 $\varepsilon$ ,对于每一个单元格内的对象,只可能与自身以及周围的八个单元格内的对象相似,这样就不需要与其他单元格内的对象进行比较,可以减少大量的计算量。为了保证计算的完整性,每一个单元格内的数据都需要复制 $3^d$ 次( $d$ 是向量的维数),在不同的Reduce上进行计算。实际上有些计算和复制是重复的,可以采取一定的措施来减少数据复制的比率,即每一个单元格内的对象只与其自身和左下方的单元格内的对象比较即可,这样也可以保证计算的完整性,基于这种方法,数据复制的比率可以降为 $2^d$ 。另外,作者还提出了一些优化方法,在Reduce端进一步减少计算的次数。该方法具有很好的

并行特性,很容易在MapReduce框架下实现。其不足之处是该方法只适合于维度较低的情况,一旦维度比较高时,其性能急剧下降。

为了解决这一问题,文献[19]对文献[18]方案进行了扩充,提出了一种新的基于MapReduce的连接方案——PHiDJ(Parallel High-Dimensional Join),PHiDJ方案主要通过对维度分组和变长的网格划分来提高连接速度。具体思路为:首先把 $d$ 个维度(高维)划分成若干个互不相交的组(低维),每一个组包含若干个维;然后针对每一个组,再采用文献[18]方法处理,最后进行验证、过滤。另外,文献[18]采用的是均匀网格划分法(等宽),文献[19]则采用了变长的网格划分方案,具有更好的适应性和过滤效果。作者通过大量的实验结果表明,文献[19]算法比文献[18]算法具有更好的性能,并且能够处理更高维的向量。

文献[20-21]中提出了一种基于MapReduce框架的相



似性连接查询算法 MRSimJoin( MapReduce Similarity Join ),该算法的核心思想是从数据集中随机选取  $k$  个向量作为中枢,然后将每一个向量分配到距离最近的中枢所在的分区,这样可以形成  $k$  个基本分区。由于基本分区之间的向量仍有可能相似,因此,两个不同基本分区边界中的向量组成窗体对分区。如果某个分区的大小小于单个节点所能处理的最大向量数,那么该分区就不需要再进一步划分,可以直接对该分区执行相似性连接查询操作;反之,则需要对该分区作进一步划分。每一轮划分都由一个 MapReduce Job 来负责实现。

文献[22]对 MRSimJoin 算法<sup>[20]</sup>进行了改进,提出了一种 MapReduce 框架下增量式数据相似性 ( MapReduce-based Similarity Join for Incremental Data Set, MRSJ\_IDS ) 连接算法,该算法能够支持增量式数据集的相似性连接查询。MRSJ\_IDS 算法的基本思想是:首先通过抽样的方式获得一个样本集合,然后对样本集合进行聚类,将各个类的中心作为中枢,按照 MRSimJoin 算法中的方法对全部数据进行划分,从而得到基本分区和窗体对分区。针对每一个分区创建一个 KD-树索引,并计算出相似对。对于新增数据,首先根据事先确定的划分原则找到相应的分区,再根据该分区的 KD-树索引进行查询、插入操作,从而获得新增数据对应的相似对,并对原有的 KD-树索引进行更新。

FACET( FAst and sCalable maprEduce similariTy join )<sup>[23]</sup>以余弦相似度来衡量向量之间的相似度。由于文献[8~10]中提出的前缀过滤方法只适用于集合数据,作者针对向量数据和余弦相似度提出了新的前缀过滤机制和长度过滤机制,并结合 MapReduce 框架设计了并行实现算法 FACET<sup>[23]</sup>,可以快速、精确计算出所有的相似对。

SAX-Based HDSJ( Symbolic Aggregate approXimation based High-Dimensional Similarity Join )<sup>[24]</sup>采用 PAA 对高维向量进行降维,并在此基础上,利用符号累积近似法 ( Symbolic Aggregate Approximation, SAX ) 转换成 SAX 字符串,基于 SAX 可以对原始高维向量进行分组,由于 SAX 字符串之间的距离是向量之间原始距离的下界,因此,可以利用 SAX 进行有效过滤。SAX 的过滤效果直接影响到相似性连接查询的效率。为了进一步提高过滤效果和减小过滤代价,文献[25]在 SAX-Based HDSJ<sup>[24]</sup>的基础上提出了基于多 PAA 的相似性连接查询方案 MP-V-SJQ。

为了减少不必要的比较,并实现各个计算节点的负载均衡,文献[26]中提出了基于动态网格划分的相似性连接查询方案 Grid-Based SJ( Grid-Based Similarity Join ):首先通过采样的方法进行动态网格划分,并根据划分结果构造网格索引;然后依据网格索引对所有数据进行分配和计算,尽量确保各计算节点负载均衡。

SJT( Similarity Join Tree )<sup>[27]</sup>通过计算所有点与某个选定点之间的距离,将原始数据映射到一维空间内,并在一维空间内使用距离阈值进行等宽划分。如果某块内的数据个数超过一定阈值,可以对该块进行进一步划分。为了能够较好地描述上述数据划分思路,作者在 SJT<sup>[27]</sup> 中设计了相似性连接树。既可以减少不必要的比较计算,也可以实现各个计算节点的负载均衡。

K 最近邻 ( K-Nearest Neighbor, KNN ) 相似性连接查询:文献[28]最早对基于 MapReduce 框架的 KNN 连接查询问题进行了研究,为了减少比较次数,降低网络传输代价,文献[28]

中提出了一种基于空间填充曲线技术的近似 KNN 连接查询方案 zKNNJ( z-order based K-Nearest Neighbor Join )。

zKNNJ 的基本思路是:首先采用 z-order 方法将  $d$  维空间的点转换成一维数据,并按照 z-order 值进行排序;在此基础上, $d$  维空间的 KNN 连接查询可以转换成一维上的范围查询;为了提高查询结果的准确度,可以将原始向量转换成多个一维序列。基于上述思想,作者提出了基于 MapReduce 的并行计算方案 H-zKNNJ。H-zKNNJ 基本思路是:首先将集合  $R$  和  $S$  按照上述的转换方法,转换成两个一维的序列;接着将两个一维序列分别划分成  $n$  个组,划分的时候要求  $R$  和  $S$  具有相同的划分边界,并且要确保划分的均衡性,作者提出了一种基于采样技术的划分方案;将向量进行划分以后,对于  $R$  中的每一组  $R_i$ ,从  $S$  中找出可能满足 KNN 查询要求的集合  $S'_i$ (可能包含  $S$  中多个组的数据),最后将  $R_i$  和  $S'_i$  中的向量进行比较即可得到 KNN 的查询结果。但是 H-zKNNJ 也存在一些不足之处,如只能返回近似的结果、不能有效地处理较高维度的向量等。

与文献[28]返回近似的 KNN 连接查询结果不同,文献[29]中提出了一种精确的 KNN 连接查询方案,该方案主要基于维诺图 ( Voronoi Diagram ) 对数据进行划分。基本思路是:对于给定的两个向量集合  $R$  和  $S$ ,首先采用维诺图把集合  $R$  划分成  $k$  个互不相交的子集  $R_1, R_2, \dots, R_k$ ;然后针对  $R$  中的每一个子集  $R_i$ ,按照一定的方法,从集合  $S$  中找到一个对应的子集  $S_i$ , $S_i$  需要满足如下要求:  $\forall r \in R_i, KNN(r, S) \subseteq S_i$ ,  $S$  的各个子集之间可能会有重叠。按照上述划分方案,集合  $R$  中的每一个子集  $R_i$  只需要与对应的子集  $S_i$  进行比较,这样就可以大大降低网络传输的代价,同时也减少很多不必要的计算。上述方案的难点有两个:一是如何对集合  $R$  进行划分;二是如何为  $R$  的每一个子集  $R_i$  寻找相应的  $S_i$ 。在利用维诺图对集合  $R$  进行划分时,核心是中枢 ( pivot ) 的选择,作者分别提出了随机选择 ( random selection )、最远选择 ( farthest selection ) 和  $k$ -means 选择 ( $k$ -means selection) 三种不同的选择方案,并分别进行了实验验证。实验结果表明,该方案在处理中低维度向量时效果较好,但是无法有效处理超高维度向量。

文献[30]在嵌套循环连接框架的基础上,分别提出了精确 KNN ( Distributed Sketched Grid based KNN Join using MapReduce, DSGMP-J ) 连接算法和近似 KNN ( Voronoi Diagram based KNN Join using MapReduce, VDMP-J ) 连接算法。在 DSGMP-J 算法中,首先采用 DSG ( Distributed Sketched Grid ) 对空间进行划分,然后针对每一个单元格内的数据建立一个本地索引,这样就可以利用本地的 DSG 索引来求出局部 KNN 结果。DSGMP-J 方案虽然实现起来比较简单,但是在进行数据划分时,没有考虑数据的真实分布情况,只是采用了简单的均匀划分方案。为了解决这一问题,作者采用维诺图对数据进行划分,在此基础上,提出了一种近似的 KNN 查询方案 VDMP-J。

Top- $k$  相似性连接查询:文献[31]中提出了基于 MapReduce 框架的 Top- $k$  相似性连接查询方案。作者首先提出了两种串行化算法,分别是 divide-and-conquer 算法和 branch-and-bound 算法;然后又提出了两种基于 MapReduce 框架的并行化算法,分别是完全对划分算法 ( All pair Partitioning Algorithm, TopK-P-MR ) 和关键对划分算法 ( Essential Pair Partitioning Algorithm, TopK-F-MR )。TopK-P-MR 算法实际上



是一个嵌套循环连接算法,需要分两个阶段来实现,首先找出局部的 Top- $k$  结果,然后再找出全局的 Top- $k$  结果。在计算局部 Top- $k$  时,可以利用到前面提出的串行化算法。但是该方案的时间复杂度是  $O(n^2)$ , 即任意两个向量之间都需要计算一次,计算代价过大。为了解决这一问题,作者又提出了 TopK-F-MR 算法,该算法首先通过采样的方法,找出第  $k$  个最小的距离,并以该距离作为 Top- $k$  结果的距离上界,然后根据该上界对数据进行过滤和划分,这样就可以减少很多不必要的比较和计算,大大提高计算的效率。

针对大规模高维向量,文献[32]中提出了一种基于 MapReduce 框架的并行 Top- $k$  连接查询算法 SAX-Top- $k$ 。SAX-Top- $k$  算法首先提出了一个基于采样技术的 Top- $k$  阈值估计方法,并结合符号累积近似法(SAX)设计了高效的过滤方案,最后结合 MapReduce 框架提出了基于 SAX 的并行 Top- $k$  连接查询算法。

文献[33]中针对高维数据提出了基于局部敏感哈希(Locality-Sensitive Hashing, LSH)的距离,并将基于 LSH 的距离转换成高维数据签名的海明距离,在此基础上,设计了基于 Spark 的 Top- $k$  相似性连接查询方案。与基于 Hadoop 的方案相比,文献[33]方案的计算速度更快,具有更好的扩展性。

### 3 空间数据相似性连接查询

空间数据相似性连接是指给定两个空间数据集合  $R$  和  $S$ , 找出所有满足空间关系要求的空间数据对。其中,空间数据可以是点(兴趣点,如一栋房子、一个商铺、一个邮筒、一个公交站等)、线(街道)、多边形(住宅小区、医学图片中的细胞等),空间关系可以是欧氏距离、相交(重叠)等。根据返回结果的不同,又可以分为相交连接、Top- $k$  连接、KNN 连接和空间聚集连接等。

文献[34]中提出了一种基于 MapReduce 框架的空间数据连接算法 SJMR(Spatial Join with MapReduce)。该算法通过一个 MapReduce Job 来完成,在 Map 阶段,作者提出了一个空间划分函数,将空间对象划分到不同的子区域中,每一子区域中的数据由一个 Reduce 任务负责处理。为了确保划分的均匀性,作者提出了基于空间填充曲线的划分方法。在 Reduce 阶段,采用过滤-验证框架进行处理,在过滤阶段,为了提高过滤效果,作者提出了 Plane Sweeping 技术,从而可以减少候选对的数量。由于一个空间对象可能会被划分到不同的子区域中,所以在 Reduce 端可能会出现很多重复的比较,作者提出了一种基于参考点的方法,从而确保一对空间数据最多只被比较一次。

文献[35]中针对海量空间数据,提出了一种基于 MapReduce 框架的 Top- $k$  空间连接查询算法(Top- $k$  Spatial Join Algorithm using MapReduce, TKSJMR),该算法要求找出  $k$  个与其他空间数据具有最大交叠数的对象。主要通过三个阶段来完成,每一个阶段由一个 MapReduce Job 来实现。第一个阶段为空间连接阶段,主要负责找出所有相交的空间数据对;第二个阶段为连接结果统计阶段,负责统计出每个对象与其他对象相交的总数目;第三个阶段为 Top- $k$  结果获取阶段,主要是从上一阶段的统计结果中找出相交数最大的  $k$  个对象。为了进一步提高效率,作者进行了一些优化,如为了减少必要的比较,在空间连接阶段,作者采用了基于网格的划分方法;为了避免重复比较,又提出了基于参考点的方法。同时为

了减少 MapReduce Job 的数量,作者把后面两个阶段进行了合并。

文献[36]中提出了一种基于 MapReduce 框架的快速空间聚集连接查询算法 MRFM(Map-Reduce-Filter-Merge)。文献[37]首先提出了基于 MapReduce 框架的并行 R-tree 索引构建方法,并在 R-tree 索引的基础上提出了基于 MapReduce 的 KNN 连接查询算法。文献[38]中针对空间连接问题,提出了一种新的“可控-复制”框架,基于该框架可以减少集群节点之间的网络传输代价,能够有效处理基于“重叠”和“包含”两种谓词的空间连接查询问题。

文献[39]中针对 MapReduce 框架下的空间关键字连接查询问题进行了研究,提出了基于前缀过滤和网格化分技术相结合的空间文本对象过滤算法,并在此基础上又提出了两种优化方法,从而进一步提升了空间关键字连接查询的性能。

### 4 概率数据相似性连接查询

文献[40]中和 MELODY-JOIN<sup>[41]</sup>主要针对大规模概率数据,提出了基于 EMD(Earth Mover's Distance)距离的相似性连接查询算法。很多概率数据(如传感器搜集的温度、湿度数据、图像的 RGB 颜色分布数据等)都可以用直方图来表示,EMD 距离可以很好地度量直方图之间的相似性。但是 EMD 距离的计算具有很高的时间复杂度,所以传统的单机算法很难高效地处理大规模直方图概率数据的相似性连接查询问题。为此,文献[40]主要研究了如何在 MapReduce 框架下高效处理海量概率数据的 Top- $k$  连接查询问题。其核心思想是:基于 EMD 距离对偶问题的可行解,可以为每一个概率数据计算一个键值,从而将原始空间数据转换到一维空间上,这样针对原始概率数据的阈值查询就可以转换成一维键值空间上的范围查询。基于上述基本思想,作者首先提出了基于嵌套循环连接的 Top- $k$  相似性连接查询算法 Top- $k$  BNLJ(Top- $k$  Block Nested Loop Join)。该算法主要分为三个阶段,第一阶段通过采样方法得到 Top- $k$  相似性连接的阈值上界  $\tau$ ;第二个阶段以  $\tau$  为阈值上界,找出局部的 Top- $k$  结果;第三个阶段主要负责从所有的局部 Top- $k$  结果中找出全局的 Top- $k$  结果。Top- $k$  BNLJ 算法中,每个数据都会被复制  $m$  次( $m$  是数据块的个数),因此网络传输代价较高,为了解决这一问题,作者又提出了一种基于数据局部性的 Top- $k$  相似性连接查询算法 Top- $k$  DLPJ(Top- $k$  Data Locality Preserving Join)。Top- $k$  DLPJ 算法也由三个阶段完成:第一个阶段除了通过采样的方式计算出 Top- $k$  连接的阈值上界以外,同时还需要根据抽样数据找出在一维空间中的分位点,可以将原始数据近似划分成大小相等的子集。第二个阶段根据前面找到的分位点和阈值上界,为每一个子集  $R_i$  找出一个对应的子集,确保与  $R_i$  中的任何一个概率数据满足阈值  $\tau$  的所有数据都在子集中  $R'_i$ ,这样每一个子集  $R_i$  只需要与子集  $R'_i$  比较即可,从而可以大大降低数据传输的代价和比较的次数。第三个阶段主要负责从所有的局部 Top- $k$  结果中找出全局的 Top- $k$  结果。

MELODY-JOIN<sup>[41]</sup>在 MapReduce 框架下,针对直方图概率数据提出了一种基于 EMD 距离的阈值相似性连接方案。其核心思想是:首先将原始的直方图概率数据转换到 EMD 距离的标准下界空间,然后采用特定的网格划分方法对标准下界空间进行划分;接下来计算每一个记录与每一个单元格的 EMD 距离的下界,如果该下界值大于给定的阈值,则该记录



就不需要与该单元格内的所有记录进行比较;否则需要进一步比较。同时,为了解决负载均衡的问题,作者提出了基于 Quantile Grid 的划分方法,从而使得每一个单元格包含的记录数量近似相等。但是,MELODY-JOIN<sup>[41]</sup> 需要三个 MapReduce Job 来实现,需要消耗三次 Job 启动时间;另外,MELODY-JOIN<sup>[41]</sup> 的负载均衡策略仅仅依赖于连接负载(join workload),无法有效处理转换空间内的数据倾斜问题。Heads-Join<sup>[42]</sup> 通过引入 EMD 下界和上界对 MELODY-JOIN<sup>[41]</sup> 进行了进一步扩展,既可以处理范围连接(range join),又可以处理 Top-k 连接,并将 Heads-Join 框架在 MapReduce、BSP 和 Spark 模式下分别进行了实现。

为了解决 MELODY-JOIN<sup>[41]</sup> 存在的问题,文献[43]中提出了基于映射的数据划分框架 EMD-MPJ (Mapping-based Partition Join),它只需要两个 MapReduce Job 即可完成,其数据划分方案主要是基于线性规划的对偶理论进行设计。为了实现负载均衡,EMD-MPJ 设计了针对 Reduce 端的连接代价模型,并据此进行连接负载的分配。

文献[44]中针对大规模概率集合数据提出了两种基于 MapReduce 的并行化方法:基于 Map 端过滤的连接和基于 Reduce 端过滤的连接。基于 Map 端过滤的连接算法主要根据集合的存在概率,在 Map 端将那些没有可能与其他任何概率集合相似的集合直接过滤掉;基于 Reduce 端过滤的连接算法主要采用基于概率总和以及概率上界的过滤方法来减少候选相似对的数量,从而降低计算代价。

## 5 字符串相似性连接查询

林学民、李国良、王炜等学者对集中式环境下的字符串相似性连接查询问题进行了细致深入的研究,并提出了大量创新性的成果<sup>[45~50]</sup>。文献[51~53]主要探讨了基于 MapReduce 框架的海量字符串数据可扩展相似性连接查询问题。文献[51]以编辑距离作为字符串间的相似性度量,以 trie 树结构为基础,提出了一种新的索引结构 PeARL。PeARL 索引由一系列 trie 树构成,每一棵 trie 树用来索引以某个前缀(prefix)开头的字符串。在进行连接查询时,master 节点首先从根节点开始逐步扫描 trie 树,针对 trie 树的每一对内部节点,计算其对应前缀的编辑距离,如果大于给定的阈值,则对该对节点就可以过滤掉;否则继续往下扫描,直到遇到叶子节点对(字符串节点对)时,为其生成一个 map task。以此类推,为每一个可能相似的字符串节点对生成一个 map task,最后以并行的方式来计算字符串之间的实际编辑距离。

文献[52]主要对基于划分的签名机制进行了扩展,从而可以支持基于集合相似度(杰卡德相似度)的字符串连接。为了解决文献[8]中存在的过滤效果不理想、数据倾斜等问题,文献[52]中提出了一种基于划分的签名机制,将每个字符串按照某种规则划分成若干个分段,如果两个字符串相似,它们至少包含有一个相同的分段。基于该特性,就可以为每一个分段生成一个  $\langle key, value \rangle$  对,这样就可以取得较好的过滤效果。为了进一步减少  $\langle key, value \rangle$  对的数量,作者又提出了一种基于合并的算法,从而可以减少网络传输的代价。文献[53]对 PassJoin(Partition-based Method for Similarity Joins) 算法<sup>[54]</sup>进行了扩展,提出了一种更快的算法 PassJoinK,并结合 MapReduce 框架,对 PassJoinK 算法进行并行化,提出了可扩展的字符串相似性连接查询算法 PassJoinKMRS。

## 6 图数据相似性连接查询

文献[55]中针对海量图数据的相似性连接查询问题,提出了可扩展的前缀过滤方案,从而减少比较次数;设计了一种有效的候选项约减方法来降低数据传输代价;并基于 MapReduce 框架设计了可扩展的图数据相似性连接查询算法。文献[56]主要研究了基于编辑距离的图相似性连接查询问题,作者主要提出了一种基于“过滤-验证”机制的算法 MGSJoin(Graph Similarity Joins in MapReduce),采用布隆过滤器技术来减少冗余计算和网络传输代价,并集成多路连接策略来增强验证阶段的效率。文献[57]主要解决了海量 RDF 数据的连接查询问题。针对 RDF 数据的 SPARQL 查询往往包含很多连接操作,查询代价比较高,当数据规模比较大时,传统的单机算法无法满足性能要求,于是文献[57]中提出了基于 MapReduce 框架的高效 RDF 数据连接方案。首先将原始的 RDF 数据按照谓词(predicate)进行分解重组,每一个谓词对应一个谓词文件;然后在这些谓词文件的基础上,把 SPARQL 查询转换成一系列的 MapReduce Job,其中可能包含很多连接操作;作者提出了一种新颖的树形结构索引(all possible join tree)来索引所有可能的执行计划;最后依据代价模型来选择最优的查询计划,从而获得最快的响应时间。

## 7 研究工作展望

目前,在云计算环境下针对多种不同数据类型(集合数据、空间数据、概率数据等)的大数据相似性连接查询算法的研究已经取得了一些成果,但是仍然存在诸多挑战性问题值得进一步深入研究:

### 1) 大规模超高维数据相似性连接查询技术。

“高维”是大数据的一个重要特征,基因序列、轨迹数据、视频、音频、图片等非结构化数据都具有高维特性,高维大数据的有效分析和管理是大数据面临的一个重大挑战。高维向量相似性连接查询主要面临两大挑战:一是如何设计高效的过滤方案。当向量维度较高时,现有的以树形索引为基础的过滤方案会面临“维度灾难”问题。当向量维度逐渐增大时,索引的过滤效果逐渐降低,当维度超过一定阈值时,索引的性能甚至不如顺序扫描。二是如何设计高效的可扩展算法。随着向量维度的不断增高,两个向量之间相似度的计算代价会比较大;随着参与连接的向量集合规模的不断扩大,相似性连接的时间复杂度呈指数级增长。传统的集中式处理算法已经无法有效处理大规模超高维向量的相似性连接查询问题。

### 2) 复杂数据类型的大数据相似性连接查询技术。

已有的相似性连接查询研究工作主要集中在常见的集合数据、字符串数据和向量数据。然而,除了这三种数据类型之外,还有很多结构更为复杂的数据类型,如轨迹数据、时间序列、基因数据、流数据、图和 XML 文档等。这些数据的结构往往更为复杂,相似度的计算代价更高,并且,由于类型不同,已有的相似性连接查询技术并不能有效处理这些更为复杂的数据类型。因此,需要针对这些复杂数据类型的结构和特点,研究新的相似度计算方法、过滤技术和并行化方案。

### 3) 增量式大数据相似性连接查询技术。

目前的大数据相似性连接查询技术大都是基于 MapReduce 框架,然而,MapReduce 是一种批处理模型,无法有效处理实时查询和增量式查询。基于 Spark 等新的计算框



架的增量式大数据相似性连接查询技术,有待进一步深入研究。

#### 4) 基于哈希学习的近似 KNN 连接查询技术。

局部敏感哈希技术是解决海量高维向量 KNN 连接查询的一种有效方案。常用的局部敏感哈希函数虽然具有位置敏感性,但并不能有效保证哈希映射前后数据之间的相对位置关系,影响 KNN 连接查询结果的质量。将哈希学习思想与局部敏感哈希技术相结合,通过学习的方法来对局部敏感哈希函数进行学习,使得学习到的 LSH 函数能够最大限度地保持哈希映射前后数据之间的相对位置关系。在此基础上,结合 MapReduce 框架,研究高维向量并行近似 KNN 连接查询算法,有效应对扩展性问题。

#### 5) 面向隐私保护的大数据相似性连接查询技术。

隐私保护是大数据时代的一个重要研究课题。轨迹数据、基因数据、社交网络数据等都包含了大量的个人敏感信息,如何在确保相似度计算准确性和效率的同时,又能最大限度地保护隐私数据,成为一个亟待解决的研究课题。目前,面向隐私保护的大数据相似性连接查询技术研究工作还处于起步阶段,需要进一步深入研究,例如,可以尝试将差分隐私等最新的隐私保护技术应用到大数据相似性连接查中。

## 8 结语

相似性连接查询是一种十分重要的操作,在很多数据挖掘和数据分析任务中都有应用。随着数据规模的不断增长,针对大数据的相似性连接查询问题出现了新的挑战。本文针对集合、向量、空间数据、概率数据等不同类型大数据的相似性连接查询技术相关工作进行了深入研究,对其优缺点进行了归纳总结,最后指出了大数据相似性连接查询面临的若干挑战性问题。

## 参考文献(References)

- [1] 庞俊,谷峪,许嘉,等.相似性连接查询技术研究进展[J].计算机科学与探索,2013,7(1): 1–13.(PANG J, GU Y, XU J, et al. Research advance on similarity join queries[J]. Journal of Frontiers of Computer Science & Technology, 2013, 7(1): 1–13.)
- [2] 林学民,王炜.集合和字符串的相似度查询[J].计算机学报,2011,34(10): 1853–1862.(LIN X M, WANG W. Set and string similarity queries: a survey [J]. Chinese Journal of Computers, 2011, 34(10): 1853–1862.)
- [3] YU M H, LI G L, DENG D, et al. String similarity search and join: a survey[J]. Frontiers of Computer Science, 2016, 10(3): 399–417.
- [4] 庞俊,于戈,许嘉,等.基于 MapReduce 框架的海量数据相似性连接研究进展[J].计算机科学,2015,42(1): 1–5.(PANG J, YU G, XU J, et al. Similarity joins on massive data based on MapReduce framework[J]. Computer Science, 2015, 42(1): 1–5.)
- [5] SILVA Y, REED J, BROWN K, et al. An experimental survey of MapReduce-based similarity joins[C]// Proceedings of the 9th International Conference on Similarity Search and Applications. Berlin: Springer, 2016: 181–195.
- [6] KIMMETT B, SRINIVASAN V, THOMO A. Fuzzy joins in MapReduce: an experimental study[J]. Proceedings of the VLDB Endowment, 2015, 8(12): 1514–1517.
- [7] LIN J. Brute force and indexed approaches to pairwise document similarity comparisons with MapReduce[C]// Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM, 2009: 155–162.
- [8] VERNICA R, CAREY M J, LI C. Efficient parallel set-similarity joins using MapReduce[C]// Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2010: 495–506.
- [9] 李瑞,王朝坤,郑伟,等.基于 MapReduce 框架的近似复制文本检测[J].计算机研究与发展,2010,47(增刊1): 400–406.(LI R, WANG C K, ZHENG W, et al. Near duplicate text detection based on MapReduce[J]. Journal of Computer Research and Development, 2010, 47(S1): 400–406.)
- [10] RONG C T, LU W, WANG X, et al. Efficient and scalable processing of string similarity join[J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 25(10): 2217–2230.
- [11] ELSAYED T, LIN J, OARD D. Pairwise document similarity in large collections with MapReduce[C]// HLT-Short 2008: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies. Stroudsburg, PA, USA: ACL, 2008: 265–268.
- [12] METWALLY A, FALOUTSOS C. V - SMART - Join : a scalable MapReduce framework for all-pair similarity joins of multisets and vectors[J]. Proceedings of the VLDB Endowment, 2012, 5(8): 704–715.
- [13] BARAGLIA R, MORALES G, LUCCHESE C. Document similarity self-join with MapReduce[C]// Proceedings of the 10th IEEE International Conference on Data Mining. Piscataway, NJ: IEEE, 2010: 731–736.
- [14] RONG C T, LIN C B, SILVA Y, et al. Fast and scalable distributed set similarity joins for big data analytics[C]// Proceedings of the 2017 IEEE 33rd International Conference on Data Engineering. Piscataway, NJ: IEEE, 2017: 1–12.
- [15] DENG D, LI G L, WEN H, et al. An efficient partition based method for exact set similarity joins[J]. Proceedings of the VLDB Endowment, 2015, 9(4): 360–371.
- [16] WANG J J, LIN C. MapReduce based personalized locality sensitive hashing for similarity joins on large scale data[J]. Computational Intelligence and Neuroscience, 2015, 2015: Article No. 37.
- [17] LUO W, TAN H, MAO H, et al. Efficient similarity joins on massive high-dimensional datasets using MapReduce[C]// Proceedings of the 13th IEEE International Conference on Mobile Data Management. Piscataway, NJ: IEEE, 2012: 1–10.
- [18] SEIDL T, FRIES S, BODEN B. MR-DSJ: distance-based self-join for large-scale vector data analysis with MapReduce[C]// Proceedings of the 15th BTW Conference on Database Systems for Business, Technology, and Web. Berlin: Springer, 2013: 37–56.
- [19] FRIES S, BODEN B, STEPIEN G, et al. PHIDJ: parallel similarity self-join for high-dimensional vector data with MapReduce [C]// Proceedings of the 30th IEEE International Conference on Data Engineering. Piscataway, NJ: IEEE, 2014: 796–807.
- [20] SILVA Y N, REED J M, TSOSIE L M. MapReduce-based similarity join for metric spaces[C]// Proceedings of the 1st International Workshop on Cloud Intelligence. New York: ACM, 2012: Article No. 3.
- [21] SILVA Y N, REED J M. Exploiting MapReduce-based similarity joins[C]// Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2012: 693–696.
- [22] 徐媛媛,陈华辉.基于 MapReduce 增量式数据集的相似性连接[J].计算机应用研究,2014,31(11): 3369–3384.(XU Y



- Y, CHEN H H. MapReduce-based similarity join for incremental data set[J]. Application Research of Computers, 2014, 31(11): 3369 – 3384.)
- [23] YANG B, KIM H, SHIM J, et al. Fast and scalable vector similarity joins with MapReduce[J]. Journal of Intelligent Information Systems, 2016, 46(3): 473 – 497.
- [24] MA Y Z, MENG X F, WANG S Y. Parallel similarity joins on massive high-dimensional data using MapReduce[J]. Concurrency and Computation: Practice and Experience, 2016, 28(1): 166 – 183.
- [25] MA Y Z, JIA S J, ZHANG Y X. A novel approach for high-dimensional vector similarity join query[J]. Concurrency and Computation: Practice and Experience, 2017, 29(5): 1 – 12.
- [26] JANG M Y, SONG Y, CHANG J. A density-aware similarity join query processing algorithm on MapReduce[M]// PARK J J, JIN H, KHAN M K, et al. Advanced Multimedia and Ubiquitous Engineering. Berlin: Springer, 2016: 469 – 475.
- [27] LIU W, SHEN Y M, WANG P. An efficient MapReduce algorithm for similarity join in metric spaces[J]. The Journal of Supercomputing, 2016, 72(3): 1179 – 1200.
- [28] ZHANG C, LI F, JESTES J. Efficient parallel kNN joins for large data in MapReduce[C]// Proceedings of the 15th International Conference on Extending Database Technology. New York: ACM, 2012: 38 – 49.
- [29] LU W, SHEN Y, CHEN S, et al. Efficient processing of k nearest neighbor joins using MapReduce [J]. Proceedings of the VLDB Endowment, 2012, 5(10): 1016 – 1027.
- [30] 戴健, 丁治明. 基于 MapReduce 快速 kNN Join 方法[J]. 计算机学报, 2015, 38(1): 99 – 108. ( DAI J, DING Z M. MapReduce based fast kNN join [J]. Chinese Journal of Computers, 2015, 38(1): 99 – 108.)
- [31] KIM Y, SHIM K. Parallel Top-K similarity join algorithms using MapReduce[C]// Proceedings of the 2012 IEEE 28th International Conference on Data Engineering. Washington, DC: IEEE Computer Society, 2012, 510 – 521.
- [32] 马友忠, 慈祥. 海量高维向量的并行 Top-k 连接查询[J]. 计算机学报, 2015, 38(1): 86 – 98. ( MA Y Z, CI X. Parallel Top-k join on massive high-dimensional vectors[J]. Chinese Journal of Computers, 2015, 38(1): 86 – 98.)
- [33] CHEN D H, SHEN C G, FENG J Y, et al. An efficient parallel Top-k similarity join for massive multidimensional data using spark [J]. International Journal of Database Theory and Application, 2015, 8(3): 57 – 68.
- [34] ZHANG S B, HAN J Z, LIU Z Y, et al. SJMR: parallelizing spatial join with MapReduce on clusters[C]// Proceedings of 2009 IEEE International Conference on Cluster Computing and Workshops. Piscataway, NJ: IEEE, 2009: 1 – 8.
- [35] 刘义, 陈萍, 景宁, 等. 海量空间数据的并行 Top-k 连接查询 [J]. 计算机研究与发展, 2011, 48(增刊3): 163 – 172. ( LIU Y, CHEN L, JING N, et al. Parallel Top-k spatial join query processing on massive spatial data[J]. Journal of Computer Research and Development, 2011, 48(S3): 163 – 172.)
- [36] LIU Y, CHEN L, JING N, et al. MRFM: an efficient approach to spatial join aggregate[C]// Proceedings of the WAIM 2012 International Workshops: GDMM, IWSN, MDSP, USDM, and XM-LDM. Berlin: Springer, 2012, 140 – 150.
- [37] 刘义, 景宁, 陈萍, 等. MapReduce 框架下基于 R-树的 k-近邻连接算法[J]. 软件学报, 2013, 24(8): 1836 – 1851. ( LIU Y, JING N, CHEN L, et al. Algorithm for processing k-nearest join based on R-tree in MapReduce[J]. Journal of Software, 2013, 24(8): 1836 – 1851.)
- [38] GUPTA H, CHAWDA B, NEGI S, et al. Processing multi-way spatial joins on Map-Reduce[C]// Proceedings of the 16th International Conference on Extending Database Technology. New York: ACM, 2013, 113 – 124.
- [39] ZHANG Y, MA Y, MENG X. Efficient spatio-textual similarity join using MapReduce [C]// Proceedings of the 2014 IEEE/WIC/ ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies. Piscataway, NJ: IEEE, 2014: 52 – 59.
- [40] 雷斌, 许嘉, 谷峪, 等. 概率数据上基于 EMD 距离的并行 Top-k 相似性连接算法[J]. 软件学报, 2013, 24(增刊2): 188 – 199. ( LEI B, XU J, GU Y, et al. Parallel Top-k similarity join algorithm on large probabilistic data based on earth mover's distance [J]. Journal of Software, 2013, 24(S2): 188 – 199.)
- [41] HUANG J, ZHANG R, BUYYA R, et al. MELODY-JOIN: efficient earth mover's distance similarity joins using MapReduce[C]// Proceedings of the 30th IEEE International Conference on Data Engineering. Piscataway, NJ: IEEE, 2014: 808 – 819.
- [42] HUANG J, ZHANG R, BUYYA R, et al. Heads-Join: efficient earth mover's distance similarity joins on Hadoop[J]. IEEE Transactions on Parallel and Distributed Systems, 2016, 27(6): 1660 – 1673.
- [43] XU J, LEI B, GU Y, et al. Efficient similarity join based on earth mover's distance using MapReduce [J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(8): 2148 – 2162.
- [44] MA Y Z, MENG X F. Set similarity join on massive probabilistic data using MapReduce[J]. Distributed and Parallel Databases, 2014, 32(3): 447 – 464.
- [45] WANG J N, LI G L, FENG J H. Extending string similarity join to tolerant fuzzy token matching[J]. ACM Transactions on Database Systems, 2014, 39(1) : Article No. 7.
- [46] LI G L, DENG D, FENG J H. Pass-Join +: a partition-based method for string similarity joins with edit-distance constraints[J]. ACM Transactions on Database Systems, 2013, 38(2) : Article No. 9.
- [47] JIANG Y, LI G, FENG J H, et al. String similarity joins: an experimental evaluation[J]. Proceedings of the VLDB Endowment, 2014, 7(8): 625 – 636.
- [48] WANG W, QIN J B, XIAO C, et al. VChunkJoin: an efficient algorithm for edit similarity joins[J]. IEEE Transactions on Knowledge & Data Engineering, 2013, 25(8): 1916 – 1929.
- [49] LU J H, LIN C B, WANG W, et al. String similarity measures and joins with synonyms[C]// SIGMOD 2013: Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2013: 373 – 384.
- [50] XIAO C, WANG W, LIN X M, et al. Efficient similarity joins for near duplicate detection[J]. ACM Transaction of Database Systems, 2011, 36(3) : Article No. 15.
- [51] RHEINLÄNDER A , LESER U . Scalable sequence similarity search and join in main memory on multi-cores[C]// Euro-Par 2011: Proceedings of the 2011 International Conference on Parallel Processing. Berlin: Springer, 2011, 2: 13 – 22.
- [52] DENG D, LI G L, HAO S, et al. MassJoin: a MapReduce-based algorithm for string similarity joins[C]// Proceedings of IEEE 30th International Conference on Data Engineering. Piscataway, NJ: IEEE, 2014: 340 – 351.

(下转第 1006 页)



- 法[J]. 计算机应用研究, 2018, 35(1): 105–112. (XIAO W Q, YAO S J, WU S M. Improved Top- $N$  collaborative filtering recommendation algorithm[J]. Application Research of Computers, 2018, 35(1): 105–112.)
- [5] 许海玲, 吴潇, 李晓东, 等. 互联网推荐系统比较研究[J]. 软件学报, 2009, 20(2): 350–362. (XU H L, WU X, LI X D, et al. Comparison study of Internet recommendation system[J]. Journal of Software, 2009, 20(2): 350–362.)
- [6] JEONG B, LEE J, CHO H. Improving memory-based collaborative filtering via similarity updating and prediction modulation[J]. Information Sciences, 2010, 180(5): 602–612.
- [7] ZHAO C, PENG Q, LIU C. An improved structural equivalence weighted similarity for recommender systems[J]. Procedia Engineering, 2011, 15: 1869–1873.
- [8] 李克潮, 蓝冬梅. 一种属性和评分的协同过滤混合推荐算法[J]. 计算机技术与发展, 2013, 23(7): 116–119. (LI K C, LAN D M. A collaborative filtering hybrid recommendation algorithm for attribute and rating [J]. Computer Technology and Development, 2013, 23(7): 116–119.)
- [9] VOZALIS M G, MARGARITIS K G. Using SVD and demographic data for the enhancement of generalized collaborative filtering[J]. Information Sciences, 2007, 177(15): 3017–3037.
- [10] 杨阳, 向阳, 熊磊. 基于矩阵分解与用户近邻模型的协同过滤推荐算法[J]. 计算机应用, 2012, 32(2): 395–398. (YANG Y, XIANG Y, XIONG L. Collaborative filtering and recommendation algorithm based on matrix factorization and user nearest neighbor model [J]. Journal of Computer Applications, 2012, 32(2): 395–398.)
- [11] CHEN G, WANG F, ZHANG C. Collaborative filtering using orthogonal nonnegative matrix tri-factorization[J]. Information Processing & Management, 2009, 45(3): 368–379.
- [12] 郁雪, 李敏强. 一种结合有效降维和 K-means 聚类的协同过滤推荐模型[J]. 计算机应用研究, 2009, 26(10): 3718–3720. (YU X, LI M Q. Collaborative filtering recommendation model based on effective dimension reduction and K-means clustering[J]. Application Research of Computers, 2009, 26(10): 3718–3720.)
- [13] 张林, 王晓东, 姚宇. 基于项目聚类和时间因素改进的推荐算法[J]. 计算机应用, 2016, 36(增刊2): 235–238. (ZHANG L,
- WANG X D, YAO Y. Improved recommendation algorithm based on item clustering and time factor [J]. Journal of Computer Applications, 2016, 36(S2): 235–238.)
- [14] TSI C F, HUNG C. Cluster ensembles in collaborative filtering recommendation[J]. Applied Soft Computing, 2012, 12(4): 1417–1425.
- [15] 李振博, 徐桂琼, 查九. 基于用户谱聚类的协同过滤推荐算法[J]. 计算机技术与发展, 2014, 24(9): 59–67. (LI Z B, XU G Q, ZHA J. A collaborative filtering recommendation algorithm based on user spectral clustering [J]. Computer Technology and Development, 2014, 24(9): 59–67.)
- [16] XU D, TIAN Y. A comprehensive survey of clustering algorithms [J]. Annals of Data Science, 2015, 2(2): 165–193.
- [17] SEHGAL G, GRAG D K. Comparison of various clustering algorithms[J]. International Journal of Computer Science and Information Technology, 2014, 5(3): 3074–3076.
- [18] XUE G R, LIN CH X, YANG Q, et al. Scalable collaborative filtering using cluster-based smoothing[C]// Proceeding of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM, 2005: 114–121.
- [19] MARY S S, SELVI R T. A study of  $K$ -means and cure clustering algorithms[J]. International Journal of Engineering Research and Technology, 2014, 3(2): 1985–1987.
- [20] 奉国和, 梁晓婷. 协同过滤推荐研究综述[J]. 图书情报工作, 2011, 55(16): 126–130. (FENG G H, LIANG X T. A summary of collaborative filtering recommendations[J]. Library and Information Service, 2011, 55(16): 126–130.)

This work is partially supported by the National Natural Science Foundation of China (61404069), the Liaoning Provincial Department of Education Scientific Research Project (LJYL048).

**WANG Yonggui**, born in 1967, M. S., professor. His research interests include large data, database, data warehouse.

**SONG Zhenzhen**, born in 1989, M. S. candidate. Her research interests include recommendation algorithm, data mining.

**XIAO Chenglong**, born in 1984, Ph. D., associate professor. His research interests include hardware and software co-design, embedded systems, high-level synthesis.

the 24th International Conference on Scientific and Statistical Database Management. Berlin: Springer, 2012: 250–259.

This work is partially supported by the National Natural Science Foundation of China (61602231), the National Key R&D Plan Project (2016YFE0104600), the Science and Technology Open Cooperation Project of Henan Province (172106000077, 152106000048), the Key Scientific Research Project of Higher Education of Henan Province (16A520022).

**MA Youzhong**, born in 1981, Ph. D., associate professor. His research interests include big data, Web data management.

**ZHANG Zhihui**, born in 1979, M. S., lecturer. His research interests include data mining.

**LIN Chunjie**, born in 1981, M. S., lecturer. His research interests include data mining, rough set.

(上接第 986 页)

- [53] LIN C, YU H Y, WENG W, et al. Large scale similarity join with edit-distance constraints[C]// Proceedings of 19th International Conference on Database Systems for Advanced Applications. Berlin: Springer, 2014: 328–342.
- [54] LI G L, DENG D, WANG J N, et al. Pass-join: a partition-based method for similarity joins[J]. Proceedings of the VLDB Endowment. Berlin: Springer, 2011, 5(3): 253–264.
- [55] PANG J, GU Y, XU J, et al. Efficient graph similarity join with scalable prefix-filtering using MapReduce[C]// Proceedings of 15th International Conference on Web-Age Information Management. Berlin: Springer, 2014: 415–418.
- [56] CHEN Y F, ZHAO X, GE B, et al. Practising scalable graph similarity joins in MapReduce[C]// Proceedings of the 2014 IEEE International Congress on Big Data. Washington, DC: IEEE Computer Society, 2014: 112–119.
- [57] ZHANG X F, CHEN L, WANG M. Towards efficient join processing over large RDF graph using MapReduce[C]// Proceedings of