众包数据库综述

柴成亮[1)] 李国良[1)] 赵天宇[1)] 骆昱宇[1)] 于明鹤[2)]

[1)](清华大学计算机系 北京 100084)

[2)](东北大学软件学院 沈阳 110167)

摘 要 现如今,很多数据处理与分析的任务仅仅依靠机器算法难以达到理想的效果.因此,众包技术应运而生,其利用群体的智慧来解决对于计算机而言比较难的问题.其中,众包平台(例如 Amazon Mechanical Turk)为众包技术的应用提供了有力的支撑.平台上有成千上万的网络大众来为任务发布者解决问题.然而,对于任务发布者而言,其与众包平台交互是不方便的,因为平台会要求任务发布者设置很多参数甚至书写代码.所以研究者们借鉴传统数据库的思想,提出了众包数据库的概念,其封装了任务发布者、众包平台以及众包工人之间的复杂交互过程,为发布者提供友好的 API.使发布者可以通过简单的类 SQL 语言与平台交互.在这篇综述中,我们首先介绍众包的概念;然后介绍设计众包数据库时需考虑的一些基本技术,例如真值推理、任务分配,代价优化等;接着我们介绍几种主流的众包数据库系统.此外,我们会介绍对于不同的数据库算子,包括选择、连接、排序等优化技术.最后我们会介绍该领域未来的研究方向与挑战.

关键词 数据库;众包;查询优化;质量控制;成本控制

中图法分类号 TP391 **DOI 号** 10.11897/SP.J.1016.2020.00948

Crowd-Powered Database System: A Survey

CHAI Cheng-Liang[1)] Li Guo-Liang[1)] ZHAO Tian-Yu[1)] LUO Yu-Yu[1)] YU Ming-He[2)]

[1)](*Department of Computer Science, Tsinghua University, Beijing* 100084)

[2)](*Software College of Northeastern University, Shenyang* 110167)

Abstract Nowadays, many data management tasks cannot purely rely on machine-based algorithms to be resolved. Therefore, crowdsourcing has attracted the interest of many researchers, which leverages the crowd ability to address the problems that are hard for the computer. Thanks to crowdsourcing platforms, e.g., Amazon Mechanical Turk, we can easily hire hundreds of thousands of workers to resolve these computer-hard tasks. The technical difficulty of crowdsourcing is the complexity of interactions among the above three components, which makes the requesters hard to use and manage their tasks. For example, it is inconvenient for the requester to interact with the crowdsourcing platforms, which require the requesters to set parameters and write codes to display the tasks. Inspired by traditional DBMS, crowdsourcing database systems have been proposed to encapsulate the complexities of interacting with the crowd. The challenges include how to easily use crowdsourcing platforms, how to design query optimization models to optimize crowdsourcing costs, quality and latency and how to support complex crowdsourcing operations. In this paper, we will survey a wide spectrum of existing studies on crowdsourcing database systems. We first give an overview of crowdsourcing, and then introduce the fundamental techniques in designing crowdsourcing databases, including truth inference, task assignment, cost control, etc. In this part, we focus on reviewing sophisticated techniques on improving quality, reducing cost and reducing latency. Next,

收稿日期:2019-07-28;在线出版日期:2020-02-19.本课题得到国家自然科学基金(61632016,61925205)、国家"九七三"重点基础研究发展计划项目基金(2015CB358700)资助.**柴成亮**,博士研究生,主要研究方向为数据清洗与融合、众包数据管理.E-mail:chaicl15@mails.tsinghua.edu.cn.**李国良**(通信作者),博士,教授,中国计算机学会(CCF)高级会员,主要研究领域为数据清洗与融合、众包数据管理.E-mail:liguoliang@tsinghua.edu.cn.**赵天宇**,博士研究生,主要研究方向为时空数据库、众包计算.**骆昱宇**,硕士研究生,主要研究方向为数据清洗与融合、数据可视化.**于明鹤**,博士,主要研究方向为信息检索和大数据.

we will illustrate several popular crowd-powered database systems, including Deco, Qurk, CrowdDB and CDB. We mainly discuss the query language, query optimization models and supporting operations in these databases. Moreover, we review techniques on designing different operators, including selection, join, sort, etc. In this part, we mainly focus on how to optimize the cost, quality and latency for these operators. Finally, we discuss the future works and challenges.

Keywords　database; crowd-powered; query optimization; quality control; cost optimization

1　引　言

现如今，对于许多数据处理与分析的任务（例如实体匹配[1]），仅仅依靠机器算法不能达到很好的效果. 幸运的是，众包（crowdsourcing）模式的出现在很大程度上解决了这一问题. 众包概念最初是由美国《连线》的记者 JeffHoew 在 2006 年提出的，其定义是："一个公司或机构把过去由员工执行的工作任务，以自由自愿的形式外包给非特定的（而且通常是大型的）网络大众的做法[2]". 也就是众包利用人的智慧来帮助我们解决大型的、对于机器来说很难的问题. 在过去的十年中，众包模式已经不知不觉惠及到几乎每一个人并成为数据挖掘和数据管理的一大研究热点[3-38]. 一些影响力极其深远的工程，例如维基百科，百度知道，百度百科还有学术界的 ImageNet，都是通过众包模式来构建的. 步入大数据时代，数据量在不断增大，数据的表现形式也越来越多样化，导致众包的需求量增大以及任务的形式增多，在线众包平台应运而生，例如亚马逊的 Amazon Mechanical Turk（AMT）①、Crowdflower② 和 Upwork③. 在这些平台上，我们可以轻松地雇佣千上万的工人（Worker）来帮助我们解决问题. 解决问题的流程可以概括为以下三步：（1）任务发布者（Requester）发布许多任务到众包平台；（2）众包平台把任务分配给工人；（3）工人回答问题并返回答案给任务发布者.

然而，对于任务发布者来说与众包平台交互是比较不方便的，因为众包平台会要求任务发布者设置很多参数甚至书写代码来布置任务. 为了解决这一问题，受传统数据库管理系统的思想启发，研究人员们提出了众包数据库例如 CrowdDB[17]，Qurk[31]，Deco[37] 和 CDB[24]. 一方面，这些数据库提供了描述性的编程接口，其允许任务发布者利用简单的类 SQL 语言来发布任务并与众包平台交互. 另一方面，众包数据库封装了平台与工人交互的复杂流程. 针对不同的数据库算子，众包数据库会自主选择不同的策略来分配任务并推理答案. 这样大大提升了答案质量，降低了众包开销，也使得任务发布者不需要亲自设计相关策略. 在这种设计思想下，当任务发布者给出一个类 SQL 查询，众包数据库首先解析这个查询，生成带有众包算子的查询计划. 接着根据计划生成任务发布到众包平台上，最后收集众包工人的答案并产生最终的结果.

众包数据库与传统数据库的不同主要有以下两点. 第一，传统数据库采用的是封闭环境（Closed-World）的模型. 该模型仅能处理数据库中已经存在的数据. 然而众包数据库采用的是开放环境[17]（OpenWorld），该模型可以利用人工来获取外部世界的数据，加载到数据库后再进行处理. 例如，收集一条新的记录或者填充一个属性. 第二，众包数据库可以利用人的智慧来处理数据库相关算子. 例如，比较两个实体是不是指代同一实体，记录之间的排序，为实体打分等等. 这两个不同是我们设计众包数据库的主要动机. 接着我们将从以下三个方面来介绍众包数据库.

（1）众包概述

当任务发布者有一个任务需要利用众包处理时，他首先需要将该任务分解为很多微任务. 接着他需要设计匹配这些微任务的题目形式（网页前端技术）. 随后设置很多参数，包括但不限于为任务定价、设定答题时间、设定答题工人质量等. 之后便可以在众包平台发布任务，平台会将任务分配给工人. 当一个问题被工人回答，任务发布者可以选择接受或拒绝这个答案. 如果接受了该答案，他会付给工人相应的报酬.

（2）众包数据库相关技术

那么与传统数据处理算法相比，众包数据管理有什么挑战呢？① 有偿. 任务发布者需要付钱给工人，所以控制成本很重要. 我们可以利用人机混合（Human-machine Hybrid）以及答案推理（Answer

① https://www.mturk.com/
② http://www.crowdflower.com
③ https://www.upwork.com

《计算机学报》

Reasoning)的方法来降低成本.人机混合指得是首先利用机器算法对一些简单的问题进行判断,然后利用人来处理剩余的问题.答案推理指的是利用已知答案推理出其它问题的答案,例如在实体识别的问题中,我们已知"IPhone 6=IPhone six","IPhone 6≠IPad 2",那么我们可以推理出"IPhone six≠IPad2";② 易错.人是可能会返回错误答案的,那么如何在容忍错误答案的同时推出高质量的结果是一个重大挑战.我们可以利用真值推理(Truth Inference)以及任务分配(Task Assignment)的方法来降低成本.真值推理指的是为了容忍错误,我们可以把一个任务分给多个人,然后整合他们的答案并推导出正确答案.任务分配指的是我们给每个工人分配最适合他的任务,从而这些工人可以避免因为分配到不适合的任务而给出错误答案;③ 延迟.相对于计算机的计算能力来说,人给出答案的速度是较慢的,那么如何降低众包数据处理的延迟也是一大研究挑战.我们可以利用并行提问(Answering in Parallel)的方式来降低延迟,即每次提问多个任务.

(3)众包数据库系统以及相关算子

利用以上技术,研究人员设计了许多众包数据库系统,例如 CrowdDB,Qurk,Deco 和 CDB.为了提升数据库的性能,这些数据库系统重点关注于优化成本,提升质量以及降低延迟.另外,对于不同的算子包括选择、连接、排序、最大(小)值、计数、收集,这些数据库会设计相应的优化策略去优化.

目前已有很多有关于众包的总结性工作[39-45].VLDB'16[39]主要总结任务分配和执行任务过程中人的因素;VLDB'15[39]重点关注质量控制中的真值推理;ICDE'15[40]主要总结了众包的一部分算子,众包数据挖掘的内容.VLDB'12[41]讲述现在流行的众包平台以及它们的设计原则.文献[42]则从工业界的角度来探讨众包数据管理.文献[43]主要关注众包方面的基本技术如质量、成本和延迟.本文则重点关注众包数据库系统以及系统设计相关的众包算子.

本文在第 2 节介绍众包的基本概念,包括众包的定义及其基本流程;第 3 节介绍众包相关技术,包括质量控制、延迟控制和代价控制,这些技术在众包数据库的设计中同样需要考虑;第 4 节中介绍众包数据库的概念,并具体介绍了四种当下比较流行的众包数据库系统;第 5 节介绍众包数据库的算子,包括连接、选择、排序、收集等;第 6 节对未来工作进行展望.

2 众包概述

2.1 众包基本概念

众包是一种新兴的商业模式,学术界和工业界关于其定义一直没有统一的结论.文献[44]总结了约 40 种众包定义.综合考虑其内容,我们给出如下定义.

定义 1(众包). 众包通过整合互联网上的大众来处理对计算机来说很难的任务,是一种公开面向互联网大众的分布式的问题解决机制.

如图 1,我们可以根据两个维度对众包进行分类,任务粒度和激励机制.

图 1 众包任务的分类

按照任务粒度来分,可以分为微观任务(micro-task)和宏观任务(macro-task).微观任务(例如标记一张图片)指的是需要单个工人花费至多几分钟就能完成的任务.而宏观任务(例如翻译一篇文章)指的是单个工人需要花费数小时乃至几天才能完成的任务.在学术研究领域,研究者们主要关注于能被分解为很多微观任务的大任务,所以在本综述中,我们也主要关注微观任务.但是宏观任务同样是一个在未来很有挑战的研究问题.

按照工人的激励机制来分,众包任务可以分为以下三类:(1)任务发布者需要用金钱来激励工人答题.例如,ImageNet 利用众包构建了一个图片标注数据集.它首先构建了一个概念层次结构,如动物、狗、哈士奇,然后让人去标注某图片属于该层次结构上的哪一类;(2)工人出于兴趣自愿来答题.例如网络大众贡献维基百科词条、ESP 游戏用利用玩家来识别图片中的物体;(3)任务是隐藏的,工人们不知道他们在回答问题.例如,reCAPCHA 利用众包来数字化两千万张模糊的报纸;谷歌利用用户的点击来提升搜索引擎的性能.在本综述中,我们主要

关注需要任务发布者付费的任务，即第一类任务.

2.2　众包的工作流程

在众包平台中，任务发布者发布任务，工人们在众包平台上选择感兴趣的问题来回答然后返回答案.下面，文章将分别从任务发布者、众包平台和工人的角度来介绍众包流程.

2.2.1　任务发布者角度的众包流程

假设一个任务发布者有一个实体解析问题，目的是从 1000 个产品中找到相同的实体.(1)那么请求者首先使用众包平台提供的模板设计任务，调用 API 或编写代码与众包平台交互.例如，如果发布者使用模板，他可以设计任务的用户界面(例如，显示一对产品，并要求工人在“相同实体”和“不同实体”之间进行选择)；(2)发布者需要设置任务的一些属性，例如，单个任务的价格、回答单个任务的工人数、回答任务的最长或最短持续时间；(3)发布者将任务发布到平台，并从平台收集答案.任务发布者是整个众包流程中任务最繁琐，要求技术难度最高的角色，也是众包数据库系统重点优化的对象.图 2 概括了整个众包流程，接下来我们讨论以上每个步骤的细节.

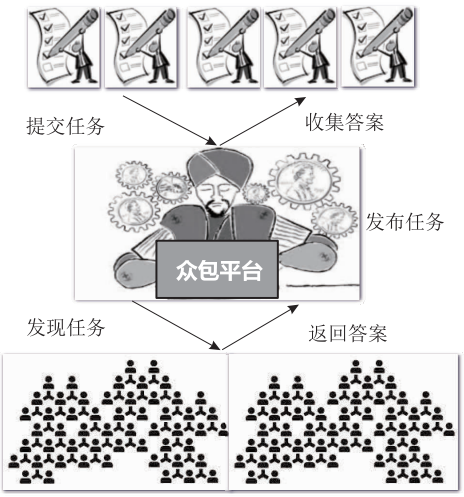

图 2　众包工作流程

2.2.1.1　任务设计

现实中有很多不同种类的任务.对于不同的任务，需要设计不同的交互界面来支持，例如：

(1)单选题.单选任务要求工人从多个选项中选择一个答案.例如，在实体解析中，给定一对实体，它要求工人选择它们是否指代同一实体(选项：是或否).在手机图片分类中，给出一个手机图片，它要求工人选择它的分类(选项：苹果、三星、谷歌等).

(2)多选题.多选任务要求工人从多个选项中选择多个答案.例如，对于给定两张手机图片，要求工人它们是否具有相同的品牌、颜色、相似的价格和大小.

(3)填空题.填空任务要求工人填写一些记录中指定属性的值.例如，对于一位 NBA 球星，任务要求工人提供该球星的身高体重.因此，工人提供的值是不限制类型的，可以是数字、文本或类别.

(4)收集题.收集任务要求工人收集某领域内的信息.例如，要求工人提供所有在役 NBA 球员姓名或美国前 100 所顶尖大学名称.值得一提的是单项选择和多项选择任务使用“封闭世界(closedworld)”模型，其中工人只能从给定的一组选项中选择答案；而填空任务和收集任务使用“开放世界(open world)”模型，工人则可以提供开放的答案.大多数众包算子可以使用这四种任务类型来实现，并且有多种用户界面来支持每个算子.

2.2.1.2　任务参数配置

任务发布者主要需要设置以下三种参数：

(1)价格.发布者需要设置每个任务的价格.通常任务价格从几美分到几美元不等.值得一提的是定价是一个复杂的博弈论问题.通常，低价格虽然节省成本，但是会导致工人回答任务积极性变差，质量不高；高价格可以吸引多个工人，从而提高效率；但支付过高的价格并不能无限度地提高回答质量[44].因此，为任务设定适当的价格是一个具有挑战性的问题.

(2)时间.发布者可以为每个任务设置时间限制.对于每个任务，请求者可以设置回答它的时间限制(例如 60 s)，意思是工人必须在这个时间限制内完成它.发布者还可以设置任务的到期时间，例如 24 h，即如果任务在 24 h 之内还没有工人去回答，那么会撤销这个任务.

(3)质量.发布者可以选择众包平台提供的质量控制技术，或者设计自己的质量控制方法.例如，发布者可以通过向工人提供一些小测来测试工人的质量，而不能通过测试的工人则无法回答接下来的任务.发布者还可以使用隐藏测试，将一些已经有正确答案的题目混合到正常任务中(但工人不知道有一些隐藏任务)，然后筛选出质量较低的工人.我们将在第 3 节中详细讨论质量控制技术.

2.2.1.3　任务布置

发布者需要预付资金到众包平台，其中包括工人成本和平台成本.工人成本可以通过任务数量×每个任务的价格×为每个任务分配的工人数量来计算，平台成本等于工人成本×平台抽成比例(例如

《计算机学报》

5%).预付款后,发布者可以将任务发布到平台.接下来,发布者可以监控任务的回答进程,例如检查已完成任务的百分比和检查返回答案的质量.最后,发布者可以收集所有答案,还可以评估工人们的绩效.众包平台根据这些绩效为员工分配质量评估结果.

2.2.2 工人角度的众包流程

从工人的角度来看,工人可以浏览和选择任务发布者发布的任务.在接受任务后,工人们必须在请求者指定的时间段内完成任务.如果工人完成了一项任务,任务发布者可以通过或拒绝工人的回答.经通过的工人将从任务发布者处获得报酬.如果工人的质量不高,他们也可以被任务发布者或平台淘汰.

2.2.3 众包平台角度的众包流程

当任务发布者们发布一些任务后,众包平台会公布这些任务,工人们可以看到平台上的任务.当一个工人提交一份答案时,平台收集答案,并将答案返回相应的发布者.当一个工人请求任务时,平台会将一个任务分配给该工人.通常,分配的任务是由平台随机指定的.在一些平台中,任务发布者可以控制任务的分配,这是质量控制的一个重要手段.

2.2.4 众包流程的痛点

在传统众包的流程中,任务发布者需要做大量的工作,例如设计很多参数,设计 HTML 模板.这些都是很繁杂的工作.尤其当一个任务需要多轮迭代时,任务发布者还需要设计真值推理以及下一轮问题的选择算法,以至于需要编写程序.这对于非计算机专家来说是很难的.因此出现了众包数据库,在这里,任务发布者只需要准备数据(甚至是脏数据),简单定义单位任务的价钱,然后发布数据和相关类 SQL 语言即可.剩下的任务分解、发布、收集结果、推理答案、迭代发布统统交给众包数据库来完成,大大降低了发布任务的难度.具体流程在第 4 节中详述.

3 众包相关技术

质量控制、成本控制和延迟控制是众包相关技术的三要素,分别致力于提升质量,降低成本和减少延迟.其中提升质量重点关注于如何为工人建模,推理正确答案以及分配任务.降低成本主要关注于利用机器算法进行剪枝以及答案的推理.减少延迟主要关注于如何并行地发布任务.这三者直接存在互相依赖的关系.例如,如果我们希望提高质量,那么大概率就要多问一些问题然后利用更多的信息来推理正确答案,从而导致成本的提高.在众包数据库中,这些传统众包相关技术(质量、代价和延迟)等都可以被应用.因此,了解这些技术是设计并调优众包数据库的基础.

3.1 质量控制

3.1.1 工人模型

现有工作采用不同的方法来为工人建模,我们将它们总结归纳如下:

概率模型(worker probability)[19,27,45-47].概率模型将每个工人的质量建模成一个参数 $q\in[0,1]$,表示工人给出正确答案的概率.例如,如果一个工人有 80%的概率正确回答这个问题,那么这个工人的质量会被建模为 $q=0.8$.还有一些工作扩展了概率模型,例如文献[48-50]扩展到一个任意大小的实数,q 的值越大,说明这个工人越有能力回答正确的答案.还有文献[51]引入了置信区间来为工人建模(例如[0.6,0.8]),表示求出的置信度的大小.通常来说,工人回答的问题越多,我们求出的工人质量 q 的置信区间越高,例如[0.9,0.95].

混淆矩阵模型(confusion matrix)[48,52-54].混淆矩阵模型通常用来为回答单选题(ℓ 个选项)的工人建模.例如,一个情感分析任务让工人给出一段文字的情感倾向(积极、消极和中立),那么 $\ell=3$.混淆矩阵表示如下:

$$\boldsymbol{Q}=\begin{bmatrix} Q_{1,1} & Q_{1,2} & \cdots & Q_{1,\ell} \\ Q_{2,1} & Q_{2,2} & \cdots & Q_{2,\ell} \\ \vdots & \vdots & \ddots & \vdots \\ Q_{\ell,1} & Q_{\ell,2} & \cdots & Q_{\ell,\ell} \end{bmatrix},$$

其中每个元素 $Q_{i,j}(1\leq i,j\leq\ell)$ 表示在正确答案是第 j 个选项的前提下,工人的答案是第 i 个选项的概率.形式化的表示为

$Q_{i,j}=\Pr(\text{工人的答案是 } i \mid \text{任务的真实答案是 } j)$.

举例来说,如果一个工人对于以上的情感分析任务的质量是 $\boldsymbol{Q}=\begin{bmatrix} 0.8 & 0.4 & 0.2 \\ 0.1 & 0.4 & 0.1 \\ 0.1 & 0.2 & 0.7 \end{bmatrix}$.如果第一个选项是积极,第二个选项是中立,第三个选项是消极,那么 $Q_{3,2}=0.2$ 表示如果这段文本表达的是中立态度,那么该工人回答消极态度的概率是 0.2.

3.1.2 计算工人质量

我们得到了工人质量模型,接下来我们可以根据以下方法计算工人的质量.

资格测试.资格测试包含一组已知真实答案的

任务.工人必须通过资格测试(例如,在这些已知答案的任务上取得良好的正确率),才能回答真正的任务.根据工人的回答,可以得出工人的质量.例如,如果工人被建模为概率,并且工人正确回答了 10 个已知答案任务中的 8 个,那么质量为 $q=0.8$.

注入已知答案[27].该方法将已知答案任务与实际任务混合在一起.例如,假设每次给一个工人分配 10 个任务,其中 20%的任务(即 2 个任务)被设置为已知答案的任务.根据工人对这些任务的回答,可以计算出工人的质量.与资格测试不同的是,工人们并不知道这些已知答案任务的存在.这样可以防止工人在通过资格测试之后不认真答题.

EM 模型[20,48,52-59].期望最大化(Expectation-Maximization)算法利用所有工人对所有任务的答案作为输入,迭代更新每个工人的质量和每个任务的真实答案直到收敛.通常来讲,经常给出高质量答案的员工将被定位为较高质量员工,由高质量员工回答的答案将更有可能被作为真正的答案.一般 EM 框架采用迭代方法,每次迭代采用 E 步和 M 步,如下所示:

E 步骤.利用前一个 M 步骤得到的参数计算隐变量的分布;

M 步骤.计算最大化期望对数似然函数的参数.该函数是通过 E 步骤得到的隐变量计算而来.

例如,在文献[20]中,隐变量包含真实答案的概率分布,参数包含工人的质量和以及先验.在 E 步骤中,每个任务的真实答案的分布是根据工人的质量和先验计算的(上一个 M 步骤);在 E 步骤中,工人的质量和先验是根据每个任务真实答案的概率分布(上一个 E 步骤)计算的.

图模型[45,50,60-62].现有基于图模型的工作中将工人的质量建模为一个节点(或多个节点的组合),并利用图模型推理[63]来间接获得工人的质量.其中图模型的每个节点表示一个随机变量,每个边表示节点对之间的可能关系(例如条件依赖关系).比如,Karger 等人[50]使用二分图对工人和任务建模,并提出基于信念传播的迭代方法[88]来推导工人质量.Liu 等人[26]使用有向图对工人和任务建模,并应用变分推理方法迭代地计算工人质量.

3.1.3　答案聚合

由于仅依靠单个工人的答案,对于正确答案的推断一定有偏差,所以大多数现有的工作[27,45,52,55,64]使用投票策略作为一种质量控制策略.基本思想是将每个任务分配给多个工人,并通过汇总这些工人的所有答案来推断其结果.具体来说,投票策略可以看作是一个函数,它以两组参数作为输入:(1)多个工人对一个任务的回答;(2)每个回答该任务的工人的质量.输出是任务的推断结果.典型的投票策略包括多数投票[27,45]、加权多数投票[65-66]、贝叶斯投票[55]等.

多数投票选择获得最高票数的结果.假设一个任务是要求工人判断"IBM"是否等于"Big Blue".如果三名工人给出的答案分别为"是"、"是"和"否",则多数人投票结果为是,因为"是"获得的票数最多(2"是",1"否").加权多数投票采用相似的策略,但是其考虑到每个工人的质量.一个高质量的工人的票将被赋予更高的权重.对于上面的例子,如果三名工人的质量分别为 0.2、0.6 和 0.9,则"是"和"否"的聚合权重分别为 $0.2+0.6=0.8$ 和 0.9,因此结果返回"否".另一种投票策略是贝叶斯投票,它不仅考虑到每个工人的质量,而且利用贝叶斯定理计算出每个答案是真值的概率分布.同样根据上面的例子,我们有:

$$Pr(\text{"是"}) \propto 0.2\times 0.6\times(1-0.9)=0.012,$$

$$Pr(\text{"否"}) \propto (1-0.2)\times(1-0.6)\times 0.9=0.288.$$

通过归一化,我们可以得到答案"是"和"否"的概率分别为 4%和 96%.所以我们选择"否"为最终答案.

3.1.4　任务分配

定义 2(任务分配).　给定 n 个任务,当一个工人来请求回答时,任务分配问题指的是应该选择哪 k 个问题来分配给这个工人.

在 AMT 中,这个问题通常以离线方式解决,即所有任务的分配在问题展示给工人之前就已经完成了.正如文献[27,67]中指出,这种方法的主要缺点是不考虑任务的难度:对于简单的任务,即使从工人那里收到比较少的答案,也可以准确确定其最终结果,所以没必要继续分配更多任务.而更困难或有争议的任务,可能需要更多的工人来回答.值得一提的是任务发布者可能只有有限的预算来雇佣工人.因此,为了在有限的预算下获得最佳答案,任务的合理分配是非常重要的.此外,该问题是非常复杂的.给定 n 个任务,有 $\binom{k}{n}$ 组候选任务.而快速选择最优分配方法对任务发布者来说是必要的.因为如果等待很长时间,工人会离开.考虑 $n=100$ 和 $k=5$,简单的枚举将考虑所有 100 万种可能的组合,计算成本非常高.此外,我们需要高效的在线任务分配算法.合理的分配时间是指分配过程可以在 $O(n)$[68]内完

《计算机学报》

成,即与任务数量成线性关系.为了解决这个问题,现有的工作[27,67]通常侧重于为工人分配最合适的任务.为明确"合适"的概念,现有工作[66-67,69-71,102]主要考虑以下两个因素:

任务的不确定性.根据多个工人的答案来分析目前任务的不确定度,优先选择不确定度较高的任务.例如对于一种答案为"是"或"否"的任务,任务 A 目前收到 5 个"是",0 个"否".任务 B 收到 3 个"是",3 个"否".那么我们显然应该选择任务 B 去提问.这种不确定性可以用信息熵来衡量,熵越大代表不确定性越高,越需要优先提问.

工人质量.考虑到当前请求任务的工人的质量,并将任务分配给具有相应专业知识的工人.通过主题模型等方法,我们可以将众包任务分类,例如时政、运动、娱乐等等.同样不同的工人会擅长不同的领域,我们也可以将工人擅长的类别分类.当一个工人请求任务时,我们可以将他擅长的任务分配给他,以期获得更高的质量提升.

3.2 成本控制

尽管众包平台为任务发布者提供了相对经济的解决方案,但是面对大数据时,众包成本依旧是很高的.因此,如何在保证质量的前提下,减少众包成本是一个很有挑战性的问题.在这一节中,我们将介绍几种减少成本的方法.

3.2.1 基于机器算法的剪枝

基于机器算法的剪枝是指使用计算机算法预先检查所有任务并删减掉不需要工人去检查的简单任务.基本思想是在许多情况下,很多任务可以很容易地由计算机完成,因此工人只需要做最具挑战性的任务.这种思想已广泛应用于实体匹配[1,53,61,72-75,94-97]和搜索[76].例如,许多众包实体匹配算法通常使用基于计算机的技术(称为相似连接[77])来剪枝掉相似值小于阈值(例如 0.3)的记录对,然后利用工人去检查剩下的实体对.

3.2.2 任务选择

在给定质量要求的情况下,任务分配技术可以通过选择最优的任务来最小化人力成本,以满足质量要求.不同的众包算子需要不同的任务分配策略,例如连接(实体匹配)[61,72-75]、排序[19,78-79]、分类[81]等.我们在设计众包数据库时,对于给定的众包算子首先使用任务选择策略来决定一组最优的任务,一旦这些任务被选择并发送到众包平台,则使用任务分配策略从工人中收集高质量的答案.其中具体的方法会在第 3 节中针对不同算子详细讨论.

3.2.3 答案推理

某些情况下,由某些众包算子生成的任务具有一些内在的关系,这些关系可用于成本控制.具体地说,给定一组任务,在从工人中获得一些任务的结果后,我们可以使用这些信息来推断其他相似的任务的结果,从而节省成本.许多众包算子都有这样的性质,例如连接[72-73,75,80]、规划[21,81]、挖掘[5].比如,假设有三个众包连接任务:(A,B)、(B,C)和(A,C).如果我们已经知道 A 等于 B,B 等于 C,那么我们可以根据传递性推断 A 等于 C,从而避免检查(A,C)的成本.

3.2.4 抽样

基于抽样的技术指的是只利用工人来处理一部分数据样本,然后利用工人对样本的回答来推断他们对全部数据的回答.这类技术在众包聚合算子[20,30]和数据清理[82]中非常有效.例如,文献[82]提出一个抽样清洗框架(sample-clean),允许工人仅清洗一小部分数据样本,并使用清洗的样本从完整数据中获得高质量的结果.

3.3 延迟控制

延迟实际指的是所有任务从发送到众包平台上到全部完成的时间.然而,这一时间非常难估计,因为工人的数量和单位任务的完成时间都是很难确定的.我们首先介绍一种统计模型来模拟工人的答题策略从而估计答题时间.接着为了简化问题,我们将延迟建模为轮数,从而更方便地优化延迟.

3.3.1 统计模型

一些现有工作[44,76]从众包平台收集统计数据,并使用这些信息来模拟工人的行为.文献[76]建立统计模型来预测工人回答任务的时间.其考虑了两种延迟:(1)从发布任务到收集到第一个工人的答案的时间间隔;(2)两个相邻答案的时间间隔.文献[44]使用统计模型预测工人在众包平台中的到达率.在这些统计模型的帮助下,研究者们可以预测一个任务是否可以在指定的时间限制内完成.通过这种预测,我们可以计算一个问题在降低延迟上的收益,从而选择最优的任务.

3.3.2 轮数模型

很多时候全部任务需要多轮进行处理.任务发布者会每轮会向工人发布多个任务,在收集答案后,可以在下一轮中发布另一组选定的任务.假设有足够的工人,文献[83]通过假设每轮花费 1 个单位时间来简化延迟的定义,然后将延迟建模为完成所有任务所需要的轮数.

《计算机学报》

一些现有的工作[83-84]使用轮数模型来进行延迟控制. 假设有 n 个任务，每轮选择 k 个任务，那么完成所有任务总共需要 $\frac{n}{k}$ 轮. 为了降低延迟，文献[83-84]使用第 3.2.3 节中的答案推理思想来减少发布的任务的数量，即利用前一轮接收到的任务的答案来决定下一轮发布的任务. 这样会减少要请求的任务总数 n. 如果我们每一轮选择 k 个无法被直接推导的任务去问工人，则总的轮数小于 $\frac{n}{k}$，从而减少了延迟.

4　众包数据库系统

从第 1 节我们可以了解到任务发布者与众包平台交互相当不方便，因为有各种参数需要设置，甚至需要编写代码. 因此，受传统数据库系统的启发，研究者们设计并实现了众包数据库系统，包括 CrowdDB[17]，Qurk[31]，Deco[37] 和 CDB[24]. 众包数据库系统的设计原则是将众包集成到关系数据库管理系统(RDBMS)中，其架构如图 3 所示. 系统提供声明性接口，允许任务发布者使用类 SQL 的语言来发布涉及众包算子的查询并利用众包驱动的算子来封装任务发布者与众包平台和工人交互的复杂性. 在这个原则下，给定一个类 SQL 查询，系统首先将查询解析成一个涉及众包算子的查询计划，然后生成要在众包平台上发布的任务，最后收集工人的答案以获得结果.

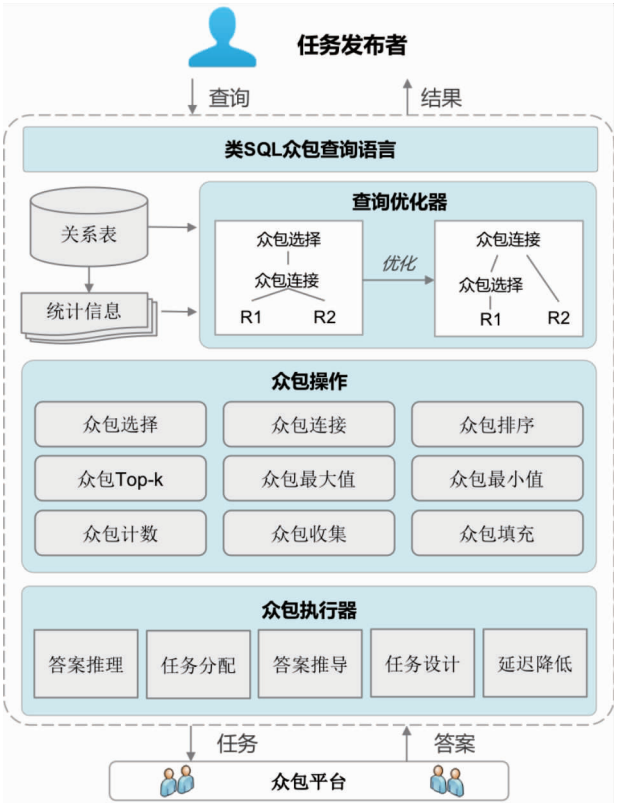

图 3　众包数据库架构

本节的 4.1 节首先概述了众包数据库系统. 然后，第 4.2 节和第 4.3 节分别介绍了众包查询语言和查询优化技术. 最后，通过对第 4.4 节中现有的众包数据库系统进行综合比较，总结本节内容.

4.1　众包数据库概述

众包数据库系统与传统关系数据库系统有两个主要区别. 首先，传统的关系数据库系统使用“封闭世界”模型，只处理基于数据库内部数据的查询；而众包数据库系统使用“开放世界”模型，利用众包收集记录或填充属性. 其次，众包数据库系统可以利用众包来回答对机器比较难的查询，例如，比较两个记录或属性、对多个记录排序和对一个记录打分等算子. 为了更好地理解设计众包数据库的动机，我们用图 3 的例子来具体说明.

给出一个查询 Select * From Professor, Paper Where Professor.Prof=Paper.Author. 由图 4 我们可以发现利用传统数据库，该查询不能得到任何结果，因为 Guo. Li 和 G. Li 虽然指的是同一个人，但是在字符串上的表达却不一样. 因此，传统数据库并不能解决这种存在实体匹配的连接问题. 再者，给出查询 Select * From Professor Where Professor. Prof=“Qizhi Yao”. 由于数据库表中没有“Qizhi Yao”的相关记录，传统数据库无法给出任何查询结果. 但是我们可以求助于众包工人提供相关结果，填充到数据表中，然后进行查询. 所以我们设计了如图 3 所示的众包数据库. 首先任务发布者给出一条涉及众包算子的查询语句(详见 4.2 节)，与传统数据库类似，我们首先解析这条查询. 然后选择最优的执行计划. 在执行计划中涉及到很多众包算子，例如众包选择、众包连接等. 根据不同的算子，我们会生成不同的众包问题发布到众包平台上，工人回答问题后返回结果. 有关任务分配和真值推理我们可以用第 3 节的内容来优化. 在本节中，我们主要关注现

Professor

Prof.	Affiliation
Guo. Li	Tsinghua
J. Fan	Renmin Univ.

Paper

Author	Title	Conf.
G. Li	xxx	SIGMOD17
Ju Fan	xxx	SIGMOD16

图 4　例子：待连接数据表

《计算机学报》

有众包数据库系统如何设计查询语言以及查询优化策略.

与传统众包技术的区别. 首先,第 3 节的技术都可以应用到众包数据库中.例如,在质量控制中,我们可以在众包数据库中保存工人质量的元数据,根据工人质量,采用 3.1.3 节和 3.1.4 节的技术进行任务分配和真值推理.在成本控制和延迟控制中,当对单个关系表进行操作时,3.2 节和 3.3 节的技术依旧可以应用.不同的是,众包数据库存在多个关系表参与的复杂查询.面对这些查询时,我们可以通过以表为单位的查询优化策略通过不同算子的执行顺序来优化代价和延迟,如接下来 4.3.1 节、4.3.2 节和 4.3.3 节所述.也可以采用更加细粒度的优化,如 4.3.4 节所述,采用以记录为单位的优化策略.这与传统众包技术的本质不同.

4.2 众包数据库查询语言

4.2.1 CrowdDB

CrowdDB[17]设计了一种称为 CrowdSQL 的查询语言以支持缺失数据的外部获取和众包比较算子.CrowdSQL 是标准 SQL 的自然扩展,新增了三个关键词:CROWD、CROWDEQUAL 和 CROWDORDER.

(1) CROWD. CrowdDB 在 CrowdSQL 的数据定义语言中引入关键字 CROWD,定义哪些属性或者数据表可以进行众包算子,从而允许 CrowdDB 可以从工人中收集该数据.如图 5 所示,有两种数据可以由工人来补全:① 记录的特定属性值可以被众包获取;② 整个记录可以被众包获取.CrowdSQL 使用 CROWD 关键字在 CrowdSQL 的 Create Table 语句中指定这两种情况.

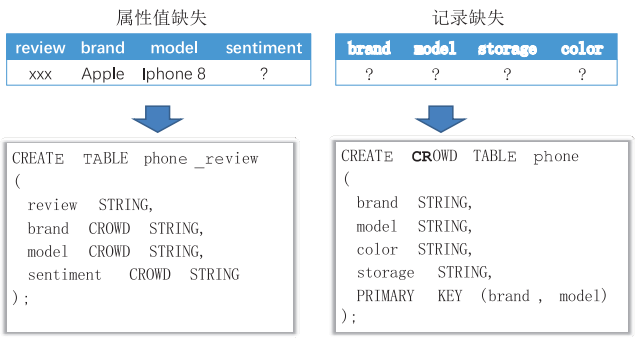

图 5 数据定义语言(CrowdDB)

(2) CROWDEQUAL 和 CROWDORDER. 这两个关键词便于众包工人来解决对机器来说比较难的问题.例如,图 6 左侧的查询利用 CROWDEQUAL(符号~=)来让工人帮助识别满足特定约束条件的手机评论,例如评价是"正面的".另一种算子是基于众包的排序,如图 6 右侧所示.给定一组可比较但很难被机器比较的记录,CROWDORDER 可以基于某种标准(例如该查询中的"价格")来处理众包排序任务.

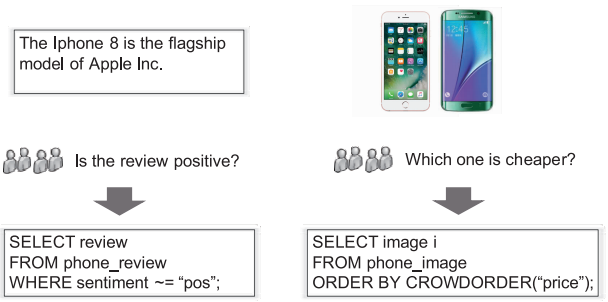

图 6 查询语言(CrowdDB)

4.2.2 Qurk

Qurk[31]引入了一种不同的方式,即使用用户定义函数(User Defined Function,UDF)来扩展 SQL 语言.Qurk 中的一个 UDF 可以被视为一个用于生成众包任务的模板.如图 7 所示,该查询定义了一个名为 isWhite 的 UDF,用于请求工人识别白色手机.

(1) Type. 该字段预定义了一些众包任务,包括选择、排序、连接和收集.

(2) UI Generation. 如 Prompt、YesText 和 NoTest,这些参数确定了任务 HTML 界面的样子.

(3) Combiner. 该参数确定了质量控制的方式.如多人投票.

```
SELECT i.image
FROM phone_image i
WHERE isWhite(i)

TASK isWhite(field) TYPE Filter:
  Prompt: "<table><tr> \
      <td><img src='%s'></td> \
      <td>Is the phone in white color?</td>\
      </tr></table>", tuple[field]
  YesText: "Yes"
  NoText: "No"
  Combiner: MajorityVote
```

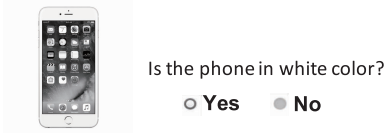

图 7 数据定义语言(Qurk)

4.2.3 Deco

Deco[37]主要侧重于众包数据收集.下面我们将介绍它的数据模型和查询语法.

(1) 数据模型. 为了便于众包数据收集,Deco 将任务发布者和系统的视图分离,如图 8 所示.一方

面，它允许任务发布者来定义概念关系，从而在逻辑上组织要收集的数据的模式. 更具体地说，它将属性分为两种：①锚(anchor)属性在传统 RDBMS 中扮演类似于主键的角色；② 依赖(dependent)属性，这些属性可以被视为记录的“属性”. 锚属性和依赖属性的设计可以指定数据收集过程中的依赖关系，如图 8 上部分所示. 另一方面，原始模式(raw schema)实际存储在底层 RDBMS 中的数据表并且对用户不可见.

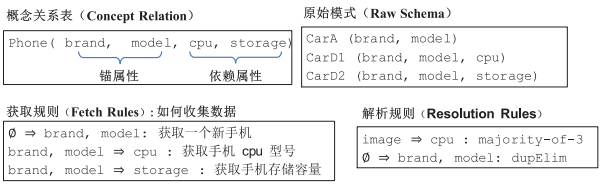

图 8　数据定义语言(Deco)

(2) 获取(Fetch). Deco 设计了数据获取规则，用来指导如何根据表模式从工人中获取数据. 一个规则的语法是：给定属性 A_1 的值，让工人获取属性 A_2 的值，表示为 $A_1 \rightarrow A_2 : P$，其中 A_1 是一组用作获取条件的属性，A_2 是要获取的属性，P 是生成众包任务的过程. 图 8 下部分提供了三个获取规则的示例，分别要求填充一个新的手机，填充现有手机的 cpu 种类和存储容量.

(3) 解析规则(Resolution Rules). 由于收集的数据可能存在不一致性，Deco 还可以指定解决这些不一致性的规则，即质量控制，如多数投票. 当前版本的 Deco 设计了两种解析规则，如图 8 所示：① majority-of-3 采用三个工人的多数投票来解决不一致问题；② dumElim 用于从工人答案集合中删除重复的记录. 显然，前者通常用于填充属性，后者经常用于收集记录.

Deco 还允许任务发布者使用关键字 MINTUPLES 编写类 SQL 语句. MINTUPLES 的用法与 SQL 中 LIMIT 关键字的用法类似. 但不同的是：MINTUPLES n 意思是至少从工人中收集 n 个元组，LIMIT n 是最多从关系表中检索 n 个元组. 图 9 提供了一个示例查询，该查询收集了至少 8 个“苹果”手机.

查询：收集至少**8**个苹果手机的信息

```
SELECT brand, model, cpu, storage
FROM Phone
WHERE brand = "Apple" MINTUPLES 8
```

图 9　查询语言(Deco)

4.3　查询的执行与优化

4.3.1　CrowdDB

在 CrowdDB 中，查询引入了以下三种算子：

(1) CrowdProbe. 该算子用于利用工人收集属性或新元组的缺失信息. CrowdProbe 的典型用户界面是一个具有多个字段的表单，用于从工人中收集信息.

(2) CrowdJoin. 该算子在两个表上实现索引嵌套循环连接，其中至少有一个表是可以被众包收集的. 特别地，内表(inner relation)必须是一个 CROWD 表(见 4.2.1 节)，图 10 中的用户界面用于众包收集内表的新元组，这些新元组可以与外表(outer relation)中的记录连接.

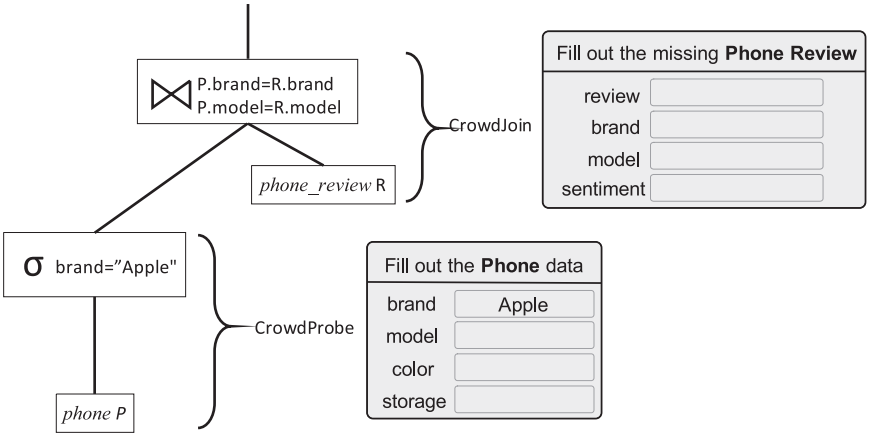

图 10　查询优化(CrowdDB)

(3) CrowdCompare. 该算子用于实现 CrowdSQL 中定义的两个关键字：CROWDEQUAL 和 CROWDORDER. CROWDEQUAL 让工人比较两个记录或属性是否指代同一实体. CROWDORDER 要求工人在预先定义的属性上给出一个排序.

图 10 给出了这些算子的例子. 算子 CrowdProbe 和 CrowdJoin 如图 10 所示. CrowdProbe 用于从工人中收集品牌是“Apple”的手机，而 CrowdJoin 则连接

手机表和手机评论表.由于有的手机评论是缺失的,那么就需要收集缺失的评论.如图 11,算子 CrowdCompare 用于支持关键字 CROWDEQUAL(例如,"IBM==BIG BLUE")和 CrowdOrder(例如,哪个图片更清晰地展示"清华大学").

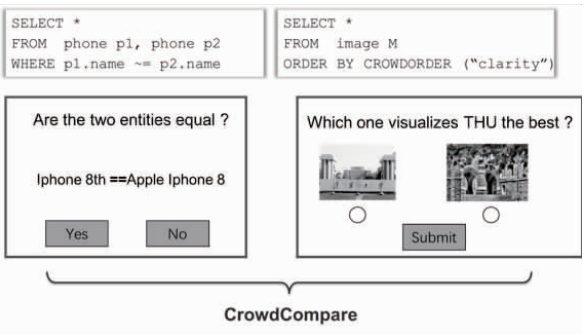

图 11 问题生成(CrowdDB)

对于查询优化,CrowdDB 采用与传统数据库相同的基于规则的优化技术,用于处理具有多个算子的查询.例如选择下推和确定最优的连接顺序.

4.3.2 Qurk

Qurk 主要专注于连接和排序的查询优化.

(1) CrowdJoin. 与 CrowdDB 类似,Qurk 同样实现了一个嵌套循环联接,并对两个表中的记录进行众包收集算子以便进行连接算子.为了降低成本,Qurk 提出了批处理多个比较的技术.图 12(a)展示了 Qurk 的两种批处理方案:简单的批处理只需工人在众包任务中对多个实体对比较,而智能批处理则要求工人连接实体以完成比较.

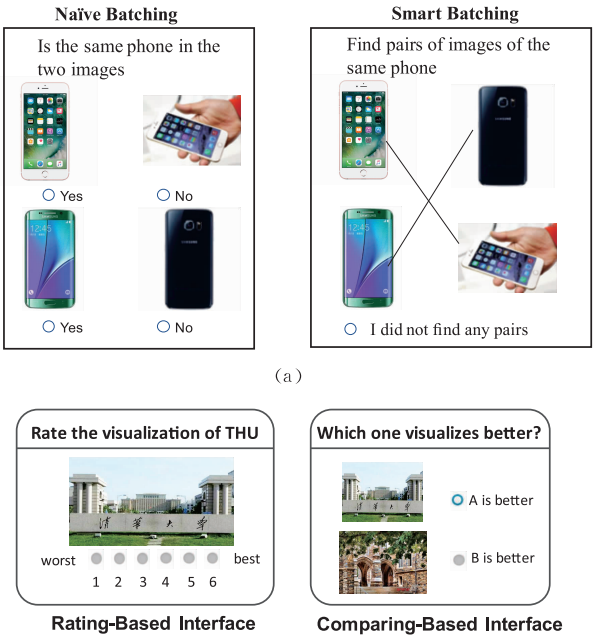

图 12 问题生成(Qurk)

(2) CrowdSort. Qurk 实现了两种排序方法,如图 12(b)所示,右图基于比较的方法要求工人直接指定实体的顺序.由于需要 $O(n^2)$ 的比较次数,该方法对于大型数据集是不可行的.另一种方法如左图所示,采用打分的方法.该方法通过友好的接口来降低人工成本,只需要 $O(n)$ 的复杂度.

Qurk 还有两个重要的成本控制模块:任务缓存和任务模型.任务缓存维护以前任务的众包答案,任务模型根据任务缓存的答案训练一个模型.因此,如果一个还未发布的任务可以从任务模型中获得置信度足够高的答案,那么它将不会被发布到众包平台.一旦任务无法从任务模型中获得有用的信息,它将被发布到众包平台.

4.3.3 Deco

如 4.2 节所述,Deco 重点关注用众包收集缺失值或新元组上.为此,Deco 设计了 Fetch 算子,该算子符可以为收集算子生成众包任务.此外,Deco 还支持其他基于机器的算子以处理获取的数据,例如左外部联接(Left Outer Join)、过滤(Filter)和扫描(Scan).

基于以上定义的算子,在给定复杂查询的情况下,Deco 定义了一个基本查询优化问题:

对于一个查询,如何找到最佳查询计划,即在所有可能的查询计划中,哪一个具有最小的成本?

为了解决这个问题,考虑到数据库中已存在一些数据的事实,成本被定义为需要从工人中获得的新数据的多少.为了找到成本最低的最佳查询计划,Deco 解决了以下两个问题.

(1) 成本估算.在执行查询计划时,数据库可能会从工人中收集新的数据,这可能会影响后续流程的成本估算.考虑此种情况,Deco 提出了一种迭代方法来估计查询计划的成本.考虑获取 8 个苹果手机的算子.该算子首先检查数据库的当前状态,并找到 2 个苹果手机、1 个三星手机和 1 个华为手机.由于苹果手机的选择率(selectivity)为 0.1,因此该数据库估计苹果手机的期望数量为 2.1.基于此,它为 Fetch 算子设置了一个 5.9 部手机目标.

(2) 根据 5.9 个苹果手机的目标,由于苹果手机的选择率为 0.1,Fetch 算子估计要从工人中至少获取 59 辆汽车.基于此,它发布相应的众包任务以获取数据.

最佳查询计划生成:简单地枚举所有可能的查

询计划的策略是昂贵的，所以 Deco 重用公共子计划以减少冗余计算. 考虑收集“苹果手机”的例子，我们可以先无条件的收集手机记录，然后使用“苹果手机”过滤这些记录. 相反，第二种反向查询计划利用“苹果手机”作为获取新记录的条件，比第一个查询计划成本要低.

4.3.4　CDB

目前的众包数据库系统都采用传统的树型查询计划，其目的是选择一个优化的表级连接顺序来优化查询. 但是这样表级的优化是比较粗糙的，CDB[24]定义了一个基于图的查询模型，以便对众包数据进行细粒度的优化. 具体地说，给定一个类 SQL 查询，CDB 会构造一个图，其中每个顶点是该查询语句中每个表的记录，每个边根据该查询语句的连接/选择谓词连接两个记录. 该图可提供了记录级别的细粒度优化. 更具体地说，每一条边都可以作为一个众包问题，询问工人是否可以连接构成这条边的两条记录. 然后，众包结果可以被标注在图上，如图 13：蓝色实线（红色虚线）的边表示两个记录能（不能）成功连接. 在此基础上，给出了一个带有 n 个连接谓词的查询，该查询的答案是一条长为 $n+1$ 的蓝色路径. CDB 可以支持基于众包的选择、连接、聚合、排序和 Top-k. 基于该图模型，CDB 在从成本、延迟和质量三个维度上考虑了查询优化.

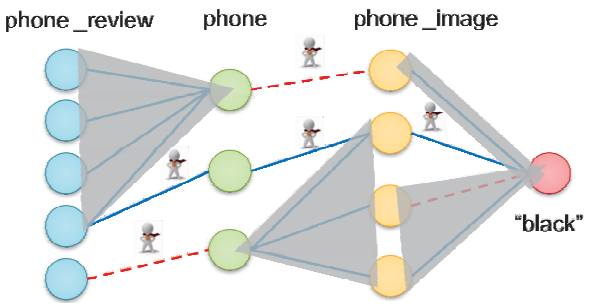

图 13　例子：CDB 的查询优化

（1）成本控制. 基于图模型的查询优化为成本控提供了很多的优化空间. 从图 13 可以看出，树模型确定了最佳连接顺序，首先在手机评论和手机之间需要发布 5 个问题，然后在手机和手机图像需要发布 2 个问题，最后在手机图像和选择条件 color＝“black”之间需要发布 1 个问题. 总的来说，这会花费 8 个任务. 但是，基于图模型的方法倾向于找出红色的边去提问，要求是这些红色的边可以剪枝掉很多不必要的提问. 例如，如图 13 所示，首先可以询问手机和手机图像之间的第一条边，并将其标注为红色虚线，以便剪枝掉与此边相关联的所有边. 然后相似地，它优先提问其他具有强力修剪能力的边. 这样基于图的模型只需要询问 5 个任务. CDB 证明了图模型中的问题选择的优化问题是 NP 难的，并提出了若干启发式算法，如贪心算法、基于期望的算法等来解决该问题.

（2）延迟控制. CDB 采用基于轮数的模型来衡量延迟，基本思想如下：给定两条边 e 和 e'，CDB 检查它们是否在同一个候选路径中（即可能是答案的路径）. 如果它是，CDB 会将这两条边标记为“冲突”，因为询问一条边可能会导致另一条边被剪枝. 这样的话同时问这两条边是浪费成本的；如果是“否”，则标注他们是“非冲突”的. 为了降低延迟，CDB 将同时询问非冲突的边但不同时询问冲突的边. 例如，考虑两个冲突的边. 如果一条边的答案其实是红色的，则不需要询问另一条边. 为了检查两个边是否在同一个候选路径中，暴力的方法是枚举一个边的所有候选. 然而，这种方法是十分昂贵的. 因此，CDB 提出了基于连通分量的方法来检测是否可以同时询问任意两个边.

（3）质量控制. 为了在工人答案的基础上得到高质量的结果，系统必须进行质量控制. CDB 的质量控制主要分成两个模块，答案推理和任务分配（技术细节见第 2 节）. 此外，CDB 还支持四种类型任务的质量控制：单项选择、多项选择、填空和收集任务.

4.4　众包数据库的索引

索引在传统数据库中有着举足轻重的作用，它可以大大提升数据库的查询效率. 那么在众包数据库中，我们依然可以对数据建立传统的索引，如 B＋树、哈希索引、位图索引等. 当一个查询不涉及众包算子时，我们依然可以利用以上索引来加速查询. 同时，众包数据库也可以支持基于众包的索引[85]. 然而，至今还没有一个数据库真正地将众包索引部署在众包数据库中. 接下来，我们将重点介绍众包数据库索引的模型及其使用方法. 然后在第 6 节进一步讨论有关索引的未来工作.

4.4.1　众包数据库索引的模型

众包数据库索引主要索引一些有序的数据，但是顺序关系却不能被计算机精确给出. 例如给出两辆肇事车辆的图片，判断哪辆车的维修费更高昂. 如图 14 所示，众包数据库索引模型由索引管理器和传统 B＋树组成. 其中 B＋树的每个节点存储 n 个有序的键值（例如按照维修费用排好序的 n 个肇事车辆的图片的地址）. 索引管理器主要负责索引的构建、查询，包括产生任务以及收集并推理答案. B＋

树的每个节点和查询实体构成一个众包任务，如图 15 所示. 最上面的图片是查询图片，下面是 B+树的一个节点. 众包工人的任务就是判断查询图片应该在哪个恰当的位置.

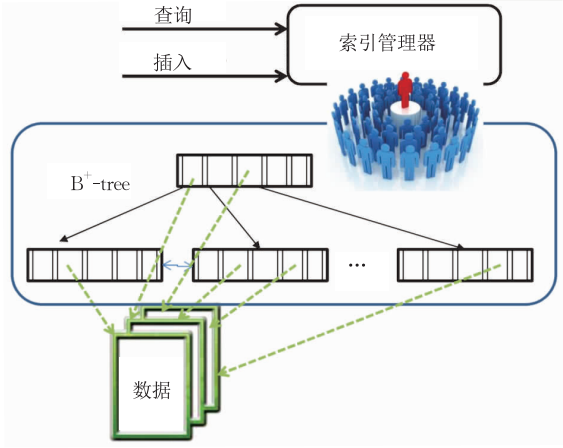

图 14 众包数据库索引模型

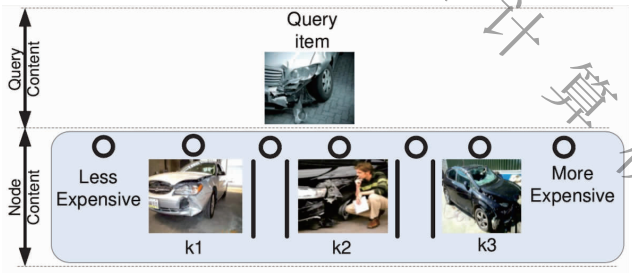

图 15 众包数据库索引任务示例

4.4.2 众包数据库索引的工作模式

我们在上一节中概述了众包数据库索引的模型，接下来我们将讨论如何利用该索引进行查询. 显而易见，给出索引和查询，最直接的查询方法是让一个众包工人自顶向下回答问题，从而定位查询所在位置. 然而，工人是会犯错的，文献[85]通过投票的方式来进行质量控制，并将投票策略分成以下三种.

(1) 基于在叶节点聚集的策略

如图 16 所示，给出一个查询，基于叶节点的策略会将之分给若干个工人，例如图中的 w_1，w_2 和 w_3. 该策略让不同工人分别将查询进行到叶节点，然后在进行投票. 如图所示，w_1，w_3 返回 12，w_2 返回 8，那么众包索引将返回 12. 该策略的优点是实现简单，多个工人可以并行工作，不用互相等待从而降低了延迟. 但缺点是工人的答案容易分散，最后的结论易错. 鉴于此，文献[85]提出了基于在全部层聚集的策略.

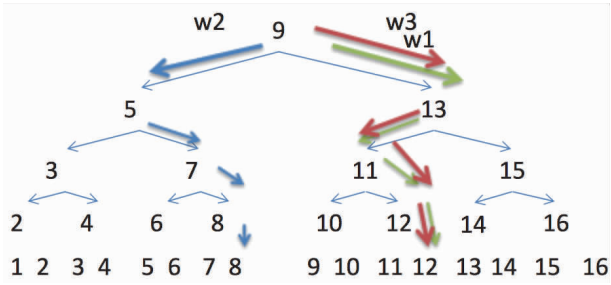

图 16 基于叶节点聚集的示例

(2) 基于在全部层聚集的策略

如图 17 所示，给出一个查询，基于全部层聚集的策略会首先对第一层的结果投票，例如 w_1，w_3 觉得比 9 大，w_2 觉得比 9 小，那么在该层的决策是走向 13. 然后再循环提问直到达到叶节点. 该方法保证了在每一层的正确率较高，以期最后的正确率较高. 但是其缺点是不可以回溯，例如到达了 13 这个点后，就再也回不到 9 的左侧. 下面提出的基于回溯的方法解决了这个问题.

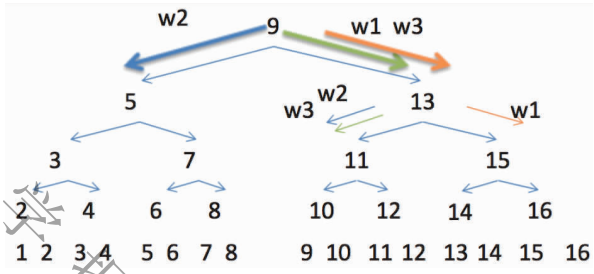

图 17 基于全部层聚集的示例

(3) 基于回溯的策略

如图 18 所示，给出一个查询，基于回溯的策略首先在第一层使用多人投票的方法，决定向哪个方向搜索. 同时按照对于不同选择，人的投票数量维护一个优先队列. 例如在图 18 中，会首先向 B 走，但是同时也会记录下工人投票给 C 的数量. 接着在点 B 提问，如果 C 的投票数比 D 要多，那么会回溯到 C 去提问.

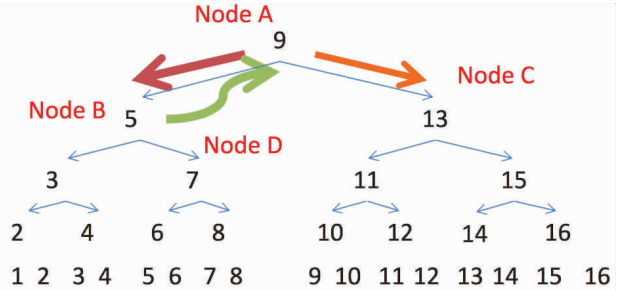

图 18 基于回溯策略的示例

《计算机学报》

4.5 众包数据库系统比较

综上，我们绘制了图 19 在多个维度上对以上四种众包数据库进行了比较.

(1) 最优化模型. 众包数据库的查询优化方法主要分为两类，基于规则的和基于代价的. CrowdDB 采用基于规则的方法，即下推选择算子和连接顺序选择，但是这种方法很有可能会丢失最优的方案. Deco 和 Qurk 采用基于代价的方法，即利用树模型来选择代价估计最小的查询计划. 正如上面分析的，对于相连接的记录，不同的树模型顺序给出的顺序是相同的，这样限制了我们的优化空间. 因此，CDB 设计了细粒度的基于图模型的优化方案，相比树模型而言省下了很多花费.

(2) 最优化目标. 对于众包查询的最优化应该考虑成本、质量和延迟之间的折中，完美的优化很难实现，例如低成本低延迟高质量是不现实的. 如图 19，很多系统会优化成本和质量，但是不优化延迟. CDB 利用了数据之间的独立性来降低延迟，是一种非常有效的方法. 考虑到质量，大多数系统仅用简单的多数投票或者其变种，仅能支持单选题. 但是 CDB 考虑到记录级别的优化，同时支持单选、多选、填充和收集.

(3) 优化的众包算子. 如图 19，CrowdDB 优化选择、连接、收集和填充. Qurk 主要优化选择和连接. Deco 则优化填充和收集，同时也支持选择和连接.

(4) 任务布置. 以上系统一般仅支持与单一众包平台相连，如亚马逊的 AMT，这样导致结果有可能被平台的误差所影响. 因此，CDB 可以将任务发布到多个平台然后跨平台地收集答案并推理结构.

	优化的基于众包的操作				优化目标			优化策略			任务布置平台
	收集	填充	选择	连接	成本	延迟	质量	代价模型	基于记录的模型	支持预算	跨平台
CrowdDB	✓	✓	✓	✓	✓	×	MV	×	×	×	×
Qurk	×	×	✓	✓	✓	×	MV	✓	×	×	×
Deco	✓	✓	✓	✓	✓	×	MV	✓	×	×	×
CDB	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

图 19　众包数据库系统比较

5 众包数据库中的算子

第 3 节中针对不同众包数据库算子的优化，都是采用一些比较简单的策略. 而实际上，有很多方法都可以用来更好地优化这些算子. 在这一节中，我们将会概述相关研究工作并着重介绍一些代表性的工作.

5.1 选择算子

给出一些记录，基于众包的选择算子目的是找出某些满足给定条件的记录. 例如，给定一些图片，目的是找出包含山水的图. 现有的基于众包选择的算子可以分为众包过滤和众包查找.

定义 3(基于众包的过滤[34-35]). 基于众包的过滤算子返回所有满足给定条件的记录.

定义 4(基于众包的查找[83]). 基于众包的查找算子返回 k 个满足给定条件的记录.

例如，给出 100 张图片，假设有 20 张包含山水. 基于众包的过滤要求找出所有这 20 张图片. 基于众包的查找会选择一个阈值，比如 5 张满足条件的图片作为结果返回. 下面将介绍解决以上两个经典问题的方法.

5.1.1 基于众包的过滤

在这一问题中，我们假设所有的待过滤元素都是独立的. 例如知道一张或几张图片的答案不能推出另外未问的图片的答案. 那么为了保证高质量的答案，对于每一个元素，我们要问多个工人，然后利用多数投票的方式推理正确答案. 如此，问题就变成了：对于每个元素，让多少个工人来回答合适呢？如果回答的人数少了，质量可能不高. 如果回答的人数多了，可能会浪费成本却并不能大幅提高质量. 因此，文献[35]把这一问题形式化为一个状态矩阵，如图 20. 每一个元素都有一个对应的状态矩阵，其中横坐标表示该图片目前获得的“满足”票数，纵坐标表示目前获得的“不满足”的票数. 显然，对于每个元素任务发布者将从(0,0)开始提问，如果获得一个“满足”票，向右前进 1 步；获得一个“不满足”票，向上前进 1 步. 在每个点(x,y)，任务发布者可以做三种选择：(1) 输出“满足”，结束当前元素的提问并开

《计算机学报》

始下一个；(2)输出“不满足”，结束当前元素的提问并开始下一个；(3)继续提问当前元素.那么如何设计提问策略才能在给定成本的前提下最大化质量呢?文献[35]给出了该问题的定义.

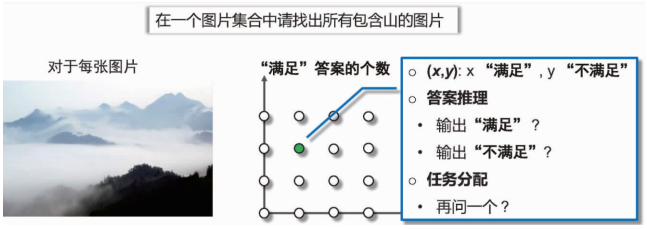

图 20 状态矩阵

定义 5(质量-成本约束). 假设每个元素最多可以问 m 个问题，质量-成本约束问题目的是找到一个最优的提问策略，使得在在该成本(Cost)约束下错误率(Error)最低(即质量最高).

图 21 展示了一个可能的策略.该策略中，如果遇到黄色的点，任务发布者将继续提问.遇到蓝色的点将停止提问该元素并输出“满足”.同理遇到红色点将输出“不满足”.那么该策略的期望成本和期望错误率的计算方法如图 21 所示.期望成本是到达每个蓝色或者红色的点的概率乘以该点的成本$(x+y)$的加和.期望错误的计算同理.到达每个点的概率可以用递归的方法来计算.那么为了找出最优的策略，最暴力的方法是枚举所有可能的策略然后选出质量最高的.文献[35]证明了这是一个 NP 难问题并给出了启发式的算法.

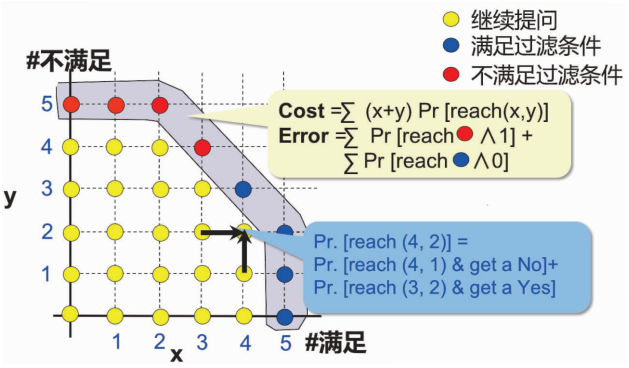

图 21 问题策略

5.1.2 基于众包的查找

文献[83]提出了一种成本-延迟折中的策略.如图 22 所示，如果我们想从 100 张图片找出两张满足要求的图片，首先我们可以串行提问.这样的话，我们仅需要花费四个问题就获得了结果，但是延迟很高，需要与工人交互四轮.如果我们并行提问，即一次性把 100 个问题都发出去，这样延迟很低，但是成本很高(100 个问题).所以我们采取一种混合的方法，例如由于我们要两张图片，所以我们第一轮问两个.其中有一个是满足的，所以我们接下来一个一个提问.我们发现这样成本是 4 个问题，延迟是 3 轮.

图 22 众包查找示例

5.2 连接算子

连接是数据库中的一个重要算子.众包数据库主要关注等值连接，定义如下.

定义 6(基于众包的连接算子). 给定一个表(或者两个表)，基于众包的连接算子目的是找出其中相等的实体对，其中“相等”指的是在现实世界中指代同一实体.

例如“iPhone Four”和“iPhone 4th Gen”，指代的是相同的手机.这一问题也被称为实体匹配或者实体解析，已经被研究了几十年.但是单纯基于机器的方法仍然无法达到很高的质量.下面我们将介绍几种方法来解决这一问题.

(1) CrowER[1]. 该文章率先提出利用众包来连接实体，其方法简单有效，主要分为两步.第一步是候选实体生成.例如，对于 n 个实体进行自连接，最直接的方法是生成$\frac{n(n-1)}{2}$对实体供工人去检查，但是这样的代价过高.因此 CrowER 首先为每对实体计算字符串上的相似度(利用 Jaccard、ED 等相似度函数).如果相似度小于某个阈值，例如 0.3，CrowER 将这些实体归为不能被连接的.然后第二步，将剩下的实体对发布到众包平台上让工人去检查.

(2) Transitivity[72]. 实验证明 CrowER 的效果要好于传统的机器学习方法，但是它的花费仍然是巨大的.因此文献[92]进一步利用传递性来降低成本.具体地，如果已知实体 $A=B$，$B=C$，那么我们可以推理出 $A=C$.这样 AC 这一对的成本就省下来了.相似地，如果已知实体 $A=B$，$B\neq C$，那么我们可以推理出 $A\neq C$，同样节省下了 AC 这一对.文献[72]证明了在已知答案的情况下，按顺序提问所有可以连接的实体对，然后再提问剩下不能连接的实

体对可以节省最多的成本.因此,在不知道答案的情况下,文献[72]先按照字符串相似度对实体对排序,然后按照该顺序提问.

(3) PartialOrder[86]. 该方法进一步利用字符串相似度进行剪枝.对于 n 个实体,每个实体有 m 个属性,对于每对实体,我们计算一个 m 维的相似度向量,其中每一维对应相应属性的相似度.对于这些相似度向量,我们构建一个偏序图,图中的每个节点是一对实体.对于两个节点 A 和 B,如果 A 在每个属性上的相似度都不比 B 小,我们连接一条由 A 到 B 的有向边.接下来,我们选择一个节点去问,如果该对实体是匹配的,我们将其标为绿色,同时其所有祖先也标为绿色(不再去提问其祖先).如果该对实体是不匹配的,我们将其标为红色,同时其所有子孙也标为红色(不再去提问其子孙).如图 23,图中共有 11 个节点,如果我们不采取偏序图的剪枝策略,我们会问 11 个问题.如果采取该策略,我们最少可以提问 4 个问题.文献[86]中详述了如何选择问题以最小化成本及提高质量.

(4) Corleone[87]. Corleone 是一个端到端的基于众包的连接框架,其架构图如图 24 所示.给出一个或两个关系表,Corleone 首先利用众包对表做 block,其目的是利用一些工人提供的规则快速移除一些明显不可能匹配的实体.接着,对于剩下的候选实体对,Corleone 采用主动学习的方法,利用众包工人训练出一个匹配器用来预测候选实体对的答案.然后利用一个准确率估计器估计当前匹配器的准确率,如果准确率可以达到要求,则输出该匹配器的匹配结果.否则,Corleone 会探测一些比较难的实体对让工人去再次标注.其重复以上流程直至得到满意的结果.

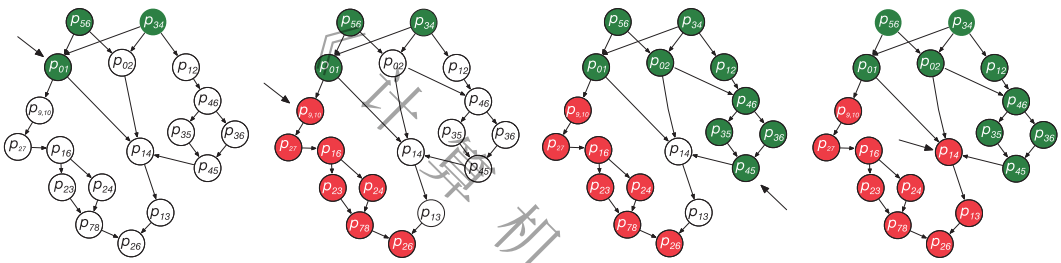

图 23　偏序图示例

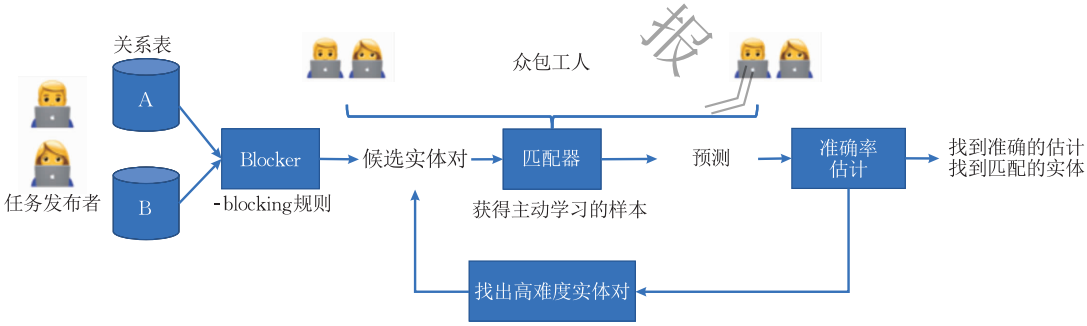

图 24　Corleone 架构图

5.3　排序/Top-k 算子

5.3.1　整体工作流程

为了获得实体的顺序或者 Top-k,我们需要产生众包任务.其中有两种众包任务被广泛采用.一个是单选题,给出两个实体,让工人选择它们两个之间的顺序关系.第二种是打分,选取多个实体组成一组,让工人对比组内实体并给每个实体打一个分.第二种方法是有明显缺陷的,首先,工人更喜欢第一种形式,因为其比较简单.其次,打分是一种很粗糙的比较,而且不同的分组之间的分数很难横向比较.因此,本文重点关注第一种方法,研究如何利用很多的成对的比较来推理出完整的顺序(见 5.3.2 节)以及如何选择最优的实体对去提问(见 5.3.3 节).

5.3.2　结果推理

5.3.2.1　基于打分的方法

证明了在图模型(如图 25)下找到一个最优的顺序或者最大值是一个 NP 难的问题,所以很多方

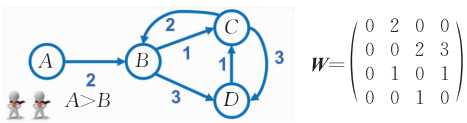

图 25　众包排序的图模型

法会在图模型的基础上给每个实体分配一个分数，然后按照分数由高到低排序.从而选择一个顺序或者 Top-k.接下来我们讨论一些分配分数的方法.

(1) BordaCount[88]. 该方法中，每个实体的分数是它在图中的出度.

(2) Copeland[90]. 该方法中，每个实体的分数是它在图中的出度减去入度.

(3) Local[19]. 以上两个方法仅仅考虑了每个节点的邻居信息，这样是不完整的.因此该方法考虑了每个节点两跳的邻居.显然，当一个点可以两跳跳到其他节点，说明它的排名很靠前.同理，如果一个点可以被其他点两跳达到，说明它的排名比较靠后.

(4) Indgree[19]. 该方法利用贝叶斯模型来计算分数.给出一个实体对，其首先利用收集的答案算出该实体对大小关系的概率分布，然后利用该分布为每个实体分配分数.

(5) MPagerank[19]. 该方法巧妙地利用了Pagerank的思想来为每个实体打分.

5.3.2.2 基于循环剪枝的方法

循环剪枝主要应用于 Top-k 的计算，其基本思想是不断自适应地移除低分数的实体，直到剩下 k 个实体.

(1) Iterative[19]. 该方法首先利用基于分数的方法移除掉一半低分数的实体.然后根据模型重新计算分数，再移除掉一半的实体，直至 k 个实体留下来.

(2) PathRank[13]. 该方法使用反向深度优先搜索遍历一个图并在其中寻找路径.如果一条路径的长度大于 k，该算法会移除该路径上后续的节点.

(3) AdaptiveReduce[90]. 给出 n 个元素，该方法选取一个信息量最丰富的子集去剪枝掉一些实体，这些实体存在于 top-k 集合的概率极小.接着其循环这一步骤直到找出 k 个元素.其中信息量可以用信息熵来衡量.

5.3.2.3 基于机器学习的方法

(1) CrowdBT[9]. 该方法可以直接估计实体的分数.假设 $o_i \succ o_j$ 的概率被表示 $\frac{e^{s_i}}{e^{s_i}+e^{s_j}}$，$s$ 表示实体的分数，那么基于获得的实体对们的答案，模型通过最大化 $\sum_{o_i \succ o_j \in L} \frac{e^{s_i}}{e^{s_i}+e^{s_j}}$ 来计算出每个实体的分数，然后根据其排序.

(2) CrowdGauss[78]. 该方法假设实体的分数服从高斯分布，分数就是高斯分布的均值.$o_i \succ o_j$ 的概率由两个标准正态分布的累积分布函数计算出来，即 $P(o_i \succ o_j)=\Phi(s_i-s_j)$. CrowdGauss 利用最大化 $\sum_{o_i \succ o_j \in L} M_{ij}\log(\Phi(s_i-s_j))$，$M_{ij}$ 是回答 $o_i \succ o_j$ 的工人的个数.

(3) HodgeRank[91]. 该方法利用矩阵分解的方式来计算实体的分数.

5.3.2.4 混合方法

有很多工作采用了打分与比较混合的方法来学习出实体的分数.对于打分算子，该方法预定义 c 个类别并且每个类别对应一段分数区间.如果一个实体的分数落在某区间内，该实体就属于那一类.下面我们介绍几种混合的方法.

(1) ScoreCombination[79]. 该方法首先选择了一些打分和比较的问题去提问，然后根据答案先推出实体的分数.该分数被建模为高斯分布，可以容忍工人的失误.对于打分算子，该方法计算了实体属于某个类别的概率.对于比较算子，该方法构建了一个比较矩阵并且根据贝叶斯理论计算观测到该矩阵的概率.最后其结合这两种算子推理出答案.

(2) RatingFirst[22]. 该方法主要应用于 top-k 的选取，其首先利用打分的方法处理所有的实体然后剪枝掉一些分数较低的实体.接下来，其选择一些实体对来进行更加细粒度的比较.同样每个实体的分数被建模成高斯分布.

(3) AdaptiveCombine[92]. 该方法由两部分组成.首先是根据当前的打分和比较的答案推理出 top-k 的结果.类似地，每个实体的分数被建模成高斯分布，然后利用答案估计高斯分布的参数并推理结果.第二步是问题选择.根据任一实体在 top-k 结果中的概率，该方法可以得到两种分布，真实的 top-k 分布和估计的 top-k 分布，该方法选择最有益于缩小两种分布的距离的问题来提问.

5.3.3 任务选择

任务选择的目的是选择哪个实体对去问工人，其方法可以被分为以下三种：

(1) 启发式方法

文献[19]证明了 top-k 的问题选择是 NP 难问题.因此其提出了多种启发式策略来选择问题.该算法首先为每个实体 o_i 打分，分数为 s_i.接着对这些实体排序 $o_1 \succ o_2 \succ \cdots \succ o_n$，最后在其中启发式地挑选 b 个实体对去问.方法如下：① 最大化策略.选择 b 对 $(o_1,o_2),(o_1,o_3),\cdots,(o_1,o_{b+1})$；② 群组策略.将相邻的靠前的实体组成一对，$(o_1,o_2),(o_3,o_4),\cdots,(o_{2b-1},o_{2b})$；③ 贪心的策略.对于每对实体计算

《计算机学报》

$s_i \cdot s_j$，然后选择 b 对该乘积最大的.

(2) 基于边界的方法

文献[90]为每一对实体估算了一个边界值并用该值选择接下来要问的实体对. 该方法首先计算了一个置信区间$[l_{ij}, u_{ij}]$，其中 l_{ij} (u_{ij}) 是 $o_i > o_j$ 的概率的下界(上界). 如果某对实体的概率不在这个区间内，说明该对实体的大小关系已经足够明显，不需要再去提问了. 因此任务发布者只需要提问那些还不是很确定的实体.

(3) 基于主动学习的方法

CrowdGauss[78]将实体的分数建模为高斯分布 $N(\boldsymbol{s}, \boldsymbol{C})$，其中 $\boldsymbol{s}$ 是一个 $1 \times n$ 的向量，表示所有实体的分数，$\boldsymbol{C}$ 是 $\boldsymbol{s}$ 的协方差矩阵. 在每轮问题选择中，CrowdGauss 都会计算每对实体的期望信息增益，然后选择增益最大的实体对去问，进而更新高斯分布的参数. 但是 CrowdGauss 没有考虑工人的质量，因此为了解决这一问题，CrowdBT[9]将工人质量建模到实体分数的计算中，提升了 top-k 计算的准确性. AdaptiveCombine[92]同时考虑了选择打分和比较问题. 给出一定预算，该方法选择最优的问题去最大化期望信息增益.

5.4　计数算子

众包计数算子可以分为两类：基于条件的计数和基数估计，两者定义如下：

定义 7(基于众包的条件计数).　利用众包在一个数据库里统计满足给定条件的记录的个数.

例如：统计某图片集合里包含猫的图片的个数.

定义 8(基于众包的基数估计).　利用众包统计某个领域内实体的个数.

例如：统计在役 NBA 球员的个数.

5.4.1　基于众包的条件计数

(1) 采样法[30]

显然，处理计数问题最暴力的方法是给出 N 个元素，我们让工人去检查每一个元素是否满足给条件. 但是该方法成本明显过高. 所以我们利用采样的方法来解决这一问题. 首先从 N 个元素采样 K 个元素，然后让人检查这 K 个元素，发现其中有 D 个元素满足条件. 因此我们可以推出大概整个数据集中有$\frac{KD}{N}$个元素满足条件.

(2) 批处理[93]

即使基于采样的方法，我们还是需要人一个一个地检查采样数据. 其实，我们可以可用人的智慧来直接处理一批数据. 如图 26 所示，我们可以利用人的认知能力，将几个元素合并起来，让人直接统计其中满足条件的元素个数.

图 26　批处理示例

5.4.2　基于众包的基数估计

基于众包的基数估计指的是让工人提供某个领域内的实体，然后根据工人的答案推理出这个领域内实体的基数. 例如，我们想知道在役 NBA 球员的个数，我们可以让工人们提供不同球员的名字，然后用统计的方法估算基数，即 Good Turing[94]算法. 该算法的主要思想是，当前收集到的实体的重复度越低，估计基数越大，反之基数越小. 该问题与收集算子有密切联系，我们会在 5.5 节继续详细讨论.

5.5　收集算子

首先我们给出收集算子的定义如下：

定义 9(基于众包的收集算子). 给定一个领域，众包收集算子的目的是利用众包工人收集该领域内所有的实体.

该算子有如下挑战：(1) 实体的完备性. 我们希望可以收集到该领域内的所有实体；(2) 少重复性. 由于工人每提供一个实体，我们就需要付相应的费用，那么工人提供的重复实体相当于浪费了成本. 因此少重复性也是一个重要挑战；(3) 正确性. 我们希望收集到的实体都是属于该领域的，因此，检测和移除错误实体也是一大挑战. 下面我们将介绍几种代表性的关于收集的工作.

(1) 成本-完备性策略

文献[95]主要考虑实体收集的完备性. 每当收集到新的实体，它需要利用 5.4.2 节的思想来估计该领域的基数. 如果收集到的实体个数与基数很相近，那么就停止收集以避免成本的浪费，所以她实现了一种成本-完备性的折中策略. 此外，它针对工人的特点对基数估计做了优化. 具体来讲，文献[95]是利用当前收集的实体的重复性来估计基数. 而基于

《计算机学报》

众包的收集会存在短时间内，某个特别优质的工人提供大量不重复实体，造成数据分布的倾斜，从而引起估计值偏大. 因此，文献[95]致力于找出这些工人并对估计值偏大的因素进行修正.

(2) 基于多人协作的收集

文献[97]主要关注于实体收集的正确性与完备性. 它主要设计了一个有趣的界面来完成实体收集. 如图 27，工人们可以在上面进行协作，大家可以检索并实时看到其他人的搜索结果，还可以利用右侧的投票给其他人的答案打分. 此外，该界面不仅支持收集一整个记录，还支持对某个属性的填充.

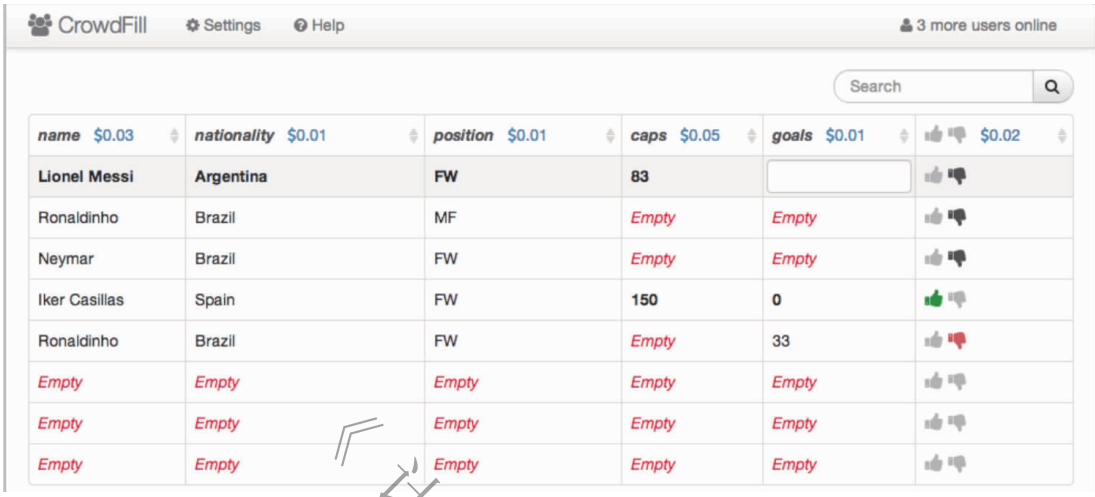

图 27 CrowdFill 界面[98]

(3) 基于激励机制的收集

然而，以上两个方法均没有考虑到少重复性，所以造成了大量成本的浪费. 而文献[97]仅仅用简单的投票来控制质量，并且该方法是被动的. 比如对于如果一个即使很明显的错误，如果没有工人愿意去揭露其错误，那么它将永远不会被发现. 因此，文献[97]设计了一个基于激励机制的框架 CrowdEC，如图 28，同时关注完备性，质量和少重复性. 他有两个主要模块，一个是工人剔除模块. 该模块通过检查工人的答案计算工人的质量，这里的质量同时考虑工人是否经常提供不重复的且正确的实体. 对于质量比较低的工人，CrowdEC 会将之剔除. 对于剩下的高质量工人，CrowdEC 会提供两种定价策略，这也是另一大模块激励机制. 一种定价策略就是普通模式，一份答案一份工钱. 另一种则是激励模式，如果工人提供了重复的答案，CrowdEC 会对之进行提醒，如果他愿意提供新的答案直至不重复，它将付给工人额外的奖励. 激励机制会决策在什么时候对哪个工人提供那种定价策略. 例如，在收集刚开始的时候，我们没必要使用激励模式，因为那时候大家提供的几乎都是不重复实体. 收集开始一段时间后，对于一些愿意接受激励机制的人，我们可以提供激励模式. 等到收集的结尾，再提供激励模式也很难收集到不重复的实体，所以我们使用普通模式即可.

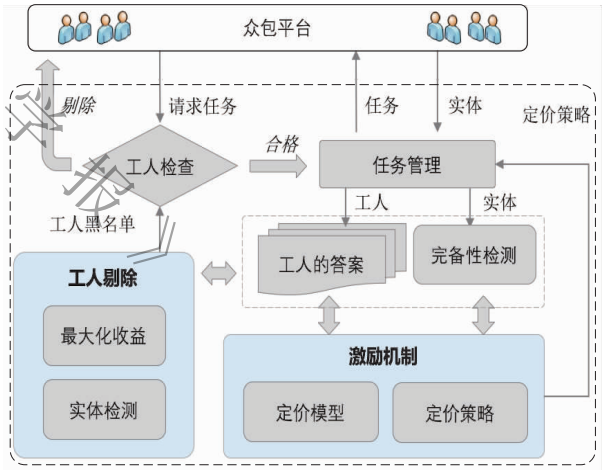

图 28 CrowdEC 框架

6 未来工作展望

在这一节中，我们将从三个角度介绍众包数据库领域的一些尚未完全解决的挑战性问题. 首先是针对众包数据库内部的算子的未来挑战，包括聚集算子、索引以及多算子的复杂查询. 接着我们会介绍众包数据库可以支持的富有挑战性的应用. 最后我们会介绍有关数据库基准的未来挑战.

6.1 众包数据库内部算子

(1) 分组(Groupby)与连接(Join). 分组与连接

是传统数据库中的重要操作.基于众包数据库的连接算子在上文中已经介绍很多了.聚集算子例如SUM、Average、Count都依赖于分组算子.具体地,分组指的是将某属性中相同的值聚合到一起,从而将数据库中的记录分成若干组.但是现实世界中数据可能会很脏,例如New York City和NY指的是相同的城市,但是传统数据库却不能将之聚合在一起.那么利用网络大众的智慧可以解决这一问题.大体来讲这两个任务可以采用实体解析的方法解决,但却存在一些挑战.首先这两种操作一般都针对存在于一个属性中的较短字符串,甚至有缩写出现.如果不采用穷举的方法,实体解析的一些剪枝策略并不好用,因为字符串太短,很难提取有用的特征.所以我们可以设计更好的策略来寻找相同实体,例如先采样一些实体进行提问,然后针对该属性学习规则,以期更有针对性地解决这一问题.

(2)索引.有关众包数据库的索引的研究目前还很基础.第一,对于众包索引的质量控制还仅仅停留在多人投票阶段,这种方法假设人们的质量都相同.但是在实际中这个假设并不一定成立,因此一些更先进的质量控制方法应该被应用进来.第二,目前研究仅支持B+树索引,我们还可以考虑比如跳表、哈希等索引.第三,目前的研究仅考虑了查询索引的质量,并没有考虑代价和延迟.由于索引的目的是快速,所以控制延迟是极其重要的.第四,现在的众包数据库系统并没有将索引模块真正实现,这也是一个系统性的挑战.

(3)多算子复杂查询.众包数据库中的优化目标与传统数据库有很大不同,包括质量、代价和延迟.现有的众包数据库系统虽然可以支持多种算子,并且在三个维度上进行优化,但是这些优化算法基本都是启发式的,缺乏有力的理论保证.虽然同时优化这三个目标并给出理论保证是不现实的(由于他们之间存在trade-off),但是我们可以给定质量要求,寻找最小代价的策略.或者给出一定预算,寻找最高质量的策略.

(4)事务(Transaction).目前,有关众包数据库的事务研究比较少,针对众包的事务定义尚未明确给出.我们认为,基于众包的事务可以存在于需要多个工人协作的任务中,例如多个工人一起填充一张表单,那么工人不能同时对一个值进行操作.再比如在众包平台上,如果一个任务只能被分配给三个工人,当超过这一数量,需要给该任务加锁.

6.2　众包数据库的应用

(1)时空众包.随着众包的普及,时空众包[98-99]也正在受到广泛关注.其主要专注于有关时空数据的任务,即要求工人在某个时间某个地点完成的任务.其组成要素也是工人、平台和任务发布者.但是不同的是,任务发布者需要发布一些时空要求,比如在哪一地点多长时间内需要完成某任务.工人也需要发布自己所在地点以及空闲时间以便与任务发布者更好地匹配.平台则负责任务分配、质量控制、激励机制、隐私控制等模块.① 任务分配指的是面对海量工人以及任务发布者(例如滴滴出行),如何分配海量任务,其优化目标包括但不限于最大化任务发布的数量[100]、最大化收益[101]、最小化工人的奔波路程[102];② 基于空间众包的质量控制与传统方法有很大不同.对工人的质量来说,时空众包需要考虑距离敏感的质量,即工人质量会随着距离的变化而变化[103].另外,对任务的质量模型来说,延迟将成为一个重要衡量指标[104];③ 激励机制可以鼓励工人更积极地加入到任务中,获得更高的质量.由于时空众包的任务是动态的,所以需要设计专门的动态激励机制来解决这一问题[105].例如在某一区域中工人如果特别少,那么工人的报酬需要增加;④ 隐私保护在时空众包中十分重要,因为其包含了客户的私人信息.在传输这些信息的时候既需要高效,又需要防止隐私的泄露[106].以上内容是近些年时空众包的研究方向.在未来,时空众包还有以下几点可以研究:首先是是否存在在线任务分配的理论保证,再者是否可以利用索引,如时空数据处理中的R-树、k-d树来加速时空众包算法的执行效率.最后是现实中虽然存在很多有关时空数据库的研究,但是尚未有将时空众包和数据库真正结合,提供查询语言,优化策略,与平台交互的统一系统.所以这也是一大研究挑战.

(2)基于众包数据库的可视化.数据可视化也是近些年的研究热点.其重点研究如何将大量且分布不同的数据更好地可视化出来,以期让用户更加了解数据.再者,数据可视化还研究如何利用可视化使得数据的处理更高效.这两点都离不开人的参与,例如,人可以标注对可视化结果的满意程度,作为反馈调整可视化策略.此外,有时候数据科学家需要做可视化的逆过程,即将图表转换为数据,这个过程利用机器来做是极其不准确的.所以人在其中也可以发挥重要的作用.由于被可视化的数据大部分为存储在数据库中的结构化数据,那么如何定义可视化查询语言,如何高质量高效率地进行可视化是一个研究挑战.

(3)众包数据库与宏观问题.本文介绍的研究内容基本都是围绕微观任务展开的.然而,实际中很多众包任务是宏观任务,例如翻译一篇文章.宏观任务的挑战在于其很难被分解为微观任务的,因为这样会丢失很多上下文信息[107].另外,工人很可能不愿意做宏观任务因为其难度高耗时长.因此,针对不同的宏观任务,如何分解为较为容易并且不影响整体答题质量的微观任务是一大挑战.例如文本数据也可以存储在数据库中,如何定义查询语言来分解任务,如何优化代价提升质量是值得研究的问题.

(4)众包数据库中的异常点检测.异常点检测是数据清洗的一个重要分支,例如在金融数据中其可以检测出那些诈骗的账户.因此我们可以利用众包来生成欺诈检测的规则.例如在凌晨某些经常发生诈骗的地区有大数额的交易记录.我们可以利用众包生成欺诈检测规则,但是人并不是专家,容易出错.我们还可以通过众包来检验生成的规则.这样两种工人做不同的任务,互相促进提高准确率.另外我们还可以加入机器算法,引入先验知识,从而减轻人的负担,降低成本开销.

(5)众包数据库与机器学习.在数据库中,数据量的增长是十分迅速的.相比于机器资源,人力是十分昂贵的.因此,面对海量数据,很多时候仅仅基于机器算法的剪枝,众包技术仍需花费极高的成本.因此,我们需要考虑众包技术与机器学习的结合,例如主动学习,半监督学习,监督学习等方法相结合来解决大数据问题.

6.3 众包数据库的基准(Benchmark)

基准(Benchmark).在传统数据库领域,有很多TPC基准用来测试数据库各个方面的性能,推动了数据库领域的发展.然而,虽然众包领域有很多公共数据集,但是仍然缺乏一个公认的基准.所以,为众包数据库系统建立完备的评估手段和准则是一个研究挑战.首先,我们应该建立多个针对不同题型,如单选、多选、填空等的数据集,并提供众包答案.这些答案应该是从众包平台如AMT收集而来,每道题应至少让十个工人回答,这样使得应用该基准的人可以灵活地选择回答每道题的工人的个数.

参 考 文 献

[1] Wang J, Kraska T, Franklin M J, et al. CrowdER: Crowdsourcing entity resolution. Proceedings of the VLDB Endowment, 2012, 5(11): 1483-1494

[2] Vander Schee B A. Crowdsourcing: why the power of the crowd is driving the future of business. Journal of Consumer Marketing, 2010: 1565-1566

[3] Von Ahn L, Dabbish L. ESP: Labeling Images with a Computer Game//Proceedings of the AAAI Spring Symposium: Knowledge Collection from Volunteer Contributors. Stanford, USA, 2005: 91-98

[4] Amsterdamer Y, Davidson S B, Kukliansky A, et al. Managing general and individual knowledge in crowd mining applications//Proceedings of the Conference on Innovative Data Systems Research. Asilomar, USA, 2015

[5] Amsterdamer Y, Davidson S B, Milo T, et al. OASSIS: Query driven crowd mining//Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. Snowbird, USA, 2014: 589-600

[6] Amsterdamer Y, Davidson S B, Milo T, et al. Ontology assisted crowd mining. Proceedings of the VLDB Endowment, 2014, 7(13): 1597-1600

[7] Amsterdamer Y, Grossman Y, Milo T, et al. Crowd mining //Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data. New York, USA, 2013: 241-252

[8] Amsterdamer Y, Grossman Y, Milo T, et al. CrowdMiner: Mining association rules from the crowd. Proceedings of the VLDB Endowment, 2013, 6(12): 1250-1253

[9] Chen X, Bennett P N, Collins-Thompson K, et al. Pairwise ranking aggregation in a crowdsourced setting//Proceedings of the 6th ACM International Conference on Web Search and Data Mining. Rome, Italy, 2013: 193-202

[10] Davidson S B, Khanna S, Milo T, et al. Using the crowd for top-k and group-by queries//Proceedings of the 16th International Conference on Database Theory. Genoa, Italy, 2013: 225-236

[11] Deng J, Dong W, Socher R, et al. ImageNet: A large-scale hierarchical image database//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Miami, USA, 2009: 248-255

[12] Eiben C B, Siegel J B, Bale J B, et al. Increased Diels-Alderase activity through backbone remodeling guided by foldit players. Nature Biotechnology, 2012, 30(2): 190

[13] Eriksson B. Learning to top-k search using pairwise comparisons //Proceedings of the Artificial Intelligence and Statistics. Scottsdale, USA, 2013: 265-273

[14] Fan J, Lu M, Ooi B C, et al. A hybrid machine-crowdsourcing system for matching web tables//Proceedings of the IEEE 30th International Conference on Data Engineering. Chicago, USA, 2014: 976-987

[15] Fan J, Zhang M, Kok S, et al. CrowdOp: Query optimization for declarative crowdsourcing systems. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(8): 2078-2092

[16] Fang Y, Sun H, Li G, et al. Effective result inference for context-sensitive tasks in crowdsourcing//Proceedings of the International Conference on Database Systems for Advanced Applications. Springer, Cham, 2016: 33-48

[17] Franklin M J, Kossmann D, Kraska T, et al. CrowdDB: Answering queries with crowdsourcing//Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data. Athens, Greece, 2011: 61-72

[18] Groz B, Milo T. Skyline queries with noisy comparisons//Proceedings of the 34th ACM Symposium on Principles of Database Systems. Melbourne, Australia, 2015: 185-198

[19] Guo S, Parameswaran A, Garcia-Molina H. So who won?: dynamic max discovery with the crowd//Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. Scottsdale, USA, 2012: 385-396

[20] Ipeirotis P G, Provost F, Wang J. Quality management on Amazon mechanical Turk//Proceedings of the ACM SIGKDD Workshop on Human Computation. Washington, USA, 2010: 64-67

[21] Kaplan H, Lotosh I, Milo T, et al. Answering planning queries with the crowd. Proceedings of the VLDB Endowment, 2013, 6(9): 697-708

[22] Khan A R, Garcia-Molina H. Hybrid Strategies for Finding the Max with the Crowd. Stanford University, Technical Report, 2014

[23] Li G. Human-in-the-loop data integration. Proceedings of the VLDB Endowment, 2017, 10(12): 2006-2017

[24] Li G, Chai C, Fan J, et al. CDB: Optimizing queries with crowd-based selections and joins//Proceedings of the 2017 ACM International Conference on Management of Data. Chicago, USA, 2017: 1463-1478

[25] Li G, Wang J, Zheng Y, et al. Crowdsourced data management: A survey. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(9): 2296-2319

[26] Liu Q, Peng J, Ihler A T. Variational inference for crowdsourcing//Proceedings of the Advances in Neural Information Processing Systems. Lake Tahoe, USA, 2012: 692-700

[27] Liu X, Lu M, Ooi B C, et al. CDAS: A crowdsourcing data analytics system. Proceedings of the VLDB Endowment, 2012, 5(10): 1040-1051

[28] Lofi C, El Maarry K, Balke W T. Skyline queries in crowd-enabled databases//Proceedings of the 16th International Conference on Extending Database Technology. Genoa, Italy, 2013: 465-476

[29] Lofi C, El Maarry K, Balke W T. Skyline queries over incomplete data-error models for focused crowd-sourcing//Proceedings of the International Conference on Conceptual Modeling. Berlin, Germany: Springer, 2013: 298-312

[30] Marcus A, Karger D, Madden S, et al. Counting with the crowd. Proceedings of the VLDB Endowment, 2012, 6(2): 109-120

[31] Marcus A, Wu E, Karger D R, et al. Crowdsourced databases: Query processing with people//Proceedings of the Conference on Innovative Data Systems Research. Asilomar, USA, 2011: 211-214

[32] Mozafari B, Sarkar P, Franklin M, et al. Scaling up crowd-sourcing to very large datasets: a case for active learning. Proceedings of the VLDB Endowment, 2014, 8(2): 125-136

[33] Nguyen Q V H, Nguyen T T, Miklós Z, et al. Pay-as-you-go reconciliation in schema matching networks//Proceedings of the 2014 IEEE 30th International Conference on Data Engineering. Chicago, USA, 2014: 220-231

[34] Parameswaran A, Boyd S, Garcia-Molina H, et al. Optimal crowd-powered rating and filtering algorithms. Proceedings of the VLDB Endowment, 2014, 7(9): 685-696

[35] Parameswaran A G, Garcia-Molina H, Park H, et al. Crowdscreen: Algorithms for filtering data with humans//Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. Scottsdale, USA, 2012: 361-372

[36] Parameswaran A, Sarma A D, Garcia-Molina H, et al. Human-assisted graph search: It's okay to ask questions. Proceedings of the VLDB Endowment, 2011, 4(5): 267-278

[37] Park H, Garcia-Molina H, Pang R, et al. Deco: A system for declarative crowdsourcing. Proceedings of the VLDB Endowment, 2012, 5(12): 1990-1993

[38] Amer-Yahia S, Roy S B. Human factors in crowdsourcing. Proceedings of the VLDB Endowment, 2016, 9(13): 1615-1618

[39] Gao J, Li Q, Zhao B, et al. Truth discovery and crowdsourcing aggregation: A unified perspective. Proceedings of the VLDB Endowment, 2015, 8(12): 2048-2049

[40] Chen L, Lee D, Milo T. Data-driven crowdsourcing: Management, mining, and applications//Proceedings of the 2015 IEEE 31st International Conference on Data Engineering. Seoul, South Korea, 2015: 1527-1529

[41] Doan A, Franklin M J, Kossmann D, et al. Crowdsourcing applications and platforms: A data management perspective. Proceedings of the VLDB Endowment, 2011, 4(12): 1508-1509

[42] Marcus A, Parameswaran A. Crowdsourced data management: Industry and academic perspectives. Foundations and Trends in Databases, 2015, 6(1-2): 1-161

[43] Feng Jian-Hong, Li Guo-Liang, Feng Jian-Hua. A survey on crowdsourcing. Chinese Journal Of Computers, 2015, 38(9): 1713-1726(in Chinese)
(冯建红，李国良，冯建华. 众包技术研究综述. 计算机学报，2015, 38(9): 1713-1726)

[44] Faradani S, Hartmann B, Ipeirotis P G. What's the right price? pricing tasks for finishing on time//Proceedings of the Workshops at the 25th AAAI Conference on Artificial Intelligence. 2011

[45] Cao C C, She J, Tong Y, et al. Whom to ask?: Jury selection for decision making tasks on micro-blog services. Proceedings of the VLDB Endowment, 2012, 5(11): 1495-1506

[46] Kazemi L, Shahabi C, Chen L. GeoTruCrowd: Trustworthy query answering with spatial crowdsourcing//Proceedings of the 21st ACM SIGSPATIAL International Conference on

《计算机学报》

Advances in Geographic Information Systems. Orlando, USA, 2013: 314-323

[47] Zhang C J, Chen L, Jagadish H V, et al. Reducing uncertainty of schema matching via crowdsourcing. Proceedings of the VLDB Endowment, 2013, 6(9): 757-768

[48] Whitehill J, Wu T, Bergsma J, et al. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise//Proceedings of the Advances in Neural Information Processing Systems. Vancouver, Canada, 2009: 2035-2043

[49] Aydin B I, Yilmaz Y S, Li Y, et al. Crowdsourcing for multiple-choice question answering//Proceedings of the 28th AAAI Conference on Artificial Intelligence. Quebec, Canada, 2014: 2946-2953

[50] Karger D R, Oh S, Shah D. Iterative learning for reliable crowdsourcing systems//Proceedings of the Advances in Neural Information Processing Systems. Granada, Spain, 2011: 1953-1961

[51] Joglekar M, Garcia-Molina H, Parameswaran A. Evaluating the crowd with confidence//Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Chicago, USA, 2013: 686-694

[52] Dawid A P, Skene A M. Maximum likelihood estimation of observer error-rates using the EM algorithm. Journal of the Royal Statistical Society: Series C (Applied Statistics), 1979, 28(1): 20-28

[53] Raykar V C, Yu S, Zhao L H, et al. Supervised learning from multiple experts: Whom to trust when everyone lies a bit//Proceedings of the 26th Annual International Conference on Machine Learning. Quebec, Italy, 2009: 889-896

[54] Venanzi M, Guiver J, Kazai G, et al. Community-based Bayesian aggregation models for crowdsourcing//Proceedings of the 23rd International Conference on World Wide Web. Seoul, South Korea, 2014: 155-164

[55] Zheng Y, Wang J, Li G, et al. QASCA: A quality-aware task assignment system for crowdsourcing applications//Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. Melbourne, Australia, 2015: 1031-1046

[56] Marcus A, Wu E, Karger D, et al. Human-powered sorts and joins. Proceedings of the VLDB Endowment, 2011, 5(1): 13-24

[57] Feng J, Li G, Wang H, et al. Incremental quality inference in crowdsourcing//Proceedings of the International Conference on Database Systems for Advanced Applications. Springer, Cham, 2014: 453-467

[58] Kurve A, Miller D J, Kesidis G. Multicategory crowdsourcing accounting for variable task difficulty, worker skill, and worker intention. IEEE Transactions on Knowledge and Data Engineering, 2014, 27(3): 794-809

[59] Smyth P, Fayyad U M, Burl M C, et al. Inferring ground truth from subjective labelling of Venus images//Proceedings of the Advances in Neural Information Processing Systems. Denver, USA, 1995: 1085-1092

[60] Dai P, Lin C H, Weld D S. POMDP-based control of workflows for crowdsourcing. Artificial Intelligence, 2013, 20(2): 52-85

[61] Demartini G, Difallah D E, Cudré-Mauroux P. ZenCrowd: Leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking//Proceedings of the 21st International Conference on World Wide Web. Lyon, France, 2012: 469-478

[62] Zhao Z, Yan D, Ng W, et al. A transfer learning based framework of crowd-selection on Twitter//Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Chicago, USA, 2013: 1514-1517

[63] Koller D, Friedman N. Probabilistic Graphical Models: Principles and Techniques. USA: MIT Press, 2009

[64] Kuncheva L I, Whitaker C J, Shipp C A, et al. Limits on the majority vote accuracy in classifier fusion. Pattern Analysis & Applications, 2003, 6(1): 22-31

[65] Littlestone N, Warmuth M K. The weighted majority algorithm. Information and Computation, 1994, 108(2): 212-261

[66] Ho C J, Jabbari S, Vaughan J W. Adaptive task assignment for crowdsourced classification//Proceedings of the International Conference on Machine Learning. Atlanta, USA, 2013: 534-542

[67] Boim R, Greenshpan O, Milo T, et al. Asking the right questions in crowd data sourcing//Proceedings of the 2012 IEEE 28th International Conference on Data Engineering. Washington, USA, 2012: 1261-1264

[68] Li G, Zheng Y, Fan J, et al. Crowdsourced data management: Overview and challenges//Proceedings of the 2017 ACM International Conference on Management of Data. Chicago, USA, 2017: 1711-1716

[69] Fan J, Li G, Ooi B C, et al. iCrowd: An adaptive crowdsourcing framework//Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. Melbourne, Australia, 2015: 1015-1030

[70] Ho C J, Vaughan J W. Online task assignment in crowdsourcing markets//Proceedings of the 26th AAAI Conference on Artificial Intelligence. Toronto, Canada, 2012

[71] Basu Roy S, Lykourentzou I, Thirumuruganathan S, et al. Task assignment optimization in knowledge-intensive crowdsourcing. The VLDB Journal—The International Journal on Very Large Data Bases, 2015, 24(4): 467-491

[72] Wang J, Li G, Kraska T, et al. Leveraging transitive relations for crowdsourced joins//Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data. New York, USA, 2013: 229-240

[73] Wang S, Xiao X, Lee C H. Crowd-based deduplication: An adaptive approach//Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. Melbourne, Australia, 2015: 1263-1277

[74] Whang S E, Lofgren P, Garcia-Molina H. Question selection for crowd entity resolution. Proceedings of the VLDB Endowment, 2013, 6(6): 349-360

[75] Vesdapunt N, Bellare K, Dalvi N. Crowdsourcing algorithms for entity resolution. Proceedings of the VLDB Endowment, 2014, 7(12): 1071-1082

[76] Yan T, Kumar V, Ganesan D. Crowdsearch: Exploiting crowds for accurate real-time image search on mobile phones//Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services. San Francisco, USA, 2010: 77-90

[77] Wang J, Li G, Feng J. Can we beat the prefix filtering?: An adaptive framework for similarity join and search//Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. Scottsdale, USA, 2012: 85-96

[78] Pfeiffer T, Gao X A, Chen Y, et al. Adaptive polling for information aggregation//Proceedings of the 26th AAAI Conference on Artificial Intelligence. Toronto, Canad, 2012

[79] Ye P, Doermann D. Combining preference and absolute judgements in a crowd-sourced setting//Proceedings of the International Conference on Machine Learning. Atlanta, USA, 2013

[80] Gruenheid A, Kossmann D, Sukriti R, et al. Crowdsourcing entity resolution: When is a=b?. Technical Report. ETH Zurich, 2012

[81] Zhang C J, Tong Y, Chen L. Where to: Crowd-aided path selection. Proceedings of the VLDB Endowment, 2014, 7(14): 2005-2016

[82] Wang J, Krishnan S, Franklin M J, et al. A sample-and-clean framework for fast and accurate query processing on dirty data//Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. Snowbird, USA, 2014: 469-480

[83] Sarma A D, Parameswaran A, Garcia-Molina H, et al. Crowd-powered find algorithms//Proceedings of the 2014 IEEE 30th International Conference on Data Engineering. Chicago, USA, 2014: 964-975

[84] Verroios V, Lofgren P, Garcia-Molina H. tDP: An optimal-latency budget allocation strategy for crowdsourced MAXIMUM operations//Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. Melbourne, Australia, 2015: 1047-1062

[85] Mahmood A R, Aref W G, Dragut E, et al. The Palm-tree Index: Indexing with the crowd//Proceedings of the 1st VLDB Workshop on Databases and Crowdsourcing DBCrowd 2013. Riva del Garda, Italy, 2016: 26-31

[86] Chai C, Li G, Li J, et al. Cost-effective crowdsourced entity resolution: A partial-order approach//Proceedings of the 2016 International Conference on Management of Data. San Francisco, USA, 2016: 969-984

[87] Gokhale C, Das S, Doan A H, et al. Corleone: Hands-off crowdsourcing for entity matching//Proceedings of the 2014 ACM SIGMOD international conference on Management of data. SNOWBIRD, USA, 2014: 601-612

[88] Adelsman R M, Whinston A B. Sophisticated voting with information for two voting functions. Journal of Economic Theory, 1977, 15(1): 145-159

[89] Pomerol J C, Barba-Romero S. Multicriterion Decision in Management: Principles and Practice. Springer Science & Business Media, 2012

[90] Busa-Fekete R, Szorenyi B, Cheng W, et al. Top-k selection based on adaptive sampling of noisy preferences//Proceedings of the International Conference on Machine Learning. Atlanta, USA, 2013: 1094-1102

[91] Jiang X, Lim L H, Yao Y, et al. Statistical ranking and combinatorial Hodge theory. Mathematical Programming, 2011, 127(1): 203-244

[92] Li K, Li X Z G, Feng J. A rating-ranking based framework for crowdsourced top-k computation//Proceedings of the 2018 International Conference on Management of Data. Houston, USA, 2018

[93] Sarma A D, Jain A, Nandi A, et al. JellyBean: Crowd-powered image counting algorithms. HCOMP, 2015

[94] Good I J. The population frequencies of species and the estimation of population parameters. Biometrika, 1953, 40(3-4): 237-264

[95] Trushkowsky B, Kraska T, Franklin M J, et al. Crowdsourced enumeration queries//Proceedings of the 2013 IEEE 29th International Conference on Data Engineering. Brisbane, Australia, 2013: 673-684

[96] Park H, Widom J. CrowdFill: Collecting structured data from the crowd//Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. Snowbird, USA, 2014: 577-588

[97] Chai C, Fan J, Li G. Incentive-based entity collection using crowdsourcing//Proceedings of the 2018 IEEE 34th International Conference on Data Engineering. Paris, France, 2018: 341-352

[98] Tong Y, Zhou Z, Zeng Y, et al. Spatial crowdsourcing: A survey. The VLDB Journal, 2019, 29(1): 217-250

[99] Tong Y, Chen L, Shahabi C. Spatial crowdsourcing: Challenges, techniques, and applications. Proceedings of the VLDB Endowment, 2017, 10(12): 1988-1991

[100] Deng D, Shahabi C, Demiryurek U. Maximizing the number of worker's self-selected tasks in spatial crowdsourcing//Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. Orlando, USA, 2013: 324-333

[101] Zhao B, Xu P, Shi Y, et al. Preference-aware task assignment in on-demand Taxi dispatching: An online stable matching approach//Proceedings of the AAAI Conference on Artificial Intelligence. Honolulum, USA, 2019: 2245-2252

[102] Tong Y, She J, Ding B, et al. Online mobile micro-task allocation in spatial crowdsourcing//Proceedings of the 2016 IEEE 32nd International Conference on Data Engineering. Helsinki, Finland, 2016: 49-60

《计算机学报》

[103] Hu H, Zheng Y, Bao Z, et al. Crowdsourced POI labelling: Location-aware result inference and task assignment// Proceedings of the 2016 IEEE 32nd International Conference on Data Engineering. Helsinki, Finland, 2016: 61-72

[104] Zeng Y, Tong Y, Chen L, et al. Latency-oriented task completion via spatial crowdsourcing//Proceedings of the 2018 IEEE 34th International Conference on Data Engineering. Paris, France, 2018: 317-328

[105] Tong Y, Wang L, Zhou Z, et al. Dynamic pricing in spatial crowdsourcing: A matching-based approach//Proceedings of the 2018 International Conference on Management of Data. Houston, USA, 2018: 773-788

[106] Kazemi L, Shahabi C. A privacy-aware framework for participatory sensing. ACM SIGKDD Explorations Newsletter, 2011, 13(1): 43-51

[107] Haas D, Ansel J, Gu L, et al. Argonaut: Macrotask crowdsourcing for complex data processing. Proceedings of the VLDB Endowment, 2015, 8(12): 1642-1653

CHAI Cheng-Liang, Ph. D. candidate. His research interests mainly include data cleaning and integration, and crowdsourcing data management.

LI Guo-Liang, Ph. D., professor. His research interests mainly include data cleaning and integration, and crowdsourcing data management.

ZHAO Tian-Yu, Ph. D. candidate. His research interests include spatial databases and crowdsourcing.

LUO Yu-Yu, M. S. candidate. His research interests include data cleaning and integration, and data visualization.

YU Ming-He, Ph. D. Her research interests include information retrieval and big data.

《计算机学报》

Background

Recently, crowdsourcing has been a sophisticated idea to help user to handle massive computer-hard tasks, such as entity resolution, image classification, etc. Thanks to crowdsourcing platforms like AMT, the requester can hire hundreds of thousands of workers to answer the tasks. However, it is rather difficult for the requester to interact with the crowdsourcing platform. Inspired by the traditional DBMS, researchers have developed the crowdsourcing database systems to encapsulate the complexity of interacting with the crowd. Based on it, the requester can process the data using the crowd with declarative languages like SQL easily. There are several significant and fundamental challenges with respect to the design of crowdsourcing database system. Firstly, the human efforts, i. e., the cost should be saved. Since the crowd is not free, we should pay them once a worker provides an answer. Therefore, some strategies should be adopted to control the cost. Current research mainly focuses on machine-based pruning and answer reasoning. The former one aims to prune some unnecessarily asked tasks based on some machine-based computation. The latter one leverages some inherent properties of the task to deduce the answer according to the obtained answers, like the transitivity property in entity resolution tasks. Secondly, since the crowd may make mistakes, we should control the quality. The general approach is to generate redundant questions corresponding to a single task and aggregate the answers to infer the truth. Thirdly, we should control the latency because workers answer much more slowly than the computer. The commonly used method is to ask the workers answer in parallel.