Crowdsourced Data Management: A Survey

(Extended Abstract)

Guoliang Li, Jiannan Wang, Yudian Zheng, Michael Franklin
Tsinghua University, Simon Fraser University, The University of Hong Kong, University of Chicago
liguoliang@tsinghua.edu.cn, jnwang@sfu.ca, ydzheng2@cs.hku.hk, mjfranklin@cs.uchicago.edu

Abstract—Many important data management and analytics tasks cannot be completely addressed by automated processes. These tasks, such as entity resolution, sentiment analysis, and image recognition can be enhanced through the use of human cognitive ability. Crowdsourcing is an effective way to harness the capabilities of people (i.e., the crowd) to apply human computation for such tasks. Thus, crowdsourced data management has become an area of increasing interest in research and industry.

We identify three important problems in crowdsourced data management. (1) Quality Control: Workers may return noisy or incorrect results so effective techniques are required to achieve high quality; (2) Cost Control: The crowd is not free, and cost control aims to reduce the monetary cost; (3) Latency Control: The human workers can be slow, particularly compared to automated computing time scales, so latency-control techniques are required.

Given a task (e.g., entity resolution), task design aims to design effective task types (e.g., devising a YES/NO question and asking workers to select an answer). Task design also needs to set the properties of tasks, e.g., deciding prices, setting time constraint, and choosing quality-control methods. The first is pricing. Usually a higher price attracts more workers. If the requester has a time constraint it is important to control latency. There are several strategies to improve quality. We can eliminate the low-quality workers (called worker elimination), assign a task to multiple workers and aggregate their answers (called answer aggregation) [9], [26], [27], or assign tasks to appropriate workers (called task assignment) [30].

(2) Cost Control. The crowd is not free, and if there are large numbers of tasks, crowdsourcing can be expensive. There are several effective cost-control techniques. The first is pruning, which first uses computer algorithms to remove some unnecessary tasks and then utilizes the crowd to answer only the necessary tasks. The second is task selection, which prioritizes which tasks to crowdsource. The third is answer deduction, which crowdsources a subset of tasks and based on the answers collected from the crowd, deduces the results of other tasks. The fourth is sampling, which samples a subset of tasks to crowdsource.

(3) Latency Control. Crowd answers may incur excessive latency for several reasons: for example, workers may be distracted or unavailable, the tasks may not be appealing to enough workers, or the tasks might be difficult for most workers. If the requester has a time constraint it is important to control latency. There are several strategies to control latency. The first is pricing. Usually a higher price attracts more workers and can reduce the latency. The second is latency modeling [21]. There are mainly two latency models: the round model and the statistical model. (a) The round model leverages the idea that tasks can be published in multiple rounds. If there are enough active workers on the crowdsourcing platform, the latency of answering tasks in each round can be regarded as constant time. Thus the overall latency is modeled as the number of rounds. (b) The statistical model is also used to model latency, which leverages the collected statistics from previous crowdsourcing tasks to build statistical models that can capture the workers’ arrival time, the completion time, etc. These derived models can then be used to predict and perhaps adjust for expected latency.

Task Design. Given a task (e.g., entity resolution), task design aims to design effective task types (e.g., devising a YES/NO question and asking workers to select an answer). Task design also needs to set the properties of tasks, e.g., deciding prices, setting time constraint, and choosing quality-control methods.

Crowdsourced Operator And Optimization. A crowdsourcing system can provide specialized operators for certain purposes (Table 1). For example, entity resolution can use a crowdsourced join to find objects referring to the same entity. In data extraction, we need to use crowdsourced selection to select relevant data. In subjective comparison scenarios we need to use crowdsourced sort to rank the results. Many operator-specific techniques have been proposed to optimize cost, quality, or latency in crowdsourcing environments.
ACKNOWLEDGMENT

This work was supported by 973 Program of China (2015CB358700), NSF of China (61632016, 61422205, 61373024, 6166116012), Shenzhen, Tencent, FDCT/116/2013/A3, MYRG105 (Y1-L3)-FST13-GZ.

REFERENCES


