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. Crowdsouring 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. There has been significant work addressing these three factors for designing crowdsourced tasks, developing crowdsourced data manipulation operators, and optimizing plans consisting of multiple operators. We survey and synthesize a wide spectrum of existing studies on crowdsourced data management.

I. Introduction

Existing algorithms cannot effectively address computer-hard tasks such as entity resolution [21], and image recognition [22]. Crowdsourcing is an effective way to address such tasks by utilizing hundreds of thousands of ordinary workers (i.e., the crowd). Consider entity resolution as an example. Suppose a user (called the "requester") has a set of objects and wants to find the objects that refer to the same entity, perhaps using different names. Although this problem has been studied for decades, traditional algorithms are still far from perfect [20]. Alternatively, s/he can harness the crowd's ability to identify the same entity. To this end, the requester first designs the tasks (e.g., a task for every pair of objects that asks workers to indicate whether the two objects refer to the same entity). Then the requester publishes their tasks on a crowdsourcing platform. Workers who are willing to perform such tasks (typically for pay or some other reward) accept the tasks, answer them and submit the answers back to the platform. The platform collects the answers and reports them to the requester. There are several important problems in crowdsourced data management as shown in Figure 1.

(1) Quality Control. Crowdsourcing may yield relatively low-quality results or even noise. For example, a malicious worker may intentionally give wrong answers. Workers may have different levels of expertise, and an untrained worker may be incapable of accomplishing certain tasks. To achieve high quality, we need to tolerate crowd errors and infer high-quality results from noisy answers. The first step of quality control is to characterize a worker's quality (called worker modeling) [30], [28], [29], [5]. Then based on the quality model of workers, 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 for latency control. 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 I). 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.

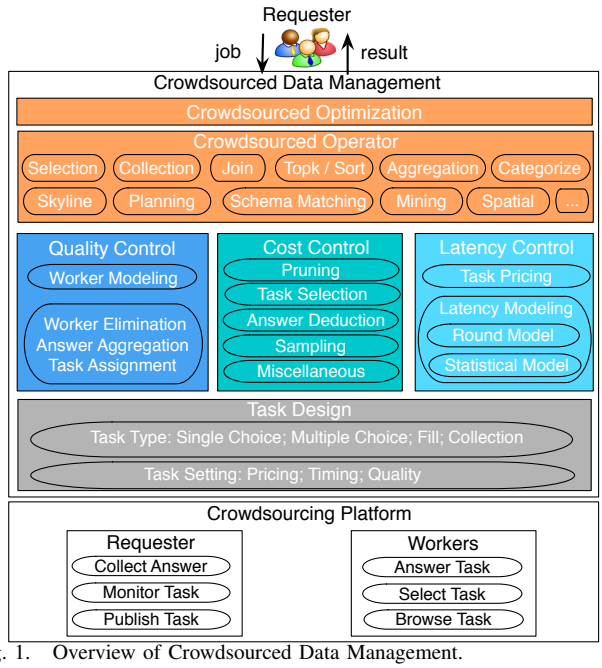

Fig. 1. Overview of Crowdsourced Data Management.

ACKNOWLEDGMENT

This work was supported by 973 Program of China (2015CB358700), NSF of China (61632016, 61472198, 61422205, 61373024, 61661166012), Shenzhou, Tencent, FDCT/116/2013/A3, MYRG105 (Y1-L3)-FST13-GZ.

REFERENCES

[1] Y. Amsterdamer, S. B. Davidson, T. Milo, S. Novgorodov, and A. Somech. Oassis: query driven crowd mining. In *SIGMOD*, pages 589–600. ACM, 2014.
[2] Y. Amsterdamer, Y. Grossman, T. Milo, and P. Senellart. Crowd mining. In *SIGMOD*, pages 241–252. ACM, 2013.
[3] X. Chen, P. N. Bennett, K. Collins-Thompson, and E. Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *WSDM*, pages 193–202, 2013.
[4] S. B. Davidson, S. Khanna, T. Milo, and S. Roy. Using the crowd for top-k and group-by queries. In *ICDT*, pages 225–236, 2013.
[5] J. Fan, G. Li, B. C. Ooi, K. Tan, and J. Feng. icrowd: An adaptive crowdsourcing framework. In *SIGMOD*, pages 1015–1030, 2015.
[6] B. Groz and T. Milo. Skyline queries with noisy comparisons. In *PODS*, pages 185–198, 2015.
[7] S. Guo, A. G. Parameswaran, and H. Garcia-Molina. So who won?: dynamic max discovery with the crowd. In *SIGMOD*.
[8] H. Heikinheimo and A. Ukkonen. The crowd-median algorithm. In *HCOMP*, 2013.
[9] H. Hu, Y. Zheng, Z. Bao, G. Li, J. Feng, and R. Cheng. Crowdsourced poi labelling: Location-aware result inference and task assignment. In *ICDE*, pages 61–72, 2016.
[10] H. Kaplan, I. Lotosh, T. Milo, and S. Novgorodov. Answering planning queries with the crowd. *PVLDB*, 6(9):697–708, 2013.
[11] C. Lofi, K. E. Maarry, and W. Balke. Skyline queries in crowd-enabled databases. In *EDBT*, pages 465–476, 2013.
[12] A. Marcus, D. R. Karger, S. Madden, R. Miller, and S. Oh. Counting with the crowd. *PVLDB*, 6(2):109–120, 2012.
[13] A. G. Parameswaran, H. Garcia-Molina, H. Park, N. Polyzotis, A. Ramesh, and J. Widom. Crowdscreen: algorithms for filtering data with humans. In *SIGMOD*, pages 361–372, 2012.
[14] A. G. Parameswaran, A. D. Sarma, H. Garcia-Molina, N. Polyzotis, and J. Widom. Human-assisted graph search: it's okay to ask questions. *PVLDB*, 4(5):267–278, 2011.
[15] H. Park and J. Widom. Crowdfill: collecting structured data from the crowd. In *SIGMOD*, pages 577–588, 2014.
[16] A. D. Sarma, A. G. Parameswaran, H. Garcia-Molina, and A. Y. Halevy. Crowd-powered find algorithms. In *ICDE*, pages 964–975, 2014.
[17] H. Su, K. Zheng, J. Huang, H. Jeung, L. Chen, and X. Zhou. Crowdplanner: A crowd-based route recommendation system. In *ICDE*, pages 1144–1155. IEEE, 2014.
[18] H. To, G. Ghinita, and C. Shahabi. A framework for protecting worker location privacy in spatial crowdsourcing. *PVLDB*, 7(10):919–930, 2014.
[19] B. Trushkowsky, T. Kraska, M. J. Franklin, and P. Sarkar. Crowdsourced enumeration queries. In *ICDE*, pages 673–684, 2013.
[20] J. Wang, T. Kraska, M. J. Franklin, and J. Feng. CrowdER: crowdsourcing entity resolution. *PVLDB*, 5(11):1483–1494, 2012.

TABLE I
CROWDSOURCED OPERATORS

Operators		Goal	Techniques
Selection	Filtering [13]	Quality	Truth Inference, Task Assignment
		Cost	Task Selection
	Find [16]	Quality	Truth Inference, Task Assignment
		Cost	Task Selection
		Latency	Round Model
	Search [22]	Quality	Truth Inference, Task Assignment
		Cost	Task Selection
		Latency	Statistical model
Collection	Enumeration [19]	Quality	Truth Inference
		Cost	Miscellaneous
	Fill [15]	Quality	Truth Inference
		Cost	Miscellaneous
		Latency	Pricing
Join	CrowdER [20]	Quality	Worker Elimination, Truth Inference
		Cost	Pruning, Miscellaneous
	Transitivity [21]	Quality	Truth Inference, Task Assignment
		Cost	Pruning, Answer Deduction
Topk/Sort	Heuristics [7]	Quality	Truth Inference, Task Assignment
		Cost	Task Selection
	ML [3]	Quality	Truth Inference, Task Assignment
		Cost	Task Selection
	Reduce [7]	Quality	Truth Inference, Task Assignment
		Cost	Task Selection, Answer Deduction
	Heap [4]	Quality	Truth Inference, Task Assignment
		Cost	Task Selection, Answer Deduction
	Hybrid [23]	Quality	Truth Inference, Task Assignment
		Cost	Task Selection
Categorize	[14]	Quality	Truth Inference, Task Assignment
		Cost	Task Selection
Aggregation	Max [7]	Quality	Truth Inference, Task Assignment
		Cost	Task Selection, Answer Deduction
	Count [12]	Quality	Worker Elimination, Truth Inference
		Cost	Sampling, Miscellaneous
	Median [8]	Quality	Truth Inference
		Cost	Sampling
	Group By [4]	Quality	Truth Inference, Task Assignment
		Cost	Task Selection, Answer Deduction
Skyline	[11]	Quality	Truth Inference, Task Assignment
		Cost	Task Selection
	[6]	Quality	Truth Inference, Task Assignment
		Cost	Task Selection
Planning	CrowdPlanr [10]	Quality	Truth Inference, Task Assignment
		Cost	Task Selection, Answer Deduction
	Route Plan [25]	Quality	Truth Inference, Task Assignment
		Cost	Task Selection, Answer Deduction
	CrowdPlanner [17]	Quality	Truth Inference, Task Assignment
		Cost	Task Selection
Schema Matching	[24]	Quality	Truth Inference, Task Assignment
		Cost	Task Selection, Answer Deduction
Mining	CrowdMiner [2]	Quality	Truth Inference, Task Assignment
		Cost	Task Selection
	OASSIS [1]	Quality	Truth Inference, Task Assignment
		Cost	Task Selection, Answer Deduction
Spatial	[18]	Quality	Truth Inference, Task Assignment
		Cost	Task Selection

[21] J. Wang, G. Li, T. Kraska, M. J. Franklin, and J. Feng. Leveraging transitive relations for crowdsourced joins. In *SIGMOD*, 2013.
[22] T. Yan, V. Kumar, and D. Ganesan. Crowdsearch: exploiting crowds for accurate real-time image search on mobile phones. In *MobiSys*.
[23] P. Ye, U. EDU, and D. Doermann. Combining preference and absolute judgements in a crowd-sourced setting. In *ICML Workshop*, 2013.
[24] C. J. Zhang, L. Chen, H. V. Jagadish, and C. C. Cao. Reducing uncertainty of schema matching via crowdsourcing. *PVLDB*, 6(9):757–768, 2013.
[25] C. J. Zhang, Y. Tong, and L. Chen. Where to: Crowd-aided path selection. *PVLDB*, 7(14):2005–2016, 2014.
[26] X. Zhang, G. Li, and J. Feng. Crowdsourced top-k algorithms: An experimental evaluation. *PVLDB*, 9(4):372–383, 2015.
[27] Y. Zheng, R. Cheng, S. Maniu, and L. Mo. On optimality of jury selection in crowdsourcing. In *EDBT*, pages 193–204, 2015.
[28] Y. Zheng, G. Li, and R. Cheng. Docs: Domain-aware crowdsourcing system. *PVLDB*, 4(10), 2016.
[29] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng. Truth inference in crowdsourcing: Is the problem solved? *PVLDB*, 5(10), 2017.
[30] Y. Zheng, J. Wang, G. Li, R. Cheng, and J. Feng. QASCA: A quality-aware task assignment system for crowdsourcing applications. In *SIGMOD*, pages 1031–1046, 2015.