Crowd-Powered Data Mining

Chengliang Chai*, Ju Fan†, Guoliang Li*, Jiannan Wang‡, Yudian Zheng#

*Tsinghua University, †Renmin University, ‡SFU, #Twitter

ABSTRACT

Many data mining tasks cannot be completely addressed by automated processes, such as sentiment analysis and image classification. Crowdsourcing is an effective way to harness the human cognitive ability to process these machine-hard tasks. Thanks to public crowdsourcing platforms, e.g., Amazon Mechanical Turk and CrowdFlower, we can easily involve hundreds of thousands of ordinary workers (i.e., the crowd) to address these machine-hard tasks. In this tutorial, we will survey and synthesize a wide spectrum of existing studies on crowd-powered data mining. We first give an overview of crowdsourcing, and then summarize the fundamental techniques, including quality control, cost control, and latency control, which must be considered in crowdsourced data mining. Next we review crowd-powered data mining operations, including classification, clustering, pattern mining, machine learning using the crowd (including deep learning, transfer learning and semi-supervised learning) and knowledge discovery. Finally, we provide the emerging challenges in crowdsourced data mining.

1 INTRODUCTION

Many data mining tasks cannot be effectively solved by existing machine-only algorithms, such as image classification [47], sentiment analysis [30], and opinion mining [7]. For example, given a set of pictures of famous places of interest in the world, we want to cluster them according to the country they belong to. Human can use their knowledge to categorize the pictures into countries like "China" or "America", but it is rather hard for machines. Fortunately, crowdsourcing has been emerged as an effective way to address such machine-hard tasks by utilizing hundreds of thousands of ordinary workers (i.e., the crowd). Thanks to the public crowdsourcing platforms, e.g., Amazon Mechanical Turk (AMT) and CrowdFlower, the access to the crowd becomes easier.

Crowdsourcing Overview. Over the past few years, crowdsourcing has become an active area from both research and industry (see a survey [40] and a book [39]). Typically, in a crowdsourcing platform (e.g., AMT [1]), there are two types of users, called *"workers"* and *"requesters"*. Requesters publish tasks on a crowdsourcing platform, while workers perform tasks and return the results. Suppose a requester has a classification problem to solve, which aims to classify an image to a category hierarchy. The requester needs to first perform *"task design"*, e.g., designing the user interface of a task (e.g., providing workers with an image and a category, and asking the workers to check whether the image belongs to the category), and set up some properties of the tasks (e.g., the price of a task, the number of workers to answer a task, the time duration to answer a task). Then the requester publishes the tasks to the platform. Workers who are willing to perform such tasks accept the tasks, answer them and submit the answers back to the platform. The platform collects the answers and reports them to the requester. If a worker has accomplished a task, the requester who publishes the task can approve or disapprove the worker's answers, and the approved workers will get paid from the requester.

Challenges in Crowdsourcing. The crowd has some different characteristics from machines. (1) *Not Free*. Workers need to be paid to answer a task, and it is important to control the *cost*. (2) *Error Prone*. Workers may return noisy results, and we need to tolerate the noises and improve the *quality*. Moreover, workers have various background knowledge, leading to different accuracies to answer different tasks. We need to capture workers' characteristics to achieve high *quality*. (3) *Dynamic*. Workers are not always online to answer tasks and we need to control the *latency*. Thus three core techniques must be considered in crowdsourcing: *"quality control"*, *"cost control"*, and *"latency control"*. Quality control aims to generate high-quality answers from workers' (possibly noisy) answers, by characterizing a worker's quality and aggregating workers' answers [2, 3, 8, 9, 18, 27, 29, 31–34, 43, 49, 64, 66, 70, 73–75, 78]. Cost control focuses on how to reduce human costs while still keeping good result quality [10, 11, 15–17, 21, 22, 24, 27, 35, 37, 38, 42, 46–48, 56, 58–60, 65, 68, 72, 76, 77]. Latency control exploits how to reduce the latency by modeling workers' latency and estimating workers' arrival rates [23, 25, 28]. Note there are trade-offs among quality, cost, and latency, and existing studies focus on how to balance them, e.g., optimizing the quality given a fixed cost, reducing the latency given a fixed cost, minimizing the cost under latency and quality constraints, etc.

Crowd Mining. There are many studies that utilize the crowdsourcing to address the data mining tasks, including classification, clustering, patterning mining, outlier detection, and knowledge base construction and enrichment.

(1) *Crowd-Powered Pattern Mining*. Crowd pattern mining tries to learn and observe significant patterns based on workers' answers. The problem of discovering significant patterns in crowd's behavior is an important but challenging task. For example, a health researcher is interested in analyzing the performance of traditional medicine and she tries to discover the association rules such that "*Garlic can be used to treat flu*". In this case, she can neither count on a database which only contains symptoms and treatments for a

KDD'18, August 19 - 23 , 2018, London, UK

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

particular disease, nor ask the healers for an exhaustive list of all the cases that have been treated. But social studies have shown that *although people cannot recall all their transactions, they can provide simple summaries (or called "personal rules") to the question*. For example, they may know that "*When I have flu, in most of the cases I will take Garlic because it indeed useful to me*". Given personal rules answered by different persons, they can be aggregated together to find an overall important rule (or the general trends). So the crowd pattern ming aims to collect the personal rules from crowd workers, aggregate them and find the overall important rules (i.e., general trends).

Crowd pattern mining typically generates a huge set of frequent patterns without providing enough information to interpret the meaning of the patterns. It would be helpful if we could also generate semantic annotations for the frequent patterns found, which would help us better understand the patterns. Existing works [4, 5, 55, 62, 67, 71] leverage the crowd ability to do semantic annotation, which mainly focus on improving the annotation accuracy and reducing the annotation cost.

(2) *Crowd-Powered Classification*. Some classification tasks are rather difficult for machines but easy for the crowd, e.g., image classification, and crowd-powered classification aims to leverage the crowd's intelligence to classify the data. Since the crowd may make mistake, existing works [6, 14, 44, 50–52, 63, 79] mainly focus on finding the correct classification from noisy crowd answers.

(3) *Crowd-Powered Clustering*. Many clustering tasks are easier for humans than machines. For example, given a set of sports events, human can easily categorize them into clusters like basketball, football according to their knowledge or experience. Some works focus on improving the clustering accuracy [26, 54, 61, 69]. Some works [13, 45, 57] not only care about the quality, but also optimize the cost or get high-quality results within a given budget.

(4) *Crowd-Powered Machine Learning*. Crowdsourcing can play an important role in machine learning, such as labeling data or debugging the model. There are several challenges of using the crowd in machine learning field. Firstly, when the number of data to be labeled but human is very large, it is expensive for hiring experts and the crowd. Therefore, we can utilize transfer learning or semi-supervised learning to do the task. Secondly, since the crowd workers are likely to make mistakes, we have to handle the errors. For example, deep learning can automatically tolerate the errors through the neural network. In this tutorial, we will discuss the usages of crowdsourcing in these advanced machine leaning algorithms.

(5) *Knowledge Discovery*. We have witnessed the booming of large-scale and open-accessible knowledge bases (KBs), which contain thousands of millions of real-world entities, categories and relationships. However, despite the impressive size, no KB is complete. For example, KBs miss many entities, especially the long-tail entities. Thus, some existing works utilize crowdsourcing for knowledge base construction and enrichment, and existing studies can be classified into the following categories. (a) Crowd-powered knowledge acquisition: Kumar et al. [36] combine the crowdsourcing with information extraction techniques for knowledge acquisition in order to fill in missing relations among entities in KBs. (b) Crowd-powered entity collection: some works [12, 20, 53] utilize crowdsourcing to collect entities that are missing in a KB, e.g., collecting all active NBA players. (c) Crowd-powered knowledge integration: some works focus on integrating multiple KBs or linking entities in KB to external sources (e.g., web tables). For example, Zhuang et al. [80] leverage the crowd to align entities from multiple knowledge bases, which focuses on reducing the cost and achieving higher quality. Fan et al. [19] solicit the crowd to link categories in a KB to columns in web tables by using a hybrid human-machine approach. On the other hand, there are some works using knowledge base for better modeling the crowd. For instance, Zheng et al. [76] and Ma et al. [44] use the KB to model workers' quality considering the domain knowledge.

2 TUTORIAL AUDIENCE AND PREREQUISITE FOR THE TUTORIAL

This is a 3 hours' tutorial. The intended audience include all KDD attendees from both research and industry communities. We will not require any prior background knowledge in crowdsourcing.

3 TUTORIAL OUTLINE

We first give an overview of crowdsourcing, including motivation of crowdsourcing, basic concepts (e.g., workers, requesters, tasks), crowdsourcing platforms, crowdsourcing workflow, and crowdsourcing applications. Then we talk about fundamental techniques to address the three challenges: quality control, cost control, and latency control. Next, we discuss crowd mining operations. Finally we provide emerging challenges.

Tutorial Structure:

- Crowdsourcing Overview (20 minutes)
  - Crowdsourcing motivation
  - Crowdsourcing workflow
  - Crowdsourcing applications
  - Crowdsourcing platforms
- Quality control (40 minutes)
  Crowd workers may return 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). For example, we can simply model a worker's quality as a probability, e.g., 0.8, i.e., the worker has a probability of 0.8 to correctly answer a task. To compute the probability, we can label some golden tasks with ground truth, and then the probability can be computed based on golden tasks. More sophisticated models will be introduced in our tutorial. Then based on the quality model of workers, there are several strategies to improve the quality in crowdsourcing. First, we can eliminate the low-quality workers (called worker elimination). For example, we can block the workers whose quality is below 0.6. Second, we can assign a task to multiple workers and infer the true answers by aggregating workers' results (called truth inference). For example, we can assign each task to five workers and then use majority voting to aggregate the answer. Third, we can

assign tasks to appropriate workers that are good at such tasks (called task assignment). Thus, in order to build a robust, reliable and online crowdsourcing system, we need to (1) design smart truth inference algorithms that can tolerate crowd errors and infer high-quality results from noisy answers; (2) design online task assignment algorithms that can wisely use the budgets by dynamically assigning tasks to appropriate workers; (3) integrate truth inference and online task assignment in a self-learned system, by iteratively updating parameters (e.g., worker quality, task answers) based on workers' feedbacks and dynamically making reasonable online task assignment.

- Cost control (30 minutes)
  The crowd is not free, and if there are large numbers of tasks, crowdsourcing could be expensive. Thus cost control is indispensable in the system to prevent overspending. There are several effective cost-control techniques. The first is pruning, which first uses machine algorithms to remove some unnecessary tasks and then utilizes the crowd to answer the necessary tasks. For example, for image classification, we can prune the categories of an image where the image has a rather small possibility to belong to these categories. The second is task selection, which prioritizes the tasks and decides which tasks to crowdsource first. For example, suppose we assign a classification task to five workers iteratively. If three workers return that the image belongs to the given category, we do not need to ask the fourth and fifth workers. The third is answer deduction, which crowdsources a subset of tasks and deduces the results of other tasks based on the answers collected from the crowd. For example, for a dog image, if the crowd returns that it belongs to the category "Dog" and then we do not need to ask whether it belongs to "Animal". The fourth is sampling, which samples a subset of tasks to crowdsource. For example, suppose we want to know the percentage of "Dog" images in a dataset, we can use sampling to compute an approximate answer. The fifth is task design, which designs a better user interface for the tasks. For example, we do not want to enumerate every category in a hierarchy, and instead we can use a tree interface to ask the workers to easily identify the corresponding category.
- Break (30 minutes)
- Latency control (20 minutes)
  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. Note that the latency does not simply depend on the number of tasks and the average time spent on each task, because crowd workers perform tasks in parallel. Existing latency-control techniques can be classified into three categories. (1) *Single-task latency control* aims to reduce the latency of one task (e.g., the latency of labeling each individual image). (2) *Single-batch latency control* aims to reduce the latency of a batch of tasks (e.g., the latency of labeling 10 images at the same time). (3) *Multi-batch latency control* aims to reduce the latency of multiple batches of tasks (e.g., adopting an iterative workflow to label a group of images where each iteration labels a batch of 2 images).
- Crowd Mining (60 minutes)
  There are many data mining tasks can be achieved with higher quality by the crowd. In this tutorial, we will talk about pattern mining(10 min), classification (10 min), clustering(10 min), machine learning (10 min) and knowledge discovery(20 min). We will illustrate how to design crowdsourced tasks according to different data mining tasks, how to achieve high quality results and how to reduce cost and latency.
- Conclusion and future directions: (10 minutes)

4 TUTORS

Chengliang Chai
Affiliation: Tsinghua University
E-mail: chaicl15@mails.tsinghua.edu.cn
Address: Department of Computer Science, Tsinghua University, Beijing, China
Phone: 86-13001266011

Ju Fan
Affiliation: Renmin University of China
E-mail: fanj@ruc.edu.cn
Address: Room 500, Information Building, Renmin University of China, No.59 Zhongguancun Street, Beijing 100872, China
Phone: 86-10-62512304

Guoliang Li (corresponding author)
Affiliation: Tsinghua University
E-mail: liguoliang@tsinghua.edu.cn
Address: Department of Computer Science, Tsinghua University, Beijing, China
Phone: 86-10-62789150

Jiannan Wang
Affiliation: Simon Fraser University
E-mail: jnwang@sfu.ca
Address: School of Computing Science, Simon Fraser University, Burnaby, Canada
Phone: 1-7787824288

Yudian Zheng
Affiliation: Twitter Inc.
E-mail: yudianz@twitter.com
Address: 1355 Market St., Suite 900, San Francisco, CA 94103, US.
Phone: 1-4153702403

5 TUTOR'S BIO AND EXPERTISE

Chengliang Chai is a third year PhD candidate student at the Department of Computer Science, Tsinghua University, Beijing, China. He obtained his B.E. in 2015 from Harbin Institute of Technology, China. He visited University of Wisconsin-Madison in 2017. His research interests mainly include data cleaning and integration and crowdsourcing (especially cost control and latency control).

Ju Fan is currently working as an associate professor at the Department of Computer Science in Renmin University, Beijing, China.

His main research interests include knowledge base and crowdsourcing (especially cost control and knowledge base construction and enrichment). He has published more than 30 papers in leading international conferences/journals, including SIGMOD, VLDB, ICDE, ICDM, etc. He got ACM China 2017 rising star award.

Guoliang Li is currently working as an associate professor at the Department of Computer Science, Tsinghua University, Beijing, China. His research interests mainly include data cleaning and integration, and crowdsourcing (especially cost control, quality control, latency control). He has published more than 100 papers in SIGMOD, KDD, VLDB, ICDE, SIGIR, VLDB Journal, TODS, TKDE, and his papers have been cited by more than 5000 times. He got VLDB 2017 Early Research Contribution Award, IEEE TCDE Early Career Award 2014, CIKM 2017 Best Paper Award, DASFAA 2014 Best Paper Runner-up, and APWeb 2014 Best Paper Award.

Jiannan Wang is currently working as an assistant professor at the School of Computer Science in Simon Fraser University, Canada. His research is focused on data cleaning, interactive analytics, and crowdsourcing (especially cost control and latency control). Prior to that, he was a postdoc in the AMPLab at UC Berkeley. He obtained his PhD from the Computer Science Department at Tsinghua University. His recent awards include a best demo award at SIGMOD 2016, a Distinguished Dissertation Award from the China Computer Federation (2013), and a Google Ph.D. Fellowship (2011).

Yudian Zheng is currently at Twitter, focusing on building large scale online distributed machine learning systems. His main research interests include data analysis, crowdsourced data management and machine learning. He has published more than 20 papers in leading international conferences/journals, including SIGMOD, KDD, VLDB, TKDE, ICDE, etc.

6 RELATED PAST TUTORIALS

We have gave a tutorial at SIGMOD 2017 [41] which focuses on crowdsourced data management. Compared with that tutorial, we focus on the fundamental techniques for crowd data mining.

7 EQUIPMENT

The tutorial does not require any special equipment. We will use our own laptop for presentation.

8 SLIDES DUE AND PREVIOUS WEBSITES

We understand the due of slides is on July 29, 2018. Actually if the proposal is accepted, we will get the slides and website ready before July 2018. Please kindly check our previous tutorials: http://dbgroup.cs.tsinghua.edu.cn/ligl/papers/sigmod17-tutorial-crowd.pdf. We have also written a survey [40] and a book [39] in crowdsourcing.

9 VIDEO SNIPPET

Here are two video links of our tutorial talks at SIGMOD 2017 from YouTube.

https://www.youtube.com/watch?v=ADAp7XMGtjw
https://www.youtube.com/watch?v=-45JkIVYhvo

REFERENCES

[1] https://www.mturk.com/.
[2] A.P.Dawid and A.M.Skene. Maximum likelihood estimation of observer error-rates using em algorithm. *Appl.Statist.*, 28(1):20–28, 1979.
[3] B. I. Aydin, Y. S. Yilmaz, Y. Li, Q. Li, J. Gao, and M. Demirbas. Crowdsourcing for multiple-choice question answering. In *AAAI*, pages 2946–2953, 2014.
[4] Y. Baba and H. Kashima. Statistical quality estimation for general crowdsourcing tasks. In *KDD*, pages 554–562, 2013.
[5] A. Basharat, I. B. Arpinar, and K. Rasheed. Leveraging crowdsourcing for the thematic annotation of the qur'an. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11-15, 2016, Companion Volume*, pages 13–14, 2016.
[6] T. Bonald and R. Combes. A minimax optimal algorithm for crowdsourcing. In *NIPS*, pages 4355–4363, 2017.
[7] C. C. Cao, L. Chen, and H. V. Jagadish. From labor to trader: opinion elicitation via online crowds as a market. In *KDD*, pages 1067–1076, 2014.
[8] C. C. Cao, J. She, Y. Tong, and L. Chen. Whom to ask? jury selection for decision making tasks on micro-blog services. *PVLDB*, 5(11):1495–1506, 2012.
[9] C. Chai, J. Fan, and G. Li. Incentive-based entity collection using crowdsourcing. In *ICDE*, pages 1529–1540, 2018.
[10] C. Chai, G. Li, J. Li, D. Deng, and J. Feng. Cost-effective crowdsourced entity resolution: A partial-order approach. In *SIGMOD*, pages 969–984, 2016.
[11] X. Chen, P. N. Bennett, K. Collins-Thompson, and E. Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *WSDM*, pages 193–202, 2013.
[12] Y. Chung, M. L. Mortensen, C. Binnig, and T. Kraska. Estimating the impact of unknown unknowns on aggregate query results. In *SIGMOD*, pages 861–876, 2016.
[13] S. B. Davidson, S. Khanna, T. Milo, and S. Roy. Top-k and clustering with noisy comparisons. *ACM Trans. Database Syst.*, 39(4):35:1–35:39, 2014.
[14] O. Dekel and O. Shamir. Vox populi: Collecting high-quality labels from a crowd. In *COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009*, 2009.
[15] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *WWW*, pages 469–478, 2012.
[16] B. Eriksson. Learning to top-k search using pairwise comparisons. In *AISTATS*, pages 265–273, 2013.
[17] J. Fan and G. Li. Human-in-the-loop rule learning for data integration. *IEEE Data Eng. Bull.*, 41(2):104–115, 2018.
[18] J. Fan, G. Li, B. C. Ooi, K. Tan, and J. Feng. icrowd: An adaptive crowdsourcing framework. In *SIGMOD*, pages 1015–1030, 2015.
[19] J. Fan, M. Lu, B. C. Ooi, W. Tan, and M. Zhang. A hybrid machine-crowdsourcing system for matching web tables. In *IEEE 30th International Conference on Data Engineering, Chicago, ICDE 2014, IL, USA, March 31 - April 4, 2014*, pages 976–987, 2014.
[20] J. Fan, Z. Wei, D. Zhang, J. Yang, and X. Du. Distribution-aware crowdsourced entity collection. *IEEE Trans. Knowl. Data Eng.*, to appear, 2017.
[21] Y. Fang, H. Sun, G. Li, R. Zhang, and J. Huai. Effective result inference for context-sensitive tasks in crowdsourcing. In *DASFAA*, pages 33–48, 2016.
[22] Y. Fang, H. Sun, G. Li, R. Zhang, and J. Huai. Context-aware result inference in crowdsourcing. *Inf. Sci.*, 460-461:346–363, 2018.
[23] S. Faradani, B. Hartmann, and P. G. Ipeirotis. What's the right price? pricing tasks for finishing on time. In *AAAI Workshop*, 2011.
[24] J. Feng, G. Li, H. Wang, and J. Feng. Incremental quality inference in crowdsourcing. In *DASFAA*, pages 453–467, 2014.
[25] Y. Gao and A. G. Parameswaran. Finish them!: Pricing algorithms for human computation. *PVLDB*, 7(14):1965–1976, 2014.
[26] R. Gomes, P. Welinder, A. Krause, and P. Perona. Crowdclustering. In *NIPS*, pages 558–566, 2011.
[27] S. Guo, A. G. Parameswaran, and H. Garcia-Molina. So who won?: dynamic max discovery with the crowd. In *SIGMOD*, pages 385–396, 2012.
[28] D. Haas, J. Wang, E. Wu, and M. J. Franklin. Clamshell: Speeding up crowds for low-latency data labeling. *PVLDB*, 9(4):372–383, 2015.
[29] C.-J. Ho, S. Jabbari, and J. W. Vaughan. Adaptive task assignment for crowdsourced classification. In *ICML*, pages 534–542, 2013.
[30] Y. Hu, F. Wang, and S. Kambhampati. Listening to the crowd: Automated analysis of events via aggregated twitter sentiment. In *IJCAI*, pages 2640–2646, 2013.
[31] P. Ipeirotis, F. Provost, and J. Wang. Quality management on amazon mechanical turk. In *KDD Workshop*, pages 64–67, 2010.
[32] M. Joglekar, H. Garcia-Molina, and A. G. Parameswaran. Evaluating the crowd with confidence. In *KDD*, pages 686–694, 2013.
[33] M. Joglekar, H. Garcia-Molina, and A. G. Parameswaran. Comprehensive and reliable crowd assessment algorithms. In *ICDE*, pages 195–206, 2015.
[34] D. R. Karger, S. Oh, and D. Shah. Iterative learning for reliable crowdsourcing systems. In *NIPS*, pages 1953–1961, 2011.
[35] A. R. Khan and H. Garcia-Molina. Hybrid strategies for finding the max with the crowd. Technical report, 2014.
[36] S. K. Kondreddi, P. Triantafillou, and G. Weikum. Combining information extraction and human computing for crowdsourced knowledge acquisition. In *IEEE 30th International Conference on Data Engineering, Chicago, ICDE 2014, IL, USA, March 31 - April 4, 2014*, pages 988–999, 2014.

[37] G. Li. Human-in-the-loop data integration. *PVLDB*, 10(12):2006–2017, 2017.

[38] G. Li, C. Chai, J. Fan, X. Weng, J. Li, Y. Zheng, Y. Li, X. Yu, X. Zhang, and H. Yuan. CDB: optimizing queries with crowd-based selections and joins. In *SIGMOD*, pages 1463–1478, 2017.

[39] G. Li, J. Wang, Y. Zheng, J. Fan, and M. J. Franklin. Crowdsourced data management. *Springer*, 2018.

[40] G. Li, J. Wang, Y. Zheng, and M. J. Franklin. Crowdsourced data management: A survey. *TKDE*, 2015.

[41] G. Li, Y. Zheng, J. Fan, J. Wang, and R. Cheng. Crowdsourced data management: Overview and challenges. In *SIGMOD*, 2017.

[42] K. Li, X. Zhang, and G. Li. A rating-ranking method for crowdsourced top-k computation. In *SIGMOD*, pages 975–990, 2018.

[43] X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang. CDAS: A crowdsourcing data analytics system. *PVLDB*, 5(10):1040–1051, 2012.

[44] F. Ma, Y. Li, Q. Li, M. Qiu, J. Gao, S. Zhi, L. Su, B. Zhao, H. Ji, and J. Han. Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation. In *KDD*, pages 745–754, 2015.

[45] A. Mazumdar and B. Saha. Clustering via crowdsourcing. *CoRR*, abs/1604.01839, 2016.

[46] B. Mozafari, P. Sarkar, M. Franklin, M. Jordan, and S. Madden. Scaling up crowd-sourcing to very large datasets: a case for active learning. *PVLDB*, 8(2):125–136, 2014.

[47] 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.

[48] T. Pfeiffer, X. A. Gao, Y. Chen, A. Mao, and D. G. Rand. Adaptive polling for information aggregation. In *AAAI*, 2012.

[49] V. C. Raykar, S. Yu, L. H. Zhao, A. K. Jerebko, C. Florin, G. H. Valadez, L. Bogoni, and L. Moy. Supervised learning from multiple experts: whom to trust when everyone lies a bit. In *ICML*, pages 889–896, 2009.

[50] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(Apr):1297–1322, 2010.

[51] H. Salehian, P. D. Howell, and C. Lee. Matching restaurant menus to crowdsourced food data: A scalable machine learning approach. In *KDD*, pages 2001–2009, 2017.

[52] V. S. Sheng, F. J. Provost, and P. G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *KDD*, pages 614–622, 2008.

[53] B. Trushkowsky, T. Kraska, M. J. Franklin, and P. Sarkar. Crowdsourced enumeration queries. In *ICDE 2013*, pages 673–684, 2013.

[54] A. Ukkonen. Crowdsourced correlation clustering with relative distance comparisons. In *ICDM*, pages 1117–1122, 2017.

[55] A. Vempala and E. Blanco. Complementing semantic roles with temporally anchored spatial knowledge: Crowdsourced annotations and experiments. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 2652–2658, 2016.

[56] N. Vesdapunt, K. Bellare, and N. N. Dalvi. Crowdsourcing algorithms for entity resolution. *PVLDB*, 7(12):1071–1082, 2014.

[57] R. K. Vinayak and B. Hassibi. Crowdsourced clustering: Querying edges vs triangles. In *NIPS*, pages 1316–1324, 2016.

[58] J. Wang, T. Kraska, M. J. Franklin, and J. Feng. CrowdER: crowdsourcing entity resolution. *PVLDB*, 5(11):1483–1494, 2012.

[59] J. Wang, G. Li, T. Kraska, M. J. Franklin, and J. Feng. Leveraging transitive relations for crowdsourced joins. In *SIGMOD*, 2013.

[60] S. Wang, X. Xiao, and C. Lee. Crowd-based deduplication: An adaptive approach. In *SIGMOD*, pages 1263–1277, 2015.

[61] F. L. Wauthier, N. Jojic, and M. I. Jordan. Active spectral clustering via iterative uncertainty reduction. In *KDD*, pages 1339–1347, 2012.

[62] X. Wei, D. D. Zeng, and J. Yin. Multi-label annotation aggregation in crowdsourcing. *CoRR*, abs/1706.06120, 2017.

[63] P. Welinder, S. Branson, S. J. Belongie, and P. Perona. The multidimensional wisdom of crowds. In *NIPS*, pages 2424–2432, 2010.

[64] X. Weng, G. Li, H. Hu, and J. Feng. Crowdsourced selection on multi-attribute data. In *CIKM*, pages 307–316, 2017.

[65] S. E. Whang, P. Lofgren, and H. Garcia-Molina. Question selection for crowd entity resolution. *PVLDB*, 6(6):349–360, 2013.

[66] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. R. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*, pages 2035–2043, 2009.

[67] B. Wu, E. Zhong, B. Tan, A. Horner, and Q. Yang. Crowdsourced time-sync video tagging using temporal and personalized topic modeling. In *KDD*, pages 721–730, 2014.

[68] P. Ye, U. EDU, and D. Doermann. Combining preference and absolute judgements in a crowd-sourced setting. In *ICML Workshop*, 2013.

[69] J. Yi, R. Jin, A. K. Jain, S. Jain, and T. Yang. Semi-crowdsourced clustering: Generalizing crowd labeling by robust distance metric learning. In *NIPS*, pages 1781–1789, 2012.

[70] D. Yuan, G. Li, Q. Li, and Y. Zheng. Sybil defense in crowdsourcing platforms. In *CIKM*, pages 1529–1538, 2017.

[71] L. Zhang, L. Tang, P. Luo, E. Chen, L. Jiao, M. Wang, and G. Liu. Harnessing the wisdom of the crowds for accurate web page clipping. In *KDD*, pages 570–578, 2012.

[72] X. Zhang, G. Li, and J. Feng. Crowdsourced top-k algorithms: An experimental evaluation. *PVLDB*, 9(4):372–383, 2015.

[73] Z. Zhao, F. Wei, M. Zhou, W. Chen, and W. Ng. Crowd-selection query processing in crowdsourcing databases: A task-driven approach. In *EDBT*, pages 397–408, 2015.

[74] Z. Zhao, D. Yan, W. Ng, and S. Gao. A transfer learning based framework of crowd-selection on twitter. In *KDD*, pages 1514–1517, 2013.

[75] Y. Zheng, R. Cheng, S. Maniu, and L. Mo. On optimality of jury selection in crowdsourcing. In *EDBT*, pages 193–204, 2015.

[76] Y. Zheng, G. Li, and R. Cheng. DOCS: domain-aware crowdsourcing system. *PVLDB*, 10(4):361–372, 2016.

[77] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng. Truth inference in crowdsourcing: Is the problem solved? *PVLDB*, 2017.

[78] 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.

[79] H. Zhuang, A. G. Parameswaran, D. Roth, and J. Han. Debiasing crowdsourced batches. In *KDD*, pages 1593–1602, 2015.

[80] Y. Zhuang, G. Li, Z. Zhong, and J. Feng. Hike: A hybrid human-machine method for entity alignment in large-scale knowledge bases. In *CIKM*, pages 1917–1926, 2017.