Crowdsourcing-based Data Extraction from Visualization Charts

Chengliang Chai† Guoliang Li† Ju Fan‡ Yuyu Luo†
†Tsinghua University, China ‡Renmin University of China
{ chaicl15@mails., liguoliang@, luoyy18@mails.}tsinghua.edu.cn, fanj@ruc.edu.cn

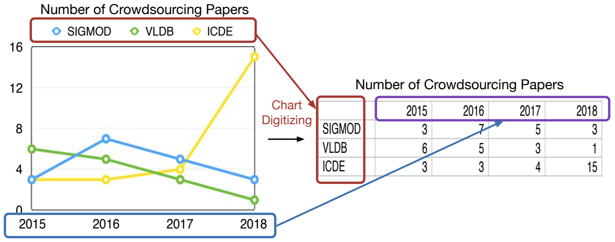
***Abstract*—Visualization charts are widely utilized for presenting structured data. Under many circumstances, people want to explore the data in the charts collected from various sources, such as papers and websites, so as to further analyzing the data or creating new charts. However, the existing automatic and semi-automatic approaches are not always effective due to the variety of charts. In this paper, we introduce a crowdsourcing approach that leverages human ability to extract data from visualization charts. There are several challenges. The first one is how to avoid tedious human interaction with charts and design simple crowdsourcing tasks. Second, it is challenging to evaluate worker's quality for truth inference, because workers may not only provide inaccurate values but also misalign values to wrong data series. To address the challenges, we design an effective crowdsourcing task scheme that splits a chart into simple micro-tasks. We introduce a novel worker quality model by considering worker's accuracy and task difficulty. We also devise an effective early-stopping mechanisms to save the cost. We have conducted experiments on a real crowdsourcing platform, and the results show that our framework outperforms state-of-the-art approaches on both cost and quality.**

I. Introduction

Charts are indispensable tools to visualize structured data due to their perceptual advantages [13]. They do not only help people understand many aspects of data, such as distribution and variation trend, but also provide intuitive comparisons for data from different sources. An example line chart, shown in Fig. 1, is used to visualize numbers of crowdsourcing papers at three leading DB conferences from 2015 to 2018. Very often, people, like data analysts, want to extract the underlying data from charts, so as to further analyze the data, update the charts, or create new charts by integrating data from various sources.

Indeed, the topic of data extraction from charts has attracted much interest in research community in recent years. Some automatic or semi-automatic chart data extraction tools have been developed [7], [9]. Automatic tools like [7] apply computer vision and machine learning models to first recognize the text in a chart and then infer the underlying data points. However, the performance of such methods is far from satisfactory: accuracy of both the text recognition and data point extraction is normally around 60% - 70% [9]. However, to support effective data analysis, users usually request for a much higher data extraction accuracy.

Crowdsourcing is an effective approach to leverage the human intelligence to do machine-hard problems [8], [2], [3], [4], [10], [6], [15]. To address the above limitations, we propose a crowdsourcing chart data extraction framework CrowdChart that harnesses the huge number of crowd workers on crowdsourcing platforms like Amazon Mechanical Turk (AMT) [1] to extract data from charts at relatively low cost. We study the following research challenges that naturally arise in the framework.



	2015	2016	2017	2018
SIGMOD	3	7	5	3
VLDB	6	5	3	1
ICDE	3	3	4	15

Fig. 1: Example for Chart Extraction

The first challenge is how to design crowdsourcing tasks. A straightforward method is to crowdsource an entire chart and ask the worker to submit a relational table. Obviously, such task is too much overwhelming to workers who are usually good at "micro"-tasks (see survey [11]). To address the problem, we design an effective crowdsourcing task scheme that splits a chart into a batch of micro-tasks, each of which extracts a specific part of the chart. Then, we can recover the relational table by aggregating crowd answers of the tasks.

The second challenge is quality control for crowdsourced chart data extraction. Although there exists some works [14], [17] on crowdsourcing numerical data, our scenario is more complicated. Quality of a worker is hard to evaluate, as it may not only depend on how careful the worker is, but also be affected by visual features of the chart, such as chart type, log-scaled y-axis, etc. Even worse, misalignment is a kind of common errors, even for careful workers, that can significantly influence the quality. For example, when extracting data, answers may be misaligned with their legend keys. For example, in the line chat of Fig. 1, a worker extracts the three data points [5,3,4] in 2017 accurately, but she may align 4 to `VLDB` and 3 to `ICDE`, leading to alignment errors. To address the challenge, we propose a truth inference model for numerical data. We introduce a Gaussian model to evaluate worker quality by considering worker's reliability and task difficulty. Then, we develop effective techniques for accurate worker estimation and truth inference.

The third challenge is how to reduce the crowdsourcing cost. To this end, we continuously evaluate quality of tasks and introduce an early-stopping strategy that terminates the tasks which already have satisfactory inferred results.

To summarize, we make the following contributions. We

2375-026X/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDE48307.2020.00177

propose a novel framework that systematically that utilizes the crowd to extract data from charts. We design a truth inference model to imporve quality and early-stopping techniques to reduce cost. We evaluate our approach on real datasets on AMT. The results demonstrate its superiority performance.

II. Problem Formulation

Chart model. Given a chart C, the data visualized in C consists of the following two elements: (1) A sequence of legend keys $K = [k_1, k_2, \dots, k_m]$; (2) a set of tuples $T = \{t_1, t_2, \dots, t_n\}$, where each tuple $t_i = [t_{i1}, t_{i2}, ..., t_{im}]$ represents the data points in the i-th labels of the horizontal axis. Note that the order of data points in each tuple t_i must be the same with the order of keys in K. Fig. 1 shows an example of chart data with three keys $K = [\texttt{SIGMOD}, \texttt{VLDB}, \texttt{ICDE}]$ and four tuples t_1 to t_4. For example, tuple $t_1 = [3, 6, 3]$ contains the data points corresponding to SIGMOD, VLDB and ICDE in 2015 respectively. Note that the pie chart is a special case with only one tuple containing the ratios or number of various keys.

Crowdsourcing task design. We harness the crowd intelligence to extract data from charts. We introduce a fine-grain approach that splits a chart into a batch of *micro*-tasks to reduce latency and improve quality. Specifically, we design four types of crowdsourcing tasks that can be categorized into two groups, i.e., the preprocessing tasks and tuple extraction task, as illustrated in Fig. 2.

As quality of chart data extraction may depend on visual features of the chart, we define the following three types of *preprocessing* tasks before extracting the data.

(1) Chart Classification Task: Intuitively, different types of charts have different difficulty levels for data extraction, which motivates us to first ask the crowd for chart classification. Given a chart C, a chart classification task is a multiple-choice question. Currently, we support four choices, `bar chart`, `line chart`, `pie chart` and `stacked bar chart`, and ask the crowd to choose the one that C belongs to. An example chart classification task is shown in Fig. 2(a), where a crowd worker will select the choice `Line Chart`.

(2) Y-axis Classification Task: Another factor affecting the difficulty is whether y-axis is log-scale. Thus, we also leverage the crowd to identify this issue as one of the preprocessing steps. Given a chart C, y-axis classification task is a Yes/No question to the crowd. An example task is shown in Fig. 2(b) where a crowd worker will select `No` for the question.

(3) Legend Identification Task: Legend is also hard for machine to identify as it have different patterns and may be located arbitrarily in the chart. Given a chart C, this task is a fill-in-blanks question that ask the crowd to collect a sequence of legend keys, i.e., K. Fig. 2(c) illustrates an example of legend identification task with three keys SIGMOD, VLDB and ICDE to be collected.

(4) Tuple extraction task. The central task for chart data extraction is to identify the tuples. Given a chart C, a sequence of legend keys $K = [k_1\, k_2, \dots, k_m]$ and a label i in horizontal axis, tuple extraction task is a fill-in-blanks question that collect the i-th tuple $t_i = [t_{i1}, t_{i2}, \dots, t_{im}]$. Fig. 2(d) shows a tuple extraction task, which aims to collect values corresponding to SIGMOD,VLDB and ICDE respectively. Thus, the chart in Fig. 2(d) can be divided into $N = 4$ tuple extraction tasks. Note that the order of the sequence in collected tuples is consistent with that of pre-collected legend keys.

The tuple extraction task is quite challenging because the workers are more error-prone to provide noisy answers. Thus, we study a truth inference problem, defined as follows.

Definition 2.1 (Truth Inference): For each point t_{ij}, given workers' answers set A_{ij}, the truth inference problem is to compute a well-estimated value $\hat{t_{ij}}$ for true value t^*_{ij}.

III. The CrowdChart Framework

We introduce a framework, called CrowdChart, for tuple extraction tasks. Once a crowd worker submits answers of a tuple extraction task, CrowdChart first aligns those answers based on the workers quality answers that have been submitted by others (Section III-A). Then we infer the truth considering the workers quality and task difficulty using the EM algorithm (Sections III-B, III-C and III-D). Then, the output of the truth inference model is an estimated truth distribution, from which we can compute the confidence of the estimated truth using an *early stopping* module. If it already has a high confidence, we do not need to assign more tasks to save the cost and return the final inferred answer (Section III-E).

A. Modeling Workers' Answers and Quality

Different from multi-choice tasks, the answers of data extraction tasks are numerical values. For a numerical task, its quality depends on how close it is to the ground truth. Formally, we use $a^w_i = [a^w_{i1}, a^w_{i2}, ..., a^w_{im}]$ to denote a sequence of answers for data points in task t_i by worker w. And we use the Gaussian distribution to model each answer given by worker w. The distribution takes the ground truth t^*_{ij} as its mean and uses variance to model worker quality, i.e.,

$$\begin{aligned} a^w_{ij} &\sim \mathcal{N}(t^*_{ij}, \phi^w_{ij}) \\ &\sim \frac{1}{\sqrt{2\pi\phi^w_{ij}}} \exp(-\frac{(a^w_{ij} - t^*_{ij})^2}{2\phi^w_{ij}}), \phi^w_{ij} = (\sigma^w_{ij})^2 \end{aligned} \tag{1}$$

where ϕ^w_{ij} is the variance and σ^w_{ij} is the standard deviation. Generally speaking, if w has a good quality, then variance ϕ^w_{ij} will be small because the answer is likely to be close to the ground truth t^*_{ij}. Motivated by this, we use q_w to denote the quality of w. Thus we have $\sigma^w_{ij} = -t^*_{ij} \ln q_w, q_w \in [0, 1]$. We use $-\ln q_w \in [0, +\infty]$ to denote the ratio. When q_w is close to 1, which indicates a high quality worker, the standard deviation σ^w_{ij} is small because $-\ln q_w$ is close to 0. For example, Given $t^*_{i', j'} = 100$, suppose that a worker with $q_w = 0.9$ ($\sigma^w_{ij} \approx 0.1 \times 100 = 10$) requests to answer it. Then we can infer that $p(80 < a^w_{ij} < 120) = 0.95$.

B. Difficulty of Data Points

Quality of workers' answers also depends on *difficulty* levels of the tasks. Not surprisingly, some complicated charts like line charts and stacked bar charts are challenging even for a human to digitize. Also, values along the log-scale Y-axis are always hard for some workers to recognize.

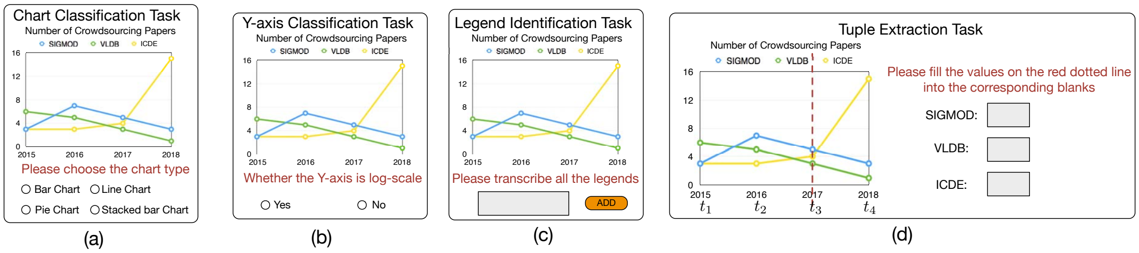


Chart Classification Task
Number of Crowdsourcing Papers
SIGMOD
VLDB
ICDE
16
12
8
4
0
2015
2016
2017
2018
Please choose the chart type
○ Bar Chart
○ Line Chart
○ Pie Chart
○ Stacked bar Chart
(a)
Y-axis Classification Task
Whether the Y-axis is log-scale
○ Yes
○ No
(b)
Legend Identification Task
Please transcribe all the legends
ADD
(c)
Tuple Extraction Task
Please fill the values on the red dotted line into the corresponding blanks
SIGMOD:
VLDB:
ICDE:
t_1
t_2
t_3
t_4
(d)

Fig. 2: Crowdsourcing task design for chart data extraction.

Formally, we model the difficulty of task t_i of a chart C, considering features $\mathbf{x}_i^1$, $\mathbf{x}_i^2$ and $\mathbf{x}_i^3$, which denotes the the chart classification, scale of Y-axis and legends number respectively. $\mathbf{x}_i^1$ is a one-hot vector with length 4, where we consider `bar chart`, `line chart`, `pie chart` and `stacked bar chart`. For example, $\mathbf{x}_i^1 = [0,1,0,0]^{\mathrm{T}}$ indicates it is a line chart. Concretely, $\mathbf{x}_i^2$ is either 1 or 0, which indicates whether the Y-aix is log-scale or not and $\mathbf{x}_i^3 = m$. Then, we use $d_i = \frac{1}{1+e^{-\sum_{k=1}^{3}\gamma_k \mathbf{x}_i^k}}$ to compute the difficulty of task t_i, where γ denotes the weights of different features.

Obviously, the more difficult the task is, the larger the difference between ground truth and workers' answer will be. Thus, we rewrite the answer quality as $\sigma_{ij}^w = -d_i^{\tau_w} t_{ij}^* \ln q_w, \tau_w \in [0,1]$, where the parameter τ_w aims to model how much the task difficulty can impact the worker w's answer.

C. Answers Alignment

Misalignment will inevitably happen when extracting data from charts because in many cases, the visual sequence of data points in the chart cannot match the sequence of these legends in the text region. This phenomenon cannot be neglected because it will influence both the workers quality and inferred truth. For example, if the misaligned answers are directly used to compute the ground truth, we will derive a truth with high bias, which results in that the worker who answered that task is estimated as a low quality worker. To this end, we propose a probability-based solution to align the answers.

Our goal is to infer the truth of data points in the task, i.e., $t_i^* = [t_{i1}^*, t_{i2}^*, ..., , t_{im}^*]$ based on the obtained answers. Given answers $a_i^w = [a_{i1}^w, a_{i2}^w, ..., a_{im}^w]$ for task t_i provided by w, we can generate a set of $m!$ possible sequences S. Each sequence $s_i \in S$ and s_{ij} denotes the j-th answer in sequence s_i. The alignment problem is to find the sequence that is most likely to match t_i^*. In other words, given the truth t_i^* and the worker's variance σ_w, we want to compute the probability of each possible sequence. However, since we do not know the ground truth, we use current estimated truth $\hat{t_i} = [\hat{t_{i1}}, \hat{t_{i2}}, ..., , \hat{t_{im}}]$ to compute the probability, $p(s_i, \hat{t_i}) = \prod_{j=1}^{m} \frac{1}{\sqrt{2\pi\phi_{ij}^w}} \exp(-\frac{(s_{ij}-\hat{t_{ij}})^2}{2\phi_{ij}^w})$. Since the number of legends in a chart is small (less than 5 in most time) in practice, it is not expensive to enumerate $m!$ sequences and select the one with the largest probability. Therefore, we select the sequence s^* with the largest probability as $s^* = \arg\max_{s_i \in S} p(s_i, \hat{t_i})$.

D. Inference Algorithm

We infer the truth and workers' quality based on current obtained answers using the maximum likelihood estimation. Given a set of parameters $\theta = \{\theta_w\}, \theta_w = \{\gamma, q_w, \tau_w\}$, the objective for the inference is to maximize the likelihood of worker answers,

$$\arg\max_{\theta} P(A|\theta) = \arg\max_{\theta} \sum_{\mathcal{T}^*} P(A, \mathcal{T}^*|\theta), \qquad (2)$$

where $\mathcal{T}^* = \{t^*\}$ is the truth of all the data points, which is taken as the hidden variable and A is answers of all data points. To solve this, we use the Expectation Maximization (EM) algorithm [5], which iteratively computes the truth distribution $\mathcal{T}^*$ and parameters θ.

E. Confidence-Aware Early Stopping

For some tasks, which have been answered by enough number of workers or a few high-quality workers, they already have derive high confidence answers, and thus do not need to be crowdsourced any more. This motivates us to design confidence-aware early stopping for saving the cost.

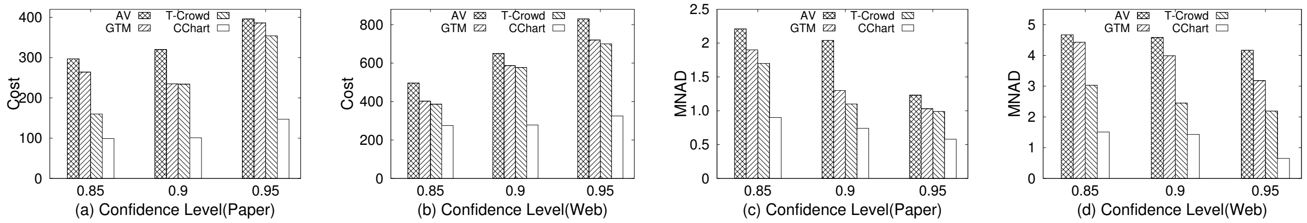
Given the truth distribution of a data point $t_{ij}^* \sim \mathcal{N}(\mu_{ij}, \sigma_{ij})$ obtained through the truth inference algorithm, we can compute the confidence if we regard μ_{ij} as the answer. We adopt the $(1-\alpha)$ confidence interval for the estimated truth, where $1-\alpha$, also known as the confidence level, is usually near to 1 such as 90%, 95%. We will trust the answer and stop to assign questions with respect to the task if it satisfies,

$$P((1-b)\mu_{ij} < t_{ij}^* < (1+b)\mu_{ij}) > 1-\alpha \qquad (3)$$

which gives the $(1-\alpha)$ confidence interval of t_{ij}^* as $r = [(1-b)\mu_{ij}, (1+b)\mu_{ij}]$, where b controls the width of the interval and is always small, like $b = 0.1$.

IV. Experiments

Experimental Settings. We use two real datasets to evaluate our approach, the details of which are summarized in Table I. (1) `Paper`: We extract 75 chartsfrom several research papers. The ground truth is the data used to draw those charts. (2) `Web`: We crawl 180 charts from the web. Specifically, for ease of collecting ground-truth, we crawl the chart from the websites with meta-data of charts. Moreover, we have implemented CrowdChart on top of CrowdOTA [16], which is an online task assignment framework built on AMT. For preprocessing tasks, we include the three kinds of task in a single human intelligence task(HIT) and pay \$0.1 for the HIT. For tuple extraction tasks, an HIT is used to extract one tuple t_i like Fig. 2(d), which costs \$0.05m where m is the number of values in t_i. In the evaluation, we mainly compare the cost and quality of CrowdChart with other baselines. (1) Cost.



(a) Confidence Level(Paper) (b) Confidence Level(Web) (c) Confidence Level(Paper) (d) Confidence Level(Web)

Fig. 3: Evaluation on Truth Inference: Cost & Quality

We utilize the monetary cost to evaluate the cost of different approaches. Note that, for different methods, the cost used for preprocessing tasks is the same, and thus we do not report this part. (2) Quality. For quality, we use the metric Mean Normalized Absolute Distance `MNAD` [12] to measure the overall absolute distance from each approachs results to the ground truths, which indicates how close the results are to the ground truths.

TABLE I: Datasets.

	$\mathcal{C}$	#Data points	#Line Chart	#Bar Chart	#Pie Chart
`Paper`	75	890	40	35	0
`Web`	180	2550	110	50	20

Evaluation on Truth Inference. We evaluate the truth inference in CrowdChart compared with the following state-of-the-art approaches with the focus on numeric data. (1) Average (`AV`): Average is a simple and intuitive method to tackle continuous answers. Given several answers of a data point by multiple workers, it computes the average as the truth. (2) `GTM` [17]: `GTM` is a truth discovery framework for numeric data, which considers the source reliability (workers' quality) and utilizes the EM algorithm to infer the truth. (3) `T-Crowd` [14]: `T-Crowd` is a crowdsourcing framework for tabular data, including both categorical and numeric data. In our scenario, we do not have categorical data, so we only compare with its technique designed for continuous data. We compare CrowdChart with `AV`, `GTM` and `T-Crowd` respectively. We set $\beta = 0.1$ and vary the confidence level from 0.85 to 0.95 to test the performance.

Figures 3 show the evaluation on crowdsourcing cost and quality. We can see from Fig 3(a) and (b) that CrowdChart saves more than two times of cost compared with other state-of-the-art works when achieving the same confidence level on the `Paper` dataset. For example, when the confidence level is 0.9, CrowdChart incurs a cost of \$101 while `AV`, `GTM` and `T-Crowd` use \$320, \$235 and \$234 respectively. This because CrowdChart will align the answers, which narrows down the variance of inferred answers and improve the workers' quality estimation. Thus CrowdChart can achieve the confidence requirement with much less number of tasks. Moreover, we can see that with increase of the confidence level, the cost grows up. This is reasonable because we should ask more to keep higher confidence.

Fig. 3(c) and (d) shows the result on quality. When confidence level is 0.9, we can see from Fig. 3(c) that on dataset `Paper`, CrowdChart achieves the best quality, with the MNAD of 0.74, which improves 30% compared with `T-Crowd` with the second smallest MNAD (1.1). CrowdChart also outperforms `AV` and `GTM` a lot. For instance, when the confidence level is 0.95, CrowdChart has an MNAD of 0.58 while `AV` and `GTM` are 1.23 and 1.03 respectively. `AV` has the worst quality because it does not consider the workers' quality and task's difficulty. `GTM` performs better than `AV` because it considers the task's difficulty. The significant improvement of CrowdChart is attributed to the truth inference techniques, such as answer alignment and worker model.

V. Conclusion

In this paper, we proposed a crowdsourcing framework to extract structured data from charts. We used well-designed tasks to interact with the crowd. We designed a truth inference model to derive accurate answers and early-stopping techniques to reduce the cost. We evaluated the framework on real datasets and the results demonstrate its superiority.

Acknowledgement. This work was supported by the 973 Program of China (2015CB358700), NSFC (61632016, 61602488, U1711261, 61472198, 61521002, 61661166012, 61502503), the Research Funds of RUC (18XNLG18).

References

[1] https://www.mturk.com/.
[2] C. Chai, J. Fan, G. Li, J. Wang, and Y. Zheng. Crowdsourcing database systems: Overview and challenges. In *ICDE 2019*.
[3] C. Chai, J. Fan, G. Li, J. Wang, and Y. Zheng. Crowd-powered data mining. *CoRR*, abs/1806.04968, 2018.
[4] C. Chai, G. Li, J. Li, D. Deng, and J. Feng. Cost-effective crowdsourced entity resolution: A partial-order approach. In *SIGMOD, 2016*.
[5] A. P. Dempster and L. et.al. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 1977.
[6] C. C. et.al. A partial-order-based framework for cost-effective crowdsourced entity resolution. *VLDB J.*, 2018.
[7] M. S. et.al. Revision: automated classification, analysis and redesign of chart images. In *UIST, 2011*.
[8] J. Fan, G. Li, B. C. Ooi, K. Tan, and J. Feng. icrowd: An adaptive crowdsourcing framework. In *SIGMOD, 2015*.
[9] D. Jung, W. Kim, H. Song, J. Hwang, B. Lee, B. H. Kim, and J. Seo. Chartsense: Interactive data extraction from chart images. In *CHI, 2017*.
[10] G. Li and C. C. et.al. CDB: optimizing queries with crowd-based selections and joins. In *SIGMOD, 2017*.
[11] G. Li, J. Wang, Y. Zheng, and M. J. Franklin. Crowdsourced data management: A survey. *TKDE*, 2016.
[12] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *SIGMOD 2014*.
[13] Y. Liu, X. Lu, Y. Qin, Z. Tang, and J. Xu. Review of chart recognition in document images. In *Visualization and Data Analysis 2013*.
[14] C. Shan, N. Mamoulis, G. Li, R. Cheng, Z. Huang, and Y. Zheng. T-crowd: Effective crowdsourcing for tabular data. In *ICDE 2018*.
[15] J. Yang, J. Fan, Z. Wei, G. Li, T. Liu, and X. Du. Cost-effective data annotation using game-based crowdsourcing. *PVLDB*, 12(1):57–70, 2018.
[16] X. Yu, G. Li, Y. Zheng, Y. Huang, S. Zhang, and F. Chen. Crowdota: An online task assignment system in crowdsourcing. In *ICDE 2018*.
[17] B. Zhao and J. Han. A probabilistic model for estimating real-valued truth from conflicting sources. *Proc. of QDB*, 2012.