Minimizing Efforts in Validating Crowd Answers

Nguyen Quoc Viet Hung, Duong Chi Thang, Matthias Weidlich[†], Karl Aberer
École Polytechnique Fédérale de Lausanne and [†] Imperial College London

ABSTRACT

In recent years, crowdsourcing has become essential in a wide range of Web applications. One of the biggest challenges of crowdsourcing is the quality of crowd answers as workers have wide-ranging levels of expertise and the worker community may contain faulty workers. Although various techniques for quality control have been proposed, a post-processing phase in which crowd answers are validated is still required. Validation is typically conducted by experts, whose availability is limited and who incur high costs. Therefore, we develop a probabilistic model that helps to identify the most beneficial validation questions in terms of both, improvement of result correctness and detection of faulty workers. Our approach allows us to guide the expert's work by collecting input on the most problematic cases, thereby achieving a set of high quality answers even if the expert does not validate the complete answer set. Our comprehensive evaluation using both real-world and synthetic datasets demonstrates that our techniques save up to 50% of expert efforts compared to baseline methods when striving for perfect result correctness. In absolute terms, for most cases, we achieve close to perfect correctness after expert input has been sought for only 20% of the questions.

Categories and Subject Descriptors: H.1.2 [User/Machine Systems]: Human information processing
Keywords: crowdsourcing; validation; guiding user feedback; expectation maximization

1. INTRODUCTION

Crowdsourcing has attracted much attention from both academia and industry, due to the high availability of ordinary Internet users (a.k.a. crowd workers) [37]. It has proved to be an efficient and scalable approach to overcome problems that are computationally expensive or unsolvable for machines, but rather trivial for humans. The number of crowdsourcing applications is tremendous, ranging from data acquisition [4], data integration [51], data mining [44], information extraction [17], to information retrieval [50]. To facilitate the development of crowdsourcing applications, more than 70 crowdsourcing platforms such as Amazon Mechanical Turk (AMT) and CrowdFlower have been developed in recent years.

Quality of crowd answers. A common crowdsourcing setup features users that post tasks in the form of questions, which are answered by crowd workers for financial rewards. Here quality control is a major obstacle. Workers have different backgrounds and wide-ranging levels of expertise and motivation [29], so that the collected answers are not always correct. To overcome this issue, tasks are often assigned to multiple workers to aggregate the results. In the presence of faulty workers giving random answers, however, the aggregated answer is not guaranteed to be correct.

The answer validation problem. To increase the trustworthiness of the obtained crowd answers (referred to as an *answer set*), crowdsourcing platforms such as AMT include a validation phase. Crowd answers are validated against the supposedly correct answers given by a human validator (henceforth called *expert*).

Validation of answer by an expert leads to a trade-off between the verified result correctness and the invested effort. The more effort the expert puts into providing answers that can be used to judge correctness of answers from crowd workers, the higher is the quality of the final answer set. Seeking expert input incurs high costs, so that, given the sheer amount of questions to be answered, only a fraction of the answer set can be validated based on the expert's answers. In fact, validating a large part of the crowd answers would negate the benefits of crowdsourcing in the first place.

Contributions. This paper targets the effective utilization of expert efforts in the validation of crowd answers. By (I) *aggregating answers* of crowd workers and (II) *guiding an expert* in the validation process, we go beyond the aforementioned trade-off and reduce the amount of expert efforts needed to achieve the same level of result correctness. Both steps, answer aggregation and expert guidance, are interrelated. On the one hand, answer aggregation exploits the reliability of workers, which is assessed based on the feedback given by an expert as part of the answer validation. On the other hand, an expert is guided based on the potential effect that the validation of a certain answer has for the aggregation of answers.

Answer aggregation.. To aggregate answers from crowd workers, we develop a probabilistic model estimating whether a certain answer is correct. Unlike traditional likelihood estimators which only take into account the answer set, see [23], our estimator is able to achieve higher accuracy by also considering expert input. In particular, the expert input is used to assess the reliability of a worker, formalized as a confusion matrix over the possible answers. The reliability of workers is then exploited to calculate the probability that a certain answer is correct. Moreover, a decision-theoretic measure allows us to conclude on the uncertainty related to an answer set based on the reliability of workers.

Since expert input is sought continuously, it is important to realize answer aggregation as pay-as-you-go process. In our approach,

SIGMOD'15, May 31–June 4, 2015, Melbourne, Victoria, Australia.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2758-9/15/05 ...$15.00.
http://dx.doi.org/10.1145/2723372.2723731 .

this is achieved by updating the model for worker reliability incrementally upon the arrival of new expert input.

Expert guidance.. To guide the validation of crowd answers by an expert, we formally define the problem of effort minimization to reach a validation goal in terms of result correctness. The problem can only be solved when assuming that workers are truthful. Even in that case, which is not realistic, however, the problem is intractable since even a restricted variant of the problem is NP-hard. Hence, we introduce two guidance strategies that cater for complementary aspects of the problem.

The first strategy aims at a maximal improvement of the result correctness. This strategy is motivated by the observation that some answer validations are more beneficial than others. Since workers and tasks are not independent, but connected by the workers' answers, a certain expert input may have a positive effect on the evaluation of the worker reliability and, thus, on the estimated result correctness. We show how a measure for the expected benefit of a validation question can be used to guide an expert.

The second strategy focuses on the detection of faulty workers (e.g. spammers), which can account for up to 40% of the worker community [29]. Faulty workers increase the cost of acquiring correct answers and contaminate the answer set by adding uncertainty. We address these issues by estimating the likelihood of a worker to be a spammer based on answer validations and show how an expert can be guided to detect faulty workers.

Both strategies have different strengths, so that we also present a hybrid approach that combines the two strategies dynamically.

We evaluated the developed approach with multiple real-world and synthetic datasets. Our techniques save up to 50% of expert efforts compared to a baseline method when striving for perfect result correctness. For most cases, we achieve close to perfect correctness with expert input on only 20% of the questions. Also, the explicit integration of answer validations as realized by our techniques is twice as effective in increasing result correctness compared to the traditional approach of integrating expert input as crowd answers. Moreover, we demonstrate robustness of the approach against erroneous expert input and show that, by distributing a cost budget between crowd workers and experts, it achieves high correctness while satisfying completion time and budget constraints.

The rest of the paper is organized as follows. Next we discuss characteristics of crowd workers and motivate the need to have their answers validated by an expert. §3 defines a formal model for crowdsourcing and gives an overview of our approach. The details on the proposed techniques are given subsequently: §4 introduces our method for probabilistic answer aggregation; §5 defines the problem of expert efforts minimization and presents heuristics to approximate a solution. Evaluation results are presented in §6, before we summarize related work in §7 and conclude in §8.

2. BACKGROUND

Crowdsourcing. Crowdsourcing is an efficient and scalable methodology that enables human computation for micro tasks [37]. Such tasks are broadly classified based on the function that is applied by crowd workers to some objects: In *discrete* tasks each object is assigned to a label [23]; *continuous* tasks require objects to be assigned to real values [48]; *partial-function* tasks define objects as rules [4]; in *similarity* tasks, each object is a pair of matches [19].

In this paper, we focus on discrete tasks, also known as *classification* tasks. They are the core of many applications such as training classifiers [44], entity extraction [10], sentiment analysis [33], and credibility evaluation [21]. As an example, we consider a classification task in which workers need to assign a label to an object, as it is done for tagging images, categorizing websites, or answering multiple-choice questions. Table 1 illustrates an exemplary crowdsourcing result, in which five workers (W_1 - W_5) assigned one out of four labels (1 - 4) to four objects (o_1 - o_4). The correct label assignments are shown in a separate column.

Table 1: Labels provided by 5 workers for 4 objects

	W_1	W_2	W_3	W_4	W_5	Correct	Majority Voting
o_1	2	3	2	2	3	2	2
o_2	3	2	3	2	3	3	3
o_3	1	4	1	4	3	1	1 or 4
o_4	4	1	2	1	3	2	1

Crowd workers. The quality of the result of a crowdsourcing task highly depends on the performance of the crowd workers. Previous studies [29] characterized different types of crowd workers to reflect their expertise: (1) Reliable workers have deep knowledge about specific domains and answer questions with very high reliability; (2) Normal workers have general knowledge to give correct answers, but make mistakes occasionally; (3) Sloppy workers have very little knowledge and thus often give wrong answers, but unintentionally; (4) Uniform spammers intentionally give the same answer for all questions; (5) Random spammers carelessly give random answers for all questions. Figure 1 illustrates the relation between worker types and the quality of crowdsourcing results for the simple case of binary classification tasks. Here, result quality is measured in terms of sensitivity (the proportion of positives that are correctly classified) and specificity (the proportion of negatives that are correctly classified). Sloppy workers, uniform and random spammers are problematic, as they increase the cost of obtaining a correct classification result.

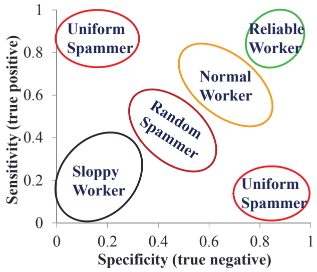

Figure 1: Characterization of worker types

For the above example given in Table 1, for instance, worker W_1 would be considered a normal worker (three out of four answers are correct), W_3 is a reliable worker (all answers are correct), whereas W_5 is a uniform spammer (same answer to all questions).

The need for answer validation. In practice, submitters of crowdsourcing tasks have limited control over the selection of crowd workers and little insights into the level of expertise and reliability of the workers that provided answers. Hence, tasks are often assigned to multiple workers to aggregate the results. Various methods for answer aggregation and estimation of worker reliability have been proposed in the literature. However, the results of automatic methods are inherently uncertain, since they are heuristic-based and no technique performs well in the general case [22].

The example in Table 1 illustrates an inconsistent label assignment due to different levels of expertise of workers. For instance, three out of four possible labels are assigned for object o_3, whereas all four possible labels are provided for object o_4. Yet, the popular approach of aggregating results by 'Majority Voting' would return only a partially correct result for the example.

State-of-the-art answer validation. To overcome the inherent uncertainty of crowdsourcing results, many crowdsourcing platforms

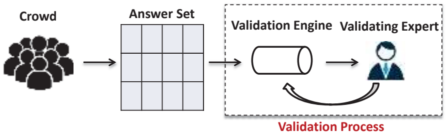

Figure 2: A simple answer validation process

such as AMT include a validation phase as depicted in Figure 2. This process features a validator (also referred to as a validating *expert*) that provides trustworthy answers. The integration of trustworthy input from experts is, in many cases, more efficient than simply enlarging the number of considered crowd workers. In fact, our evaluation in §6 shows that the inclusion of expert input, even though it is more expensive then additional input by crowd workers, is preferable in all but extreme cases (e.g., when the expert is more than 100 times more expensive than crowd workers).

Expert input is commonly considered to be correct, not only in crowdsourcing, but also in related fields, such as correction of errors in databases [49] or in active learning [5]. Then, expert input provides a ground truth for the assessment of the crowd answers. As part of our evaluation in §6, we empirically show that it is indeed reasonable to assume that experts provide correct answers. Yet, we later also investigate cases where expert input may include a certain amount of incorrect answers.

Although most crowdsourcing platforms acknowledge the need for validation, they only provide rudimentary support for the validation phase. The state-of-the-art in answer validation confronts the validating expert with the raw answer set data, complemented by simple statistics of the distribution of answer values [3, 2]. As such, the process of aggregating and validating answers from crowd workers is largely *unguided*. This is an issue given that the effort budget for validation is limited and, without guidance, validation effort is likely to be wasted on answers that have a limited potential for increasing the correctness of the overall result.

For the example in Table 1, the validation of 2 being the correct label for object o_1, for instance, would allow for assessing workers W_1, W_3 and W_4 as reliable. Feedback on object o_4 would be more beneficial, though, as it helps to identify W_3 as a reliable worker, who indeed labeled all objects correctly.

Against this background, our work is the first to propose a method for answer validation that combines crowd answers with expert feedback for pay-as-you-go quality control. With the goal to minimize validation efforts, we get the best of both worlds: the cost of crowdsourcing is lower than having an expert answering all questions, whereas answer validation increases the result correctness.

3. MODEL AND APPROACH

This section presents a crowdsourcing model and, based thereon, gives an overview of our overall approach to answer validation.

3.1 Model

We model the setting of crowdsourcing by a set of k workers $W = \{w_1, w_2, ..., w_k\}$ that provide answers for a set of n objects $O = \{o_1, o_2, ..., o_n\}$. Let $L = \{l_1, l_2, ..., l_m\}$ be a set of labels. Then, crowd answers are modeled as an $n \times k$ *answer matrix*:

$$\mathcal{M} = \begin{pmatrix} x_{11} & \dots & x_{1k} \\ \dots & \dots & \dots \\ x_{n1} & \dots & x_{nk} \end{pmatrix}$$

where $x_{ij} \in (L \cup \{\ominus\})$ for $1 \le i \le n$, $1 \le j \le k$. Here, the special label $\ominus$ denotes that a worker did not assign a label to an object. We write $\mathcal{M}(o, w)$ to denote the answer of worker w for object o.

Based on the above notions, we define an *answer set* as a quadruple $N = \langle O, W, L, \mathcal{M} \rangle$ where O is a set of objects, W a set of workers, L a set of labels, and $\mathcal{M}$ an answer matrix.

Expert input is modeled by an *answer validation function* $e : O \to (L \cup \{\ominus\})$ that assigns labels to objects. Again, the label $\ominus$ denotes that the expert has not yet assigned a label to an object.

Our model includes the reliability of workers by means of a *confusion matrix* over labels. For a worker $w \in W$ and a set of labels $L = \{l_1, l_2, ..., l_m\}$, there is an $m \times m$ confusion matrix $\mathcal{F}_w$, such that $\mathcal{F}_w(l, l') \in [0, 1]$ denotes the probability that the worker w assigns the label l' to an object for which the correct label is l.

Further, our work employs a probabilistic aggregation of crowd answers. For each combination of a label and an object, our model includes an assignment probability. For $O = \{o_1, o_2, ..., o_n\}$ as the set of objects and $L = \{l_1, l_2, ..., l_m\}$ as the set of labels, a probabilistic assignment is captured by an $n \times m$ *assignment matrix* $\mathcal{U}$. Here, $\mathcal{U}(o, l) \in [0, 1]$ denotes the probability that $l \in L$ is the correct label for object $o \in O$ and we require that the matrix defines a probability distribution for each object, i.e., $\sum_{l \in L} \mathcal{U}(o, l) = 1$.

Combining the above notions, a *probabilistic answer set* is a quadruple $P = \langle N, e, \mathcal{U}, \mathcal{C} \rangle$ where $N = \langle O, W, L, \mathcal{M} \rangle$ is an answer set, e is an answer validation function, $\mathcal{U}$ is an assignment matrix, and $\mathcal{C} = \bigcup_{w \in W} \{\mathcal{F}_w\}$ is a set of confusion matrices.

The actual result of the crowdsourcing process is a *deterministic assignment*, a function $d : O \to L$ assigning labels to objects.

3.2 The overall approach to answer validation

Validation process. Validation happens iteratively, such that in each step, an expert asserts the correct label for an object. This process halts either when reaching a *validation goal* or upon consumption of an *expert efforts budget*. The former relates to the desired quality of the result assignment, e.g., a threshold on the estimated correctness of the deterministic assignment. Since expert input is a scarce resource, the latter defines an upper bound for the number of validations and, thus, iterations of the validation process.

Starting with an answer set $N = \langle O, W, L, \mathcal{M} \rangle$, the validation process continuously updates a deterministic assignment that is considered to be correct. Each iteration of the validation process comprises the following steps:

(1) *select* an object o for which expert feedback shall be sought;
(2) *elicit* expert input on the label of object o and update $e(o)$,
(3) *conclude* the consequences of the expert input on the probabilistic answer set P;
(4) *filter* the deterministic assignment d assumed to be correct based on the probabilistic answer set P.

Instantiations of the general validation process differ in their implementation of steps (1), (3), and (4). For instance, a simple manual validation process is emulated as follows: an object is randomly *selected*; as part of the *conclusions*, the probability of the object for which feedback has been sought is updated; *filtering* selects, for all objects, the labels with highest assignment probability.

Validation framework. Using the notions introduced above, Figure 3 presents an overview of our overall approach. An initial answer set is built from the workers' responses, which is then used to construct a probabilistic answer set by means of *Answer Aggregation* under consideration of the worker reliability. Based on a probabilistic answer set and the input sought from the validating expert, we can automatically derive a deterministic assignment to be used by crowdsourcing applications, which is referred to as *Instantiation*. The quality of the deterministic assignment depends on the degree of uncertainty in the probabilistic answer set. This uncertainty stems from the decision whether to trust certain workers and select their answers when computing the assignment. *Expert*

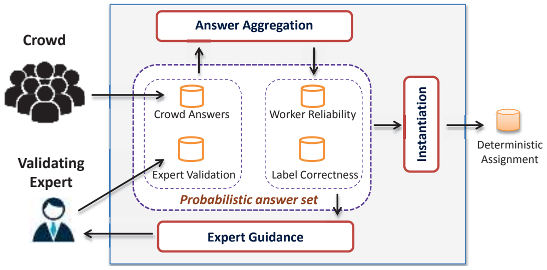

Figure 3: Framework for guided answer validation

Guidance helps to resolve the uncertainty by selecting and ranking candidate objects to seek expert input. This closes the cycle since the answer validation leads to a new assessment of the worker reliability and, thus, a new probabilistic answer set. Hence, the probabilistic answer set is updated in a pay-as-you-go process, where a deterministic assignment can be instantiated at any time.

There is the following relation between the components of the framework in Figure 3 and the validation process:

Answer Aggregation. This component assesses the reliability of workers and, based thereon, computes a probabilistic assignment of labels to objects. As such, it corresponds to step *conclude* in the validation process and creates the probabilistic answer set. The realization of this component is detailed in §4.

Expert Guidance. To guide the uncertainty reduction step, this component selects and ranks objects for which expert feedback should be sought. Hence, this component realizes step *select* in the validation process, for which the details are given in §5.

Instantiation. This component creates the deterministic assignment from the probabilistic answer set, realizing step *filter* in the validation process. It is implemented as the selection of the label with the highest probability in the assignment matrix for each object.

4. PROBABILISTIC ANSWER AGGREGATION

Given the answer set provided by the workers, a probabilistic answer set is constructed by assessing the worker reliability and computing the probabilistic assignment of labels to objects. We first describe the construction of a probabilistic answer set (§4.1) and then turn to a measure for the answer set uncertainty (§4.2).

4.1 Construction of a probabilistic answer set

In the construction of a probabilistic answer set, we consider the following aspects:

Expert validations. The expert input provides the supposedly correct labels for some of the objects. It helps not only to ensure correctness of the final deterministic assignment, but also allows for identifying reliable workers.

Worker Reliability. We expect label assignments done by reliable workers to be mostly correct, whereas unreliable workers provide mostly incorrect assignments. Yet, the level of reliability varies between workers and is not known apriori.

Assignment correctness. For each combination of labels and objects, we have to consider the possibility that the respective assignment is correct. Clearly, the correctness of such an assignment is not known except for those that have been obtained from the expert, but we can expect reliable workers to provide mostly correct assignments.

There is a mutually reinforcing relationship between workers and objects: one worker can label multiple objects and one object can be labeled by multiple workers. Aiding this relationship, expert validations provide a means to judge both, the reliability of workers and the correctness of label assignments. Therefore, we approach the construction of a probabilistic answer set using the Expectation-Maximization methodology [9], which allows for concurrent estimation of worker reliability and assignment correctness.

Requirements for Expectation-Maximization (EM). To be useful in our setting, an EM algorithm must meet two requirements. First, expert validations should be first class citizens to fully leverage their benefits. By assessing the worker reliability based on expert validations, the computation of the assignment probabilities is not limited to the objects for which expert input has been received.

Second, each iteration of the validation process changes the reliability of workers (via the confusion matrices) and the probabilistic assignment matrix only marginally. Hence, EM should proceed incrementally and avoid expensive re-computation of the worker reliability and the assignment probabilities in each iteration.

Existing Expectation-Maximization (EM) algorithms. Various EM algorithms have been proposed in the literature. Yet, none of the existing algorithms meets the aforementioned requirements. Traditional EM [23] operates in batch mode. Hence, the confusion matrices that capture worker reliability and the probabilistic assignment matrix would be re-computed starting from a random probability estimation every time new expert input arrives.

Algorithms for online EM [6], in turn, do not support the integration of a ground truth, i.e., the expert validations. They target incremental updates when the answer matrix changes (a new answer arrives), whereas our setting requires incremental updates over an unchanged answer matrix whenever the ground truth is extended (a new expert validation arrives). One may argue that expert validations may be modeled as new answers. However, this does not circumvent the issue since answers representing expert validations could be dominated by incorrect answers (e.g., due to Majority Voting). We later show experimentally that the explicit integration of expert validations, rather than considering them as ordinary answers, indeed yields more effective answer aggregation.

The *i*-EM algorithm. Against this background, we propose an incremental EM algorithm, called *i*-EM algorithm, that follows the view maintenance principle [7]. Estimation of worker reliability and assignment correctness is grounded in the results of the previous iteration of the validation process, which avoids re-computing them in each iteration. This does not only increase efficiency of the approach, but, as will be illustrated in our evaluation, also yields a better approximation compared to a random probability estimation.

The *i*-EM algorithm implements the *conclude* function of the validation process. In the s-th iteration of the expert validation process, the input of this function is given by the answer set N and the expert validation function e_s as it has been updated with the expert input received in the s-th iteration. However, the *i*-EM algorithm works incrementally, so that, in each iteration, we also maintain the state of the previous iteration in terms of the probabilistic answer set. That is, in the s-th iteration of the expert validation process, the probabilistic answer set $P_{s-1} = \langle N, e_{s-1}, \mathcal{U}_{s-1}, \mathcal{C}_{s-1}\rangle$ of the previous iteration is also part of the input to the algorithm. The algorithm then returns a new probabilistic answer set $P_s = \langle N, e_s, \mathcal{U}_s, \mathcal{C}_s\rangle$.

The actual Expectation-Maximization done in the s-th iteration of the expert validation step is also iterative, alternating between two steps, the Expectation step (E-step) and the Maximization step (M-step), until convergence. As part of EM iterations, the probabilistic assignment as well as the worker confusion matrices of the probabilistic answer set are updated. That is, in the s-th iteration of the expert validation process, the EM iterations create a sequence $\mathcal{U}_s^0, \mathcal{U}_s^1, \ldots, \mathcal{U}_s^z$ of assignment matrices and a sequence $\mathcal{C}_s^0, \mathcal{C}_s^1, \ldots, \mathcal{C}_s^z$ of sets of confusion matrices.

E-step. In an E-step, we estimate the assignment probabilities based on the worker confusion matrices. The first E-step of the s-th iteration of the expert validation process is based on the worker confusion matrices $\mathcal{C}_s^0 = \{\mathcal{F}_{w_1,s}^0, \mathcal{F}_{w_2,s}^0, \ldots, \mathcal{F}_{w_k,s}^0\}$, which are given as input from the previous iteration of the validation process, i.e., $\mathcal{C}_s^0 = \mathcal{C}_{s-1}^q = \{\mathcal{F}_{w_1,s-1}^q, \mathcal{F}_{w_2,s-1}^q, \ldots, \mathcal{F}_{w_k,s-1}^q\}$ with q being the number of EM iterations in the $(s-1)$-th iteration of the validation process. In the τ-th E-step of the s-th step of the validation process, for each object o for which no expert input has been received ($e_s(o) = \ominus$), the assignment probability of label l is estimated as:

$$\mathcal{U}_s^\tau(o,l) = \frac{p_s^{\tau-1}(l) \prod_{w \in W} \prod_{l' \in L} \left(\mathcal{F}_{w,s}^{\tau-1}(l,l')\right)^{d_w(o,l')}}{\sum_{l'' \in L} p_s^{\tau-1}(l'') \prod_{w \in W} \prod_{l' \in L} \left(\mathcal{F}_{w,s}^{\tau-1}(l'',l')\right)^{d_w(o,l')}} \quad (1)$$

where d_w is a function encoding the existence of the respective label assignment by a worker $w \in W$ in the answer matrix $\mathcal{M}$,

$$d_w(o,l) = \begin{cases} 1 & \text{if } \mathcal{M}(o,w) = l \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

and $p_s^{\tau-1}(l)$ is the prior probability of label l, given as

$$p_s^{\tau-1}(l) = \frac{\sum_{o \in O} \mathcal{U}_s^{\tau-1}(o,l)}{|O|}. \quad (3)$$

For objects for which an expert validation has been received already ($e_s(o) \neq \ominus$), assignment probabilities are trivially given as:

$$\mathcal{U}_s^\tau(o,l) = \begin{cases} 1 & \text{if } e_s(o) = l \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

M-step. In an M-step, we estimate the confusion matrices of the workers using the assignment probabilities. That is, in the τ-th M-step of the s-th step of the validation process, the probability $\mathcal{F}_{w,s}^\tau(l',l)$, which captures that worker w assigns the label l when the correct label is l', is computed as follows:

$$\mathcal{F}_{w,s}^\tau(l',l) = \frac{\sum_{o \in O} \mathcal{U}_s^\tau(o,l') d_w(o,l)}{\sum_{l'' \in L} \sum_{o \in O} \mathcal{U}_s^\tau(o,l') d_w(o,l'')} \quad (5)$$

4.2 The uncertainty of answer aggregation

The heterogeneity among the workers renders it likely that many objects, which are supposed to have a single correct label, are assigned to different labels by the workers. The model of a probabilistic answer set, as constructed by the i-EM algorithm introduced above, provides us with a truthful representation of the uncertainty related to the aggregation of the answers. To guide an expert in the validation process, the uncertainty needs to be quantified.

Let $P = \langle N, e, \mathcal{U}, \mathcal{C} \rangle$ be a probabilistic answer set constructed for answer set $N = \langle O, W, L, \mathcal{M} \rangle$. Recall that P defines an assignment $\mathcal{U}(o,l)$ for each label $l \in L$ and object $o \in O$, which represents the likelihood of l to be the correct label for o. Since the probabilities of the labels form a distribution, i.e., $\sum_{l \in L} \mathcal{U}(o,l) = 1$, we can model each object o as a random variable. Then, the overall uncertainty of the probabilistic answer set is computed by the Shannon entropy [43] over a set of random variables. More precisely, the entropy of an object o is measured as follows:

$$H(o) = -\sum_{l \in L} \mathcal{U}(o,l) \times \log(\mathcal{U}(o,l)) \quad (6)$$

The entropy of an object is the basis for the computation of the uncertainty of the probabilistic answer set P. It is defined as the sum of the entropies of all objects:

$$H(P) = \sum_{o \in O} H(o) \quad (7)$$

The entropy of an object and, thus, also of the probabilistic answer set, can only be 0, if all assignment probabilities are equal to 1 or 0. If so, there is a clear separation of correct and incorrect assignments for an object or all objects, respectively.

5. EXPERT VALIDATION GUIDANCE

This section presents techniques to guide an expert in the validation process that reduces the uncertainty of a probabilistic answer set. We first formalize the problem of effort minimization (§5.1). As the problem can be solved only under further assumptions on crowd workers and is computational hard, we present two heuristic solutions aiming at a maximal uncertainty reduction (§5.2) or the detection of faulty workers (§5.3), respectively. Then, we combine both heuristics (§5.4) and also elaborate on how to handle potentially erroneous expert input (§5.5).

5.1 The effort minimization problem

Instantiation of the generic answer validation process described in §3.2 requires the definition of a validation goal. For the answer aggregation introduced above, a reasonable validation goal is grounded in the uncertainty measure defined in §4.2.

Given the iterative nature of the validation process, we would like to minimize the number of necessary expert interaction steps for a given goal. For an answer set $N = \langle O, W, L, \mathcal{M} \rangle$, executing the answer validation process leads to a sequence of deterministic assignments $\langle d_0, d_1, \ldots, d_n \rangle$, termed a *validation sequence*, where d_i represents the assignment obtained after the i-th iteration. Given an expert efforts budget b and a validation goal Δ, we refer to sequence $\langle d_0, d_1, \ldots, d_n \rangle$ as being *valid*, if $n \leq b$ and d_n satisfies Δ. Let $\mathcal{R}(\Delta, b)$ denote a finite set of valid validation sequences that can be created by instantiations of the validation process. Then, a validation sequence $\langle d_0, d_1, \ldots, d_n \rangle \in \mathcal{R}(\Delta, b)$ is *minimal*, if for any validation sequence $\langle d'_0, d'_1, \ldots, d'_m \rangle \in \mathcal{R}(\Delta, b)$ it holds that $n \leq m$.

PROBLEM 1 (EXPERT EFFORTS MINIMIZATION). *Let $\langle O, W, L, \mathcal{M} \rangle$ be an answer set and $\mathcal{R}(\Delta, b)$ a set of valid validation sequences for an expert efforts budget b and a goal Δ. The problem of* expert efforts minimization *is the identification of a minimal sequence $\langle d_0, d_1, ..., d_n \rangle \in \mathcal{R}(\Delta, b)$.*

Assuming that the validation goal is defined in terms of the uncertainty of the probabilistic answer set, solving Problem 1 is challenging. First, the objects are not independent, due to the mutual reinforcing relationship between workers and objects. Validating one object can affect the uncertainty of label assignment of other objects. Second, the presence of malicious workers can alter the uncertainty of the answer set, as incorrect labels can be mistreated as correct labels and vice-versa. Further, even in the absence of faulty workers, finding an optimal solution requires investigation of all permutations of all subsets (with size $\leq b$) of objects, which is intractable. Appendix E outlines that even for a restricted version of the problem, finding an optimal solution is NP-hard.

5.2 Uncertainty-driven expert guidance

Our first heuristic to guide the selection of objects for validation aims at the maximal uncertainty reduction under the assumption of ethical workers. It exploits the contribution of a single validation using the notion of information gain from information theory [42].

First, we define a conditional variant of the entropy measure introduced earlier. It refers to the entropy of the probabilistic answer set $P = \langle N, e, \mathcal{U}, \mathcal{C} \rangle$, $N = \langle O, W, L, \mathcal{M} \rangle$, conditioned on the expert input on object o. Informally, it measures the expected entropy of P under a certain expert assignment.

$$H(P \mid o) = \sum_{l \in L} \mathcal{U}(o,l) \times H(P_l) \quad (8)$$

where $P_l = conclude(N, e')$ is constructed by the *i*-EM algorithm with $e'(o) = l$ and $e'(o') = e(o')$ for $o' \in (O \setminus \{o\})$.

To take a decision on which object to select, we assess the expected difference in uncertainty before and after the expert input for an object. The respective change in entropy is the information gain that quantifies the potential benefit of knowing the true value of an unknown variable [42], i.e., the correct label in our case:

$$IG(o) = H(P) - H(P \mid o). \tag{9}$$

The information gain allows for selection of the object that is expected to maximally reduce the uncertainty of the probabilistic answer set in one iteration of the validation process. This is formalized by a selection function for uncertainty-driven expert guidance:

$$select_u(O') = \underset{o \in O'}{\arg\max}\ IG(o) \tag{10}$$

5.3 Worker-driven expert guidance

Uncertainty-driven expert guidance as introduced above assumes that workers are ethical, an assumption that is often violated in practice. Recent studies found that up to 40% of the workers in a worker community may be faulty (e.g. spammers) [29]. This section thus presents a technique for expert guidance that aims at the detection of the three problematic worker types discussed in §2, i.e., uniform spammers, random spammers, and sloppy workers.

Detecting uniform and random spammers. To assess the likelihood of a worker being a uniform or random spammer, we leverage the fact that the labels provided by random spammers tend to be uniformly distributed across the correct labels, whereas the labels provided by uniform spammers are all the same. These tendencies are directly visible in the confusion matrix.

Consider two workers $\mathcal{A}$ and $\mathcal{A}'$, who provide answers for six objects from a set of two labels $\{T, F\}$, as listed in Table 2. For worker $\mathcal{A}$, the numbers of T labels and F labels are equal for objects for which the correct label is T (or F, respectively). Consequently, rows in the confusion matrix tend to have equivalent values across columns, which indicates that $\mathcal{A}$ is a random spammer. For worker $\mathcal{A}'$, the confusion matrix has only a single column with values larger than 0. Thus, $\mathcal{A}'$ is likely to be a uniform spammer.

Table 2: Answers and confusion matrices for workers $\mathcal{A}$ and $\mathcal{A}'$

	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8
Correct	T	T	F	F	T	F	T	F
$\mathcal{A}$	T	F	T	F	T	F	F	T
$\mathcal{A}'$	F	F	F	F	F	F	F	F

	Worker $\mathcal{A}$	
	T	F
T	0.5	0.5
F	0.5	0.5

	Worker $\mathcal{A}'$	
	T	F
T	0	1
F	0	1

Based on this observation, we rely on a variant of the *spammer score* proposed in [38] to estimate the probability that a worker is a uniform or random spammer. In [38], the confusion matrices are constructed from the labels that are estimated to be correct, which introduces a bias if this estimation is incorrect. Therefore, we construct the confusion matrices only based on the answer validations.

Confusion matrices that have rows with equivalent values across columns (random spammers) or a single column with values larger than 0 (uniform spammers) have similar characteristics as a rank-one matrix. Therefore, we calculate the spammer score $s(w)$ of a worker w as the distance of the confusion matrix to its closest rank-one approximation, using the Frobenius norm:

$$s(w) = \min_{\hat{\mathcal{F}}_w} \|\mathcal{F}_w - \hat{\mathcal{F}}_w\|_F \tag{11}$$

where $\mathcal{F}_w$ is the confusion matrix of worker w and $\hat{\mathcal{F}}_w$ is a matrix with rank one. This low-rank approximation problem can be solved using singular value decomposition [14]. We then set a threshold τ_s to filter uniform and random spammers from the population.

Detecting sloppy workers. Sloppy workers tend to provide labels incorrectly, which is also detected based on the confusion matrix. Following the above approach for uniform and random spammers, we construct a confusion matrix using the answer validations. As the labels provided by the sloppy workers are mostly incorrect, we can calculate the error rate of the worker. The error rate of a worker (denoted as e_w) is the sum of all values not on the main diagonal of the confusion matrix weighted by the priors of the labels. If this error rate e_w is larger than a threshold τ_p, the worker is considered as a sloppy worker.

Expert guidance. We exploit the detection techniques to guide the answer validation by selecting objects that will contribute to the identification of faulty workers. To this end, we measure the benefit of expert input on an object by the expected number of detected faulty workers. Formally, by $R(W \mid o = l)$, we denote the expected number of detected faulty workers, if the answer validation indicates that l is the correct label for object o.

$$R(W \mid o = l) = \mid \{w \mid s(w) < \tau_s\} \cup \{w \mid e_w > \tau_p\} \mid \tag{12}$$

Then, the total expected number of detected faulty workers for input on o is

$$R(W \mid o) = \sum_{l \in L} \mathcal{U}(o, l) \times R(W \mid o = l) \tag{13}$$

Hence, in each iteration of the answer validation process, the worker-driven expert guidance heuristic will select the object o with the highest total expected number of detected faulty workers, formalized by the following selection function:

$$select_w(O') = \underset{o \in O'}{\arg\max} R(W \mid o) \tag{14}$$

Handling faulty workers. A naive way to handle faulty workers is to define a threshold and exclude any worker with a spammer score higher than the threshold. However, this approach may mistakenly remove truthful workers, as illustrated by the example given in Table 3. Assuming that answer validations have been obtained only for $o_1, \ldots, o_4$, the confusion matrix would indicate that worker $\mathcal{B}$ is a random spammer, even though 4 out of 6 questions have been answered correctly. Hence, workers may be excluded too early if only a few of their answers are considered in the spammer score due to a small number of answer validations.

Table 3: Answer and confusion matrix of worker $\mathcal{B}$

	o_1	o_2	o_3	o_4	o_5	o_6
Correct	T	T	F	F	T	T
$\mathcal{B}$	T	F	T	F	T	T

	T	F
T	0.5	0.5
F	0.5	0.5

We overcome this issue by only excluding the answers of suspected faulty workers from the answer set, while continuing to collect their answers. Then, as more expert input becomes available, these answers are included again once the spammer score is higher than a threshold. In other words, any of the worker answers will be eventually be included if they are truly reliable.

5.4 A combined approach to expert guidance

There is a trade-off between the application of the uncertainty-driven and the worker-driven expert guidance. Focusing solely on uncertainty reduction may lead to contamination of the truthful workers' responses by faulty workers. On the other hand, an excessively worker-driven approach is undesirable as it may increase the overall expert efforts significantly. Therefore, we propose a dynamic weighting procedure that, in each iteration of the answer validation process, helps to choose among the two strategies.

Weighting procedure. Intuitively, there are two factors which affect the choice between the strategies:

Ratio of spammers. If a high number of faulty workers is detected, the worker-driven strategy is preferred. However, as this strategy depends on expert input, it may not be effective in the beginning when the number of answer validations is small. In this case, the uncertainty-driven strategy is favored.

Error rate. The deterministic assignment d_i captures the assignments considered to be correct in the i-th iteration of the answer validation process. If d_i turns out to be mostly incorrect, we have evidence of faulty workers in the community and, thus, favor the worker-driven strategy.

We balance both factors by combining the two strategies dynamically. In the beginning, with a low number of answer validations, it is mainly the error rate of the deterministic assignment that determines which strategy to use. At later stages, the number of detected faulty workers becomes the dominant factor.

To formalize this intuition, we denote the ratio of detected faulty workers in the i-th iteration of the answer validation process by r_i. The error rate of the deterministic assignment is computed by comparing the expert input for object o in the i-th iteration with the label l that has been assigned to o in d_{i-1}, i.e., in the previous iteration. Here, we leverage the probability $\mathcal{U}_{i-1}(o,l)$ of the probabilistic answer set $P_{i-1} = \langle N, e_{i-1}, \mathcal{U}_{i-1}, \mathcal{C}_{i-1} \rangle$, $N = \langle O, W, L, \mathcal{M} \rangle$, of the $(i-1)$-th iteration of the answer validation process. Given the answer validation that assigns l to o in the i-th iteration, the error rate is computed as:

$$\varepsilon_i = 1 - \mathcal{U}_{i-1}(o,l)$$

Using the ratio of detected faulty workers r_i and the error rate ε_i, we compute a normalized score ($\in [0,1]$) for choosing the worker-driven strategy:

$$z_i = 1 - e^{-(\varepsilon_i(1-f_i)+r_i f_i)} \tag{15}$$

where $f_i = \frac{i}{|O|} \in [0,1]$ is the ratio of answer validations. This score mediates the trade-off between the error rate ε_i and the ratio of spammers r_i by the ratio of answer validations f_i. When the ratio f_i is small, the ratio of spammers has less influence and the error rate is the dominant factor. When the ratio f_i is large, the ratio of spammers becomes a more dominant factor.

Hybrid answer validation procedure. Instantiating the general answer validation process described in §3.2, the answer validation process that incorporates both uncertainty-driven and worker-driven expert guidance is defined in Algorithm 1.

Selection of an object for which expert feedback shall be sought is done either by the worker-driven or the uncertainty-driven selection strategy (*select*$_w$ or *select*$_u$). The actual choice is realized by comparing factor z_i to a random number, thereby implementing a roulette wheel selection [18]. Thus, even if factor z_i assumes a large value, there is a chance that the uncertainty driven strategy is chosen. Once expert feedback has been elicited (line 9), the error rate is computed (line 10). Then, we run the method for detecting faulty workers (line 11). The workers detected in this step are handled if the worker-driven strategy had been selected (line 12). Further, the ratio of unethical workers r_i is calculated to compute score z_{i+1} (lines 13-14), used in the next iteration to choose between the selection strategies. Finally, we integrate the feedback by updating the answer validation function e_{i+1} (line 15), computing the probabilistic answer set P_{i+1} with the function *conclude* that implements probabilistic answer aggregation as defined in §4 (line 16), and updating the deterministic assignment set function d_{i+1} (line 17). For objects for which no expert input has been received, the correct assignment is estimated based on the probabilistic answer set using the function *filter*, as discussed in §3.2. The filtered assignments, together with the answer validations, define the deterministic assignment assumed to be correct at this validation step.

Algorithm 1: The hybrid answer validation process

```
   input  : an answer set $N = \langle O, W, L, \mathcal{M} \rangle$,
            a validation goal $\Delta$,
            an expert efforts budget $b$.
   output:  the result assignment $d$.
   // Initialization
1  $e_0 \leftarrow (o \mapsto \ominus, o \in O)$;
2  $P_0 \leftarrow conclude(N, e_0)$;
3  $d_0 \leftarrow filter(P_0)$;
4  $i, z_0 \leftarrow 0$;
5  while not $\Delta \wedge i \leq b$ do
       // (1) Select an object
6      $x \leftarrow random(0,1)$;
7      if $x < z_i$ then $o \leftarrow select_w(\{o' \in O \mid e_i(o') = \ominus\})$ ;
8      else $o \leftarrow select_u(\{o' \in O \mid e_i(o') = \ominus\})$ ;
       // (2) Elicit expert input
9      Elicit expert input $l \in L$ on $o$;
10     Calculate error rate $\varepsilon_i$;
       // (3) Handle spammers
11     Detect spammers;
12     if $x < z_i$ then Handle detected spammers ;
13     Calculate ratio of spammers $r_i$;
14     $z_{i+1} = 1 - e^{-\left(\varepsilon_i\left(1-\frac{i}{|O|}\right)+r_i\frac{i}{|O|}\right)}$;
       // (4) Integrate the answer validation
15     $e_{i+1} \leftarrow (o \mapsto l \wedge o' \mapsto e_i(o'), o' \in O, o' \neq o)$ ;
16     $P_{i+1} \leftarrow conclude(N, e_{i+1})$;
17     $d_{i+1} \leftarrow (o' \mapsto filter(P_{i+1}), o' \in O, e_{i+1}(o') = \ominus \wedge o' \mapsto e_{i+1}(o'), o' \in$
       $O, e_{i+1}(o') \neq \ominus)$ ;
18     $i \leftarrow i+1$;
19 return $d_i$;
```

Implementation. A practical implementation of the hybrid answer validation process must cope with the complexity of the computation of the information gain and the expected spammer score for each object (as part of step (1)). Therefore, to achieve an efficient implementation, we consider two techniques:

Parallelization The computations of the information gain and the expected spammer score for different objects are independent and, therefore, can be executed in parallel for all objects.

Sparse matrix partitioning Due to the implied cognitive load, workers answer a limited amount of questions. Hence, the answer matrix is sparse when having a large number of objects [25]. Therefore, we use sparse matrix partitioning [28] to divide a large answer matrix into smaller dense ones that fit for human interactions and can be handled more efficiently.

5.5 Erroneous answer validations

As detailed in §2, it is commonly assumed that the answers provided by the validating expert are correct. Yet, in practice, expert input may contain mistakes, caused not by the lack of knowledge of the expert, but stemming from the interaction as part of the validation [39]. In other words, if such erroneous answer validations are detected, they can be fixed by the expert themselves.

There are two types of erroneous answer validations: (1) the crowd is right, i.e., the aggregated answer is correct, whereas the expert validation is wrong; (2) the crowd is wrong, but the answer validation is also wrong. As illustrated later in our evaluation, case (1) is unlikely to happen since a validating expert is confronted with statistics about crowd answers, so that a decision to deviate from the aggregated answer is typically taken well-motivated. Case (2), however, is more likely to happen since an expert is more likely to confirm the aggregated answer than to deliberately deviate from it.

We cater for erroneous answer validations as in case (2) by augmenting the answer validation process with a lightweight confirmation check. This check is triggered after a fixed number of iterations of the validation process and proceeds as follows:

(I) For every object o for which expert input has been sought, a deterministic assignment $d_{\sim o}$ is constructed based on the answer set N and the expert validations e from which the expert feedback for o has been excluded.

(II) The label for object o in $d_{\sim o}$ is compared with the respective expert feedback $e(o)$. If $d_{\sim o}(o) \neq e(o)$, then $e(o)$ is identified as an erroneous answer validation as in case (2).

Later, our evaluation will demonstrate that this simple check is highly effective, which makes the answer validation process robust against erroneous expert input.

6. EVALUATION

This section presents an empirical evaluation of the proposed approach using both real-world and synthetic datasets. We first discuss the experimental setup (§6.1), before turning to an evaluation of the following aspects of our solution:

- The runtime performance of the presented approach (§6.2).
- The benefits of integrating expert input as a first class citizen in the answer aggregation (§6.3).
- The performance of the proposed *i*-EM algorithm (§6.4).
- The effectiveness of the detection of faulty workers (§6.5).
- The effectiveness of hybrid expert guidance (§6.6).
- The robustness of the approach when experts provide erroneous answer validations (§6.7).
- The cost-effectiveness of expert-based answer validation (§6.8).

6.1 Experimental setup

Datasets. Our experiments have been conducted on five real-world datasets and synthetic datasets. The real-world data provides us with a realistic crowdsourcing setup by means of five classification problems that span different application domains, such as image tagging (dataset *bluebird (bb)*) or sentiment analysis (*twt* and *art* datasets). Statistics on the sizes of the real-world datasets are given in Table 4. We further employed synthetic datasets to explore parameter spaces and understand the influence of data characteristics on the performance of the algorithms. More details on both real-world and synthetic datasets are given in Appendix A.

Metrics. In addition to the uncertainty of the probabilistic answer set defined in Equation 7, we relied on the following measures:

Relative expert efforts (E_i) is the number of expert feedbacks i relative to the number of objects n in the dataset, i.e., $E = i/n$.

Precision (P_i) measures the correctness of the deterministic assignment at each validation step. Let $g : O \rightarrow L$ be the correct assignment of labels for all objects. Then, the precision of the deterministic assignment d_i at the i-th validation step is

$$P_i = \frac{|\{o \in O \mid d_i(o) = g(o)\}|}{|O|}.$$

Percentage of precision improvement (R_i) is a normalized version of precision as it measures the relative improvement. If the precision at the i-th validation step is P_i and the initial precision is P_0, then the percentage of precision improvement is

$$R_i = \frac{P_i - P_0}{1 - P_0}.$$

Experimental environment. All experimental results have been obtained on an Intel Core i7 system (3.4GHz, 12GB RAM).

6.2 Runtime performance

Since answer validation entails interactions with the expert, it should show a good runtime performance. In this experiment, we studied the effects of the number of objects on the runtime performance. The reported time is the response time of the system during one iteration of Algorithm 1, i.e., the time the expert has to wait for the selection of the next object after providing input.

Table 4: Statistics for real-world datasets

Dataset	Domain	# Objects	# Workers	# Labels
bb	Image tagging	108	39	2
rte	Semantic analysis	800	164	2
val	Sentiment analysis	100	38	2
twt	Sentiment analysis	300	58	2
art	Sentiment analysis	200	49	2

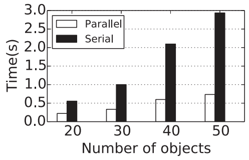

Figure 4: Response time

#questions per worker	Time (s)
10	3.1
20	4.3
40	7.5
60	9.8

Table 5: Computation time for matrix ordering

Figure 4 shows the results obtained as an average of 100 runs when using matrix partitioning (see §5.4) and the plain algorithm (*Serial*) or its parallel version (*Parallel*). Increasing the number of objects from 20 to 50, which are typically found in crowdsourcing platforms [22], increases the response time. However, even for 50 objects, the response time is less than 1 second when using parallelization, which enables immediate interactions with humans.

Further, we evaluate the start-up time required due to matrix partitioning *before* running the actual answer validation process (which does not affect the response time for the expert). We conducted an experiment with synthetic data, 16000 questions posted randomly to 1000 workers. The sparsity of the matrix is simulated by the maximal number of questions per worker which varies from 10, 20, 40, 60. Table 5 shows that the start-up time is a few seconds.

6.3 Expert validation as first-class citizen

To study the benefits of integrating expert input as a first class citizen instead of considering it as ordinary crowd answers, we compare two ways of using expert feedback. First, each expert input is a common crowd answer in the answer aggregation (*Combined*). Second, each expert input is used to validate crowd answers as proposed in our approach (*Separate*).

Figure 5 shows the results in terms of expert effort and precision improvement for the *val* dataset (results for other datasets are omitted as they exhibit similar characteristics). The *Separate* strategy outperforms the *Combined* strategy regardless of the expert efforts. This is expected—event though both approaches leverage the expert feedback, the precision of the *Combined* strategy is lower since expert answers are seen as equally important as those of the workers. Using the *Separate* strategy, expert input is deemed most important, overruling incorrect worker answers. As such, the results highlight the benefits of our method to integrate expert input as a first class citizen when aggregating crowd answers.

6.4 i-EM for answer aggregation

The *i*-EM algorithm (1) explicitly integrates answer validation from an expert as a ground truth and (2) works incrementally. Here, we evaluate the benefits of these two properties of the algorithm.

Benefits of answer validation. We evaluate the *i*-EM algorithm w.r.t the estimated assignment probability of the correct labels. For each object, a good answer aggregation would assign a higher probability to correct label than to incorrect ones. In the experiment, we keep track of the correct labels for objects and their associated

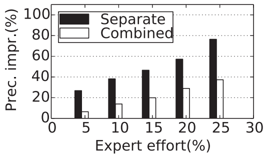

Figure 5: Ways of integrating expert input

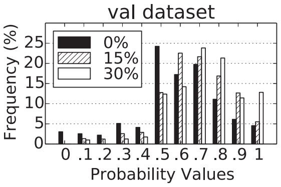

Figure 6: Benefits of answer validation

probabilities while varying the expert efforts (0%, 15%, 30%). Figure 6 presents a histogram of the probability distribution in the *val* dataset (similar results, omitted for brevity, have been obtained for the other datasets). For each object o, we measure the assignment probability $\mathcal{U}(o,l)$ of its correct label l assigned by i-EM . If the assignment probability of the label for object o is in a probability bin, the count for that bin is increased.

We note that the number of correct labels which have a probability less than 0.5 is overall small. Still, around 3% of correct labels have a probability less than 0.1 when no expert input has been integrated (expert effort 0%), meaning that answer aggregation without validation (traditional EM) may assign a very low probability for some of the correct labels. Increasing the amount of expert input, the probability range covering most of the correct labels shifts from the 0.5 bin to higher probability bins. Hence, answer aggregation with more expert input (our i-EM algorithm) is able to assess the assignment probabilities of the correct labels better than without expert input (traditional EM algorithm).

Benefits of incrementality. Most EM-like algorithms are sensitive to initialization [22]. We study robustness of our i-EM algorithm by comparing two strategies: (i) *incremental* – upon new expert input, an iteration is initialized using the previous state of the probabilistic answer set (the i-EM algorithm), and (ii) *non-incremental* – each iteration is initialized with random values (traditional EM).

As different initializations lead to different estimations of probabilities, the selection of objects for validation of our guiding strategy may be affected. Therefore, we first analyze whether the two strategies suggest the same object. We compare the two strategies at different levels of expert efforts by running them to get the assignment probabilities and the confusion matrices, calculate the information gain for each object, and assess whether the same object has the highest information gain for both strategies. Figure 7 shows the results relative to the expert effort as the percentage of cases (averaged over 100 runs) where the two strategies select the same object. Indeed, both strategies select the same objects in virtually all cases, indicating initialization robustness of the i-EM algorithm.

Further, the idea of the i-EM algorithm is that by incrementalizing answer aggregation without affecting the validation guidance, we reduce the convergence ratio of expectation maximization. We verify this hypothesis by comparing the runtime performance of the two strategies in terms of convergence. For a synthetic dataset with 50 objects and 20 workers, for which the worker population is simulated as outlined in Appendix A with a reliability of 0.65 for normal workers, the obtained results are shown in Figure 8 (average over 100 runs). It depicts the percentage of expert input relative to the percentage of iterations saved by using the *incremental* strategy.

We note that the i-EM algorithm converges faster and saves more than 30% of the iterations, with the iteration reduction becoming larger with an increased number of answer validations. This is explained by the fact that the more expert input is available, the more the estimation of label correctness and worker reliability converges to the exact values. Since the i-EM algorithm estimates label correctness and worker reliability based on the values of the previous iteration, it requires less iterations over time.

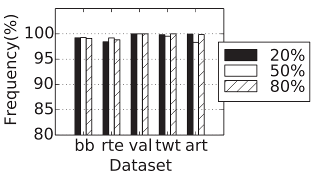

Figure 7: i-EM on guiding

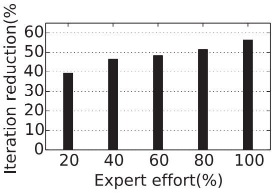

Figure 8: Efficiency of i-EM

6.5 Effectiveness of the spammer detection

Since our guiding technique includes the detection of faulty workers (e.g. spammers, sloppy workers), it is necessary to analyze the technique with different detection thresholds. Since real datasets do not have information about who is spammer, we resort to using synthetic data with 20 workers that assign one of two labels to 50 objects. We then vary the threshold τ_s to detect uniform and random spammers from 0.1 to 0.3 while keeping the threshold τ_p for sloppy workers at 0.8. We also vary the validation effort from 20% to 100%. We measured the precision (ratio of correctly identified spammers over all identified spammers) and recall (ratio of correctly identified spammers over all spammers) of the detection.

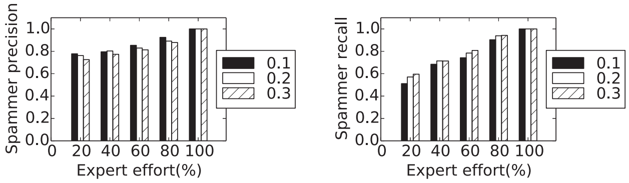

Figure 9: Efficiency of the spammer detection technique

Figure 9 (average of 100 runs) illustrates that, as the number of validations increases, both precision and recall of spammer detection increase. The confusion matrices used to detect spammers are built based on the answer validations. Hence, with more expert input, the confusion matrices better reflect the reliability of the workers. Also, we observe the trade-off between precision and recall as we increase the spammer score threshold. An increased threshold yields lower precision, but higher recall. Striving at a balance, we set the detection threshold to 0.2 in the remaining experiments.

6.6 Effectiveness of expert guidance

Next, we evaluate the effectiveness of our guiding approach for reducing expert efforts on real-world datasets. To this end, we first verified the underlying assumption of our techniques to expert guidance, i.e., that the uncertainty of a probabilistic answer set, quantified as introduced in §4.2, is correlated to the actual precision of the deterministic assignment. Our results, presented in detail in Appendix B, show that this is indeed the case and that there is a strong correlation between the measures (supported by Pearson's correlation coefficient value of -0.9461). Hence, the measured uncertainty is a truthful indicator of the result correctness.

Turning to the guidance strategies, we mimic the validating expert by using the ground-truth provided in the datasets until precision reaches 1.0. We compare the proposed approach (*hybrid*) with a method that implements the function *select* in the validation process by selecting the most 'problematic' object (*baseline*). Intuitively, we measure how 'problematic' an object is by the entropy of its probability (see Appendix C for a formal definition). This baseline method is better than random selection since it strives for the objects that are on the edge of being considered right or wrong, which are the major sources of uncertainty in the answer set.

Figure 10 shows the results for the first three real-world datasets (*bb*, *rte*, and *val*), the remaining datasets (*twt* and *art*) are discussed

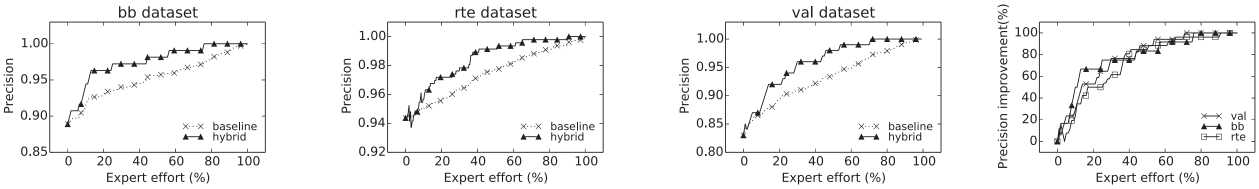

Figure 10: Effectiveness of guiding

Appendix C. The approach developed in this paper (*hybrid*) clearly outperforms the *baseline* method. For example, in the *bb* dataset, our approach leads to a precision above 0.95 with expert input on only 10% of the objects. The baseline method requires expert validation of around 50% to reach the same level of precision.

The relative improvement of precision for different expert effort levels is illustrated in the last plot in Figure 10. For instance, for 20% expert efforts, we achieve an improvement of precision of at least 50% for all datasets. Also, precision improvement is larger for smaller amounts of expert efforts, which emphasizes the effectiveness of our guidance strategy in particular for scenarios with a limited effort budget for the validation.

We further explored the effectiveness of our approach in relation to different aspects of a crowdsourcing setup using synthetic data. While the detailed results of these experiments are available in Appendix C, we summarize the main findings as follows. The presented approach outperforms the baseline method in terms of effectiveness (precision vs. expert effort) independent of (1) the number of possible labels, (2) the size of the crowd, (3) the worker reliability, (4) the difficulty of the questions, and (5) the presence of spammers. This indicates that the improvements obtained with the presented approach are not specific to particular crowdsourcing setups, but generalize to a wide range of applications.

6.7 Robustness against expert mistakes

In §5.5, we discussed two types of erroneous answer validations, the expert wrongly deviates from the aggregation of crowd answers or wrongly confirms it, along with a simple confirmation check to detect mistakes of the second type.

Types of erroneous answer validations. To analyze which of the two erroneous validations is more likely to occur, we developed a validation tool [1] that enables an expert to give feedback on crowd answers and shows the aggregated crowd answer for an object during validation. We conducted an experiment with five experts that used the tool for the two datasets *twt* and *art*, which comprised the actual questions posted to the crowd. Their input was verified against the ground truth.

In general, the number of erroneous answer validations is small. For the *twt* dataset, all experts provide correct input. For the *art* dataset, 8% of the expert input is erroneous. For these cases, we find that, throughout, the respective answer from the crowd workers is also incorrect. This indicates that indeed, the wrong confirmation of an aggregated answer is the more likely type of mistake.

Detecting erroneous answer validations. Next, we evaluate the effectiveness of the confirmation check to detect erroneous answer validations by simulating expert mistakes. For a given probability p, we change the expert input from a correct validation to an incorrect validation. The experiment is conducted on all real-world datasets with the *hybrid* selection strategy and when triggering the confirmation check after each 1% number of total validations.

Table 6 shows the percentage of detected mistakes when increasing the probability of an expert mistake. Across all datasets, the vast majority of artificially inserted mistakes is detected. For example, even with a relatively high probability for erroneous answer validations (p=15%), all mistakes in expert input are detected.

Expert guidance and erroneous answer validations. Finally, we study the relation between expert effort and precision in the presence of expert mistakes. The confirmation check is run after each 1% number of total validations. Upon each detected mistake, we allow the expert to reconsider the respective input; i.e increment the expert effort by 1. The experiment is conducted using the real-world dataset *art*, for which the experts indeed made mistakes. To aim for the worst-case scenario, we use the validations from the expert with the most mistakes.

As illustrated in Figure 11, the precision obtained with the *hybrid* strategy is still much better than the baseline method. Moreover, the actual obtained precision values are close to those obtained without erroneous answer validations (see Figure 16 in Appendix C). This result indicates that our approach is robust against potential mistakes in expert input.

6.8 Cost trade-off: Experts vs crowd workers

In the previous experiments, we have evaluated different aspects of our guiding approach for reducing expert efforts. In this final set of experiments, we aim to show that our approach is able to achieve high precision within a reasonable cost for different crowdsourcing setups. Technically, we compare two strategies: (i) *EV*, our approach that uses an expert to validate crowd answers, and (ii) *WO*, we use only the crowd and add more crowd answers with the assumption that this will increase the correctness of the answer set (i.e. aggregated results are computed by using the traditional EM).

Cost model. Our cost model for this experiment covers monetary cost and completion time.

Monetary cost: We assume that the cost of an expert input is θ-times more expensive than an answer by a worker. To estimate θ, we first consider the answer cost of a crowd worker via the average wage on AMT, which is just under 2.00$/h [40]. For the cost of an answer by an expert, we consider salary standards of traditional workers and select the most expensive case, i.e., 25$/h, the average wage in Luxembourg [46]. Then, the ratio θ between the cost per answer of an expert and a worker is about $(25\$/h)/(2\$/h) = 12.5$.

Completion Time: Crowdsourcing in practice is often subject to a time constraint (e.g., 1 hour for simple tasks on AMT). In our setting, the completion time involves (1) *crowd time*, time for the crowd workers to answer and (2) *expert time*, time for the experts to provide input for all the questions that can be covered by the budget. Crowd time is often considered to be constant, since workers work concurrently [50]. Hence, the completion time is primarily determined by the expert time, which is derived from the number of expert inputs (assuming constant validation time for all questions).

Below, we consider a setting where m workers have been hired to answer n questions. With ϕ_0 as the average cost of asking crowd workers per object, the initial cost for deriving the answers is $n \times \phi_0$. To improve the quality of the answer set, two strategies may be followed. First, an expert can be asked to validate i answers (the EV approach), which incurs an additional cost of $\theta \times\ i$ or in

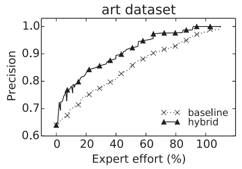

Figure 11: Guiding with expert mistakes

Dataset	p : probability of mistake			
	0.15	0.20	0.25	0.30
bb	100	100	94	82
rte	100	100	96	88
val	100	94	89	79
twt	100	100	92	87
art	100	92	88	81

Table 6: Percentage of detecting mistakes in expert validation

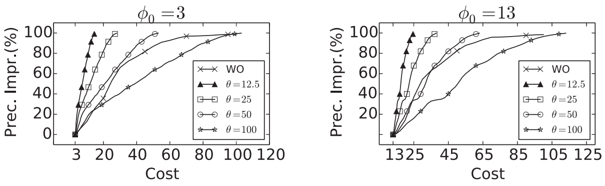

Figure 12: Collect more crowd answers vs. validate more

total $P_{EV} = \theta \times i + n \times \phi_0$. Second, the workers can be asked to answer more questions, which increases the average cost per object to $\phi > \phi_0$. Then, the total cost of the WO approach is $P_{WO} = n \times \phi$.

Trade-off with undefined budget. In general, there is a trade-off between the cost incurred by a crowdsourcing task and the result correctness. Higher cost, spent on answer validation by an expert or additional crowd answers, yields higher correctness of the aggregated answers. We analyze this trade-off to determine under which conditions hiring only additional crowd workers (*WO* approach) is less beneficial than hiring a validating expert (*EV* approach). Figure 12 illustrates the relation between the invested cost, normalized over the number of objects ($P_{WO}/n = \phi$ and $P_{EV}/n = \phi_0 + \theta \times i/n$), and the obtained improvement in precision for different expert-crowd cost ratios $\theta = 12.5, 25, 50, 100$ and initial costs $\phi_0 = 3, 13$.

The *EV* approach yields higher precision improvements for the same costs compared to the *WO* approach with different values of ϕ_0 and for $\theta = 12.5, 25, 50$. With $\phi_0 = 3$ and $\theta = 25$, for instance, to improve the precision by 80%, the EV approach requires a cost of 20 per object, while the cost of the WO approach is 40. Also, the WO approach does not achieve 100% precision even under high costs, due to faulty workers. Having more answers from these types of workers only increases the uncertainty in the answer set.

In sum, if high precision is desired, the EV approach yields better overall results. For instance, for a realistic setup with $\phi_0 = 13$ and $\theta = 12.5$, to achieve 100% precision improvement, our approach has a cost per object of 15. The WO approach, in turn, has a cost of 100, but is still not able to achieve 100% precision improvement. When expert input is very expensive ($\theta = 100$), increasing only the number of crowd workers yields better results. However, we consider an expert-crowd cost ratio of 100 to be unlikely in practice.

Trade-off with budget constraint. The results above indicate that, without budget constraints, the *EV* approach achieves higher precision with lower cost compared to the *WO* approach regardless of ϕ_0. However, using a fixed cost, the precision obtained with the *EV* approach depends on the value of the initial cost ϕ_0. We therefore analyze how to achieve the highest precision under a fixed budget b using the EV approach. That requires deciding how much of an overall budget should be spent on retrieving crowd answers. Finding an optimal value of ϕ_0 thereby determines the best budget allocation between the expert and the crowd workers. In practice, the budget b is bounded by the cost of using only an expert, i.e., $n \leq b \leq \theta \times n$. To parameterize the budget spent on expert feedback, we formulate it as $b = \rho \times \theta \times n$, where $\rho \in [1/\theta, 1]$.

Figure 13 illustrates the result correctness in terms of precision for different allocations of the budget to crowd workers ($\phi_0/(\rho \times \theta)$), when varying the ratio ρ and setting $\theta = 25$. As a reference, the figure also includes the result for the WO approach (crowd cost is 100%), which is a special case of the EV approach where all the budget is spent on crowd workers, i.e., $\theta \times\ i = 0$ and $\phi_0 = b/n$.

We observe that for each ratio ρ, there is an allocation point (ϕ_0) that maximizes precision. For instance, for $\rho = 0.4$, maximal precision is obtained with 75% of the budget being spent on crowd workers and 25% of the budget used for validation by an expert. Based on this analysis, we can therefore select the optimal allocation for a specific setup. Further, except for the case with little budget ($\rho = 0.3$), a distribution of the budget between the crowd workers and the validating expert leads to maximal precision, which highlights the benefits of integrating answer validation in a crowdsourcing setup.

Trade-off with budget and time constraints. Next, we consider a setup where the best budget allocation should be determined under both, budget and time constraints. Figure 14 extends the plot of the relation between the result precision and the budget allocation with the completion time captured by the amount of expert input (y2-axis). In this figure, point B denotes the intersection between the lines representing the time constraint (green dashed line) and the completion time (orange solid line). Based on point B, a region in which the time constraint is satisfied is identified, which, in Figure 14 is bounded by the range $[C, 100]$ in terms of the allocation of the budget to crowd workers. For this region, the maximum precision is denoted by point A. As a result, we have determined the budget allocation (x-value at point A) that yields the highest precision when satisfying both the time and budget constraint.

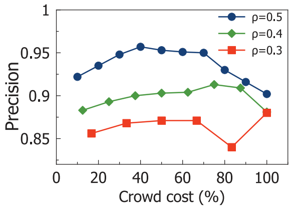

Figure 13: Allocation of fixed budget

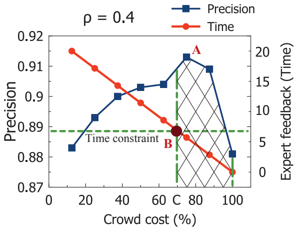

Figure 14: Balance with time and budget constraints

Finally, we analyzed the effects of faulty workers, worker reliability, and question difficulty, on the cost model and the handling of the trade-off. The details of these experiments are provided in Appendix D. We found that the presented approach of using an expert to validate crowd answers, in most cases, outperforms an approach that relies solely on crowd workers. Exceptions to this trend are observed only in borderline cases, e.g., if the budget is extremely small (meaning that only a small number of crowd workers can be hired in the first place) and experts are much more costly than crowd workers i.e $\theta \geq 100$ (which is very unlikely in practice as this means the tasks are overpaid or too difficult even for crowdsourcing). Hence, we conclude that the integration of an expert allows for more efficient crowdsourcing for a wide range of applications.

7. RELATED WORK

Crowdsourcing. Having discussed applications and types of crowdsourcing tasks already in §2, we turn to a discussion of the cost in crowdsourcing, including monetary costs and completion time. Several studies focused on minimizing cost when posting tasks [8, 50]. In this paper, we leverage existing works on task-posting mechanism as a black-box; i.e. all of the worker answers are collected in advance before being considered by our approach. The focus of our work is the guidance for minimizing a different aspect of crowdsourcing, i.e., the cost of validating crowd answers. As a side-effect, given a limited budget constraint, our approach can predict the optimal strategy of distributing the cost for the validation and for the crowd to achieve the highest output quality (see §6.8).

Regarding quality control in crowdsourcing, there is a plethora of automatic approaches that target an assessment of the worker quality, including expertise characterization [29] and spammer detection [32]. Complemented with a worker assessment mechanisms, answer aggregation tries to find the hidden ground truth from the answer set given by crowd workers. Answer aggregation methods can be classified into two categories: *non-iterative* and *iterative* approaches. Non-iterative approaches [32] use heuristic methods to compute a single aggregated value of each object separately. Iterative approaches [23] perform a series of convergent iterations by considering the answer set as a whole. Despite the above efforts, the results of automatic quality control are inherently uncertain, since they are heuristic-based and there is no technique that performs well in the general case [22]. To address this dilemma, the semi-automatic solution presented in this paper is to employ an expert to validate crowd answers.

Our approach aims at *guiding* an expert when *validating* input from crowd workers, which is different to other approaches for crowdsourcing that include experts, such as [20, 26, 27]. In particular, Karger et al. [27] rely on experts that know the reliability of crowd workers, a premise that is not realistic in the general setting for crowdsourcing explored in this work, to prove the optimality of their approach. Other work focuses on a related, but fundamentally different problem. The techniques presented in [20, 26] target the identification of correct labels for *new* objects based on the labels for known objects, whereas we aim at validation, i.e., finding the correct labels for known objects.

Truth Finding. Given a set of data items claimed by multiple sources, the truth finding (a.k.a. truth discovery) problem is to determine the true values of each claimed item, with various usages in information corroboration [15], and data fusion [13]. Similar to our crowdsourcing setting, existing work on truth finding also models the mutual reinforcing relationship between sources and data items, e.g., by a Bayesian model [52], maximum likelihood estimation [47], and latent credibility analysis [36]. In contrast to our setting, these techniques incorporate prior knowledge about various aspects of the source and the data, such as the dependence between sources [11] and the temporal dimension in evolving data [12]. As such, these techniques cannot be directly applied to our solution (workers perform the tasks individually, objects do not evolve over time). To the best of our knowledge, there is no work on employing answer validation by experts to check the results of automatic techniques. Therefore, our work on guiding validation effort can be tailored to the truth finding settings as well.

Recommendation systems. Close to our work is research on recommendation systems. Here, the core problem is, given an existing set of user ratings for particular items, to recommend one of these items that best fit a particular user in terms of information content [16]. This problem is similar to ours in the sense that we also select the objects with best information content (i.e., that yield the maximal uncertainty reduction) for answer validation. However, the underlying models of the two settings are completely different. In recommendation systems, the information of an item is measured by the notion of similarity: similar users would have similar preferences on similar items and vice-versa [41]. Whereas, this similarity assumption does not exist for workers and objects in crowdsourcing. Moreover, there is a also large body of work on recommendation systems studying malicious users [31], who provide untruthful ratings or reviews to manipulate the recommendation output. Although many detection techniques have been proposed, they cannot be applied in our context since they depend on the application domains and contextual features [35]. Most importantly, there is no method making use of validation input for identifying malicious users. As a result, our work on using a validating expert to handle spammers in crowdsourcing can be tailored for recommendation systems.

Guiding user feedback. Guiding user or expert feedback has been studied in different contexts. In the field of data integration, Jeffery et al. [24] proposed a decision theoretic framework to rank candidate matches for answer validation in order to improve the quality of a dataspace. Focusing on matching of data schemas in a network setting, Nguyen et al. [34] presented a reconciliation algorithm that leverages expert input. Yakout et al. [49], in turn, proposed an active-learning based process that requests expert input to help training classifiers in order to detect and repair erroneous data. Similar to these works, we rely on models from the fields of Decision Theory and Active Learning [42]. Despite the similarities in the applied models, there are two main differences between the aforementioned approaches to user guidance and the method presented here. First, in the above domains (data integration, schema matching), input stems from automatic tools, which renders it deterministic and traceable. In contrast, our methods have to cope with human input, which is unreliable, potentially non-deterministic or even malicious. Second, existing guidance methods aim at a different goal, which means that measures for the benefit of expert input are highly domain dependent (e.g., the approach in [24] is purely driven by the size of query results and independent of the source of user input). Our method, in turn, is tailored to the specific characteristics of crowdsourcing scenarios.

8. CONCLUSIONS AND FUTURE WORK

This paper proposed techniques to support an expert in validating crowd answers obtained for a crowdsourcing task. Based on the requirements identified for such techniques, we presented an answer validation process that features two steps: *answer aggregation* and *expert guidance*. The former relates to the creation of a probabilistic model that assesses the reliability of workers and the correctness of the crowd answers. The latter features different strategies for guiding an expert in the validation: worker-driven, uncertainty-driven, and a hybrid approach. The worker-driven method aims at detecting and removing faulty workers from the community, whereas the uncertainty-driven strives for a maximal improvement of the answer correctness under the assumption of truthful workers. Since both goals help to improve the overall result, our hybrid approach combines both methods with a dynamic weighting scheme.

Our evaluation showed that our techniques outperform respective baselines methods significantly and save up to 50% of expert efforts. Also, in most cases, close to perfect result correctness is reached with expert input for only 20% of the considered objects.

In future work, we aim to explore dynamic selection of the answer aggregation method. Since these methods have different strengths and shortcomings [22], expert input may be used to dynamically choose the most appropriate aggregation strategy. We also intend to lift our approach to other types of crowdsourcing tasks such as *continuous*, *partial-function*, and *similarity* tasks. For example, *continuous* tasks map to *discrete* tasks by changing probability mass functions to probability density functions and sums to integrals.

Acknowledgment. The research has received funding from the EU-FP7 EINS project (grant 288021) and the ScienceWise project.

9. REFERENCES

[1] https://code.google.com/p/crowdvalidator/.

[2] http://www.crowdflower.com/.

[3] http://www.mturk.com/.

[4] Y. Amsterdamer, Y. Grossman, T. Milo, and P. Senellart. Crowd mining. In *SIGMOD*, pages 241–252, 2013.

[5] A. Arasu, M. Götz, and R. Kaushik. On active learning of record matching packages. In *SIGMOD*, pages 783–794, 2010.

[6] A. Artikis, M. Weidlich, F. Schnitzler, I. Boutsis, T. Liebig, N. Piatkowski, C. Bockermann, K. Morik, V. Kalogeraki, J. Marecek, et al. Heterogeneous stream processing and crowdsourcing for urban traffic management. In *EDBT*, pages 712–723, 2014.

[7] J. A. Blakeley, P.-A. Larson, and F. W. Tompa. Efficiently updating materialized views. In *SIGMOD*, pages 61–71, 1986.

[8] C. C. Cao, J. She, Y. Tong, and L. Chen. Whom to ask?: jury selection for decision making tasks on micro-blog services. In *VLDB*, pages 1495–1506, 2012.

[9] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *J. R. Stat. Soc.*, pages 20–28, 1979.

[10] 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.

[11] X. L. Dong, L. Berti-Equille, Y. Hu, and D. Srivastava. Solomon: Seeking the truth via copying detection. In *VLDB*, pages 1617–1620, 2010.

[12] X. L. Dong, L. Berti-Equille, and D. Srivastava. Truth discovery and copying detection in a dynamic world. In *VLDB*, pages 562–573, 2009.

[13] X. L. Dong and F. Naumann. Data fusion: resolving data conflicts for integration. In *VLDB*, pages 1654–1655, 2009.

[14] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, pages 211–218, 1936.

[15] A. Galland, S. Abiteboul, A. Marian, and P. Senellart. Corroborating information from disagreeing views. In *WSDM*, pages 131–140, 2010.

[16] F. Garcin, B. Faltings, R. Jurca, and N. Joswig. Rating aggregation in collaborative filtering systems. In *RecSys*, pages 349–352, 2009.

[17] C. Gokhale, S. Das, A. Doan, J. F. Naughton, N. Rampalli, J. Shavlik, and X. Zhu. Corleone: Hands-off crowdsourcing for entity matching. In *SIGMOD*, pages 601–612, 2014.

[18] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman, 1989.

[19] R. G. Gomes, P. Welinder, A. Krause, and P. Perona. Crowdclustering. In *NIPS*, pages 558–566, 2011.

[20] Q. Hu, Q. He, H. Huang, K. Chiew, and Z. Liu. Learning from crowds under experts supervision. In *PAKDD*, pages 200–211. 2014.

[21] Z. Huang, A. Olteanu, and K. Aberer. Credibleweb: a platform for web credibility evaluation. In *CHI*, pages 1887–1892, 2013.

[22] N. Q. V. Hung, N. T. Tam, L. N. Tran, and K. Aberer. An evaluation of aggregation techniques in crowdsourcing. In *WISE*, pages 1–15, 2013.

[23] P. G. Ipeirotis, F. Provost, and J. Wang. Quality management on amazon mechanical turk. HCOMP, pages 64–67, 2010.

[24] S. R. Jeffery, M. J. Franklin, and A. Y. Halevy. Pay-as-you-go user feedback for dataspace systems. In *SIGMOD*, pages 847–860, 2008.

[25] H. J. Jung and M. Lease. Improving quality of crowdsourced labels via probabilistic matrix factorization. In *HCOMP*, pages 101–106, 2012.

[26] H. Kajino, Y. Tsuboi, I. Sato, and H. Kashima. Learning from crowds and experts. In *HCOMP*, pages 107–113, 2012.

[27] D. R. Karger, S. Oh, and D. Shah. Iterative learning for reliable crowdsourcing systems. In *NIPS*, pages 1953–1961, 2011.

[28] G. Karypis and V. Kumar. Metis-unstructured graph partitioning and sparse matrix ordering system, version 2.0. 1995.

[29] G. Kazai, J. Kamps, and N. Milic-Frayling. Worker types and personality traits in crowdsourcing relevance labels. In *CIKM*, pages 1941–1944, 2011.

[30] C.-W. Ko, J. Lee, and M. Queyranne. An exact algorithm for maximum entropy sampling. *Operations Research*, pages 684–691, 1995.

[31] S. K. Lam and J. Riedl. Shilling recommender systems for fun and profit. In *WWW*, pages 393–402, 2004.

[32] K. Lee, J. Caverlee, and S. Webb. The social honeypot project: protecting online communities from spammers. In *WWW*, pages 1139–1140, 2010.

[33] X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang. Cdas: a crowdsourcing data analytics system. In *VLDB*, pages 1040–1051, 2012.

[34] Q. V. H. Nguyen, T. T. Nguyen, Z. Miklós, K. Aberer, A. Gal, and M. Weidlich. Pay-as-you-go reconciliation in schema matching networks. In *ICDE*, pages 220–231, 2014.

[35] M. O'Mahony, N. Hurley, N. Kushmerick, and G. Silvestre. Collaborative recommendation: A robustness analysis. *TOIT*, pages 344–377, 2004.

[36] J. Pasternack and D. Roth. Latent credibility analysis. In *WWW*, pages 1009–1020, 2013.

[37] A. J. Quinn and B. B. Bederson. Human computation: a survey and taxonomy of a growing field. In *CHI*, pages 1403–1412, 2011.

[38] V. C. Raykar and S. Yu. Ranking annotators for crowdsourced labeling tasks. In *NIPS*, pages 1809–1817, 2011.

[39] J. Reason. *Human error*. Cambridge university press, 1990.

[40] J. Ross, L. Irani, M. Silberman, A. Zaldivar, and B. Tomlinson. Who are the crowdworkers?: shifting demographics in mechanical turk. In *CHI*, pages 2863–2872, 2010.

[41] N. Rubens, D. Kaplan, and M. Sugiyama. Active learning in recommender systems. In *Recommender Systems Handbook*, pages 735–767. Springer, 2011.

[42] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003.

[43] C. E. Shannon. A mathematical theory of communication. *SIGMOBILE*, pages 3–55, 2001.

[44] C. Sun, N. Rampalli, F. Yang, and A. Doan. Chimera: Large-scale classification using machine learning, rules, and crowdsourcing. In *VLDB*, pages 1529–1540, 2014.

[45] J. Surowiecki. The wisdom of crowds: Why the many are smarter than the few and how collective wisdom shapes business, economies. *Economies, Societies and Nations*, 2004.

[46] TRAVAIL. Global wage report 2012-13. International Labour Organization (ILO), 2012.

[47] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher. On truth discovery in social sensing: A maximum likelihood estimation approach. In *IPSN*, pages 233–244, 2012.

[48] P. Welinder and P. Perona. Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *CVPRW*, pages 25–32, 2010.

[49] M. Yakout, A. K. Elmagarmid, J. Neville, M. Ouzzani, and I. F. Ilyas. Guided data repair. In *VLDB*, pages 279–289, 2011.

[50] T. Yan, V. Kumar, and D. Ganesan. Crowdsearch: Exploiting crowds for accurate real-time image search on mobile phones. In *MobiSys*, pages 77–90, 2010.

[51] C. J. Zhang, L. Chen, H. V. Jagadish, and C. C. Cao. Reducing uncertainty of schema matching via crowdsourcing. In *VLDB*, pages 757–768, 2013.

[52] B. Zhao, B. I. Rubinstein, J. Gemmell, and J. Han. A bayesian approach to discovering truth from conflicting sources for data integration. In *VLDB*, pages 550–561, 2012.

APPENDIX

A. FURTHER DETAILS ON DATASETS

Real-world data. We have used real-world datasets from different domains, namely *bluebird (bb)*, *rte*, *valence (val)*, *tweet (twt)*, and *article (art)*. In the *bb* dataset, workers have to identify one of two types of birds in an image. The crowdsourcing tasks of the *rte* dataset comprise two sentences and workers need to assert whether one sentence can be inferred from the other one. In the *val* dataset, workers are asked to annotate whether a headline expresses positive or negative meaning. The tasks for the *twt* and *art* datasets are both about sentiment analysis in which the workers need to evaluate the sentiment of a tweet or a scientific article. However, the questions of the *art* dataset are more difficult than the questions of the *twt* dataset as the sentiment of scientific articles are harder to analyze. All datasets are publicly available including the ground truth which is provided by the dataset owners.

Synthetic data:. We used several generated datasets. Since this data should exhibit similar characteristics as real-world data, we considered several parameters for the data generation, in particular: (i) n – the number of objects, (ii) k – the number of workers, (iii) m – the number of labels, (iv) r – the reliability of normal workers, reflecting the probability of their answers being correct and (v) σ – the percentage of spammers in the worker population. For the synthetic dataset, we also simulated the ground truth (the correct labels) for the objects. However, it is not known by the simulated workers and only used to simulate the answer validations.

An important part of our synthetic data is the crowd simulation. We follow a practical guideline [22] to simulate the different worker characteristics of the crowd. Specially, we distribute the worker population into α% reliable workers, β% sloppy workers and γ% spammers. According to a study on crowd population at real-world crowdsourcing services [29], we assign the default values of these parameters as follows: $\alpha = 43$, $\beta = 32$ and $\gamma = 25$. In the experiments, the distribution of the worker types is the same as discussed unless stated otherwise.

B. RELATION BETWEEN UNCERTAINTY AND PRECISION

In this experiment, we study the relation between the uncertainty of the probabilistic answer set and the precision of the deterministic assignment. To this end, we perform the uncertainty-driven expert guidance on a synthetic dataset, in which we vary the number of workers from 20 to 40, the percentage of spammers from 15% to 35%, and the reliability of the workers from 0.65 to 0.75. For each combination setting of the parameters, we guide the answer validation until precision reaches 1.0 and report the uncertainty of answer aggregation along the way.

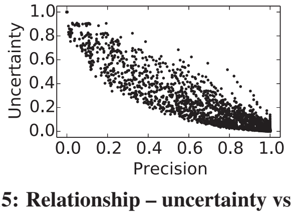

Figure 15: Relationship – uncertainty vs. precision

Figure 15 depicts the results in terms of the relation between precision and normalized uncertainty (i.e., dividing the uncertainty values by the maximum uncertainty obtained in the run). We observe a strong correlation between both measures, which is further supported by the Pearson's correlation coefficient of -0.9461. Hence, as the uncertainty decreases, the precision of the deterministic assignment increases. Also, there is a concentration of data points for precision values in $[0.8, 1.0]$ and normalized uncertainty values in $[0, 0.2]$. Hence, our approach helps to achieve near-perfect precision even when the uncertainty is not reduced to 0.

C. EVALUATIONS OF EXPERT GUIDANCE (CONT'D)

In the following experiments, we analyze the effects of the guiding strategy with different crowdsourcing setup, including the number of labels, the number of workers, worker reliability, question difficulty, and the presence of spammers. Since these experiments (except the experiment on question difficulty) require changing the workers' characteristics (which is not known for the real-world datasets), they are conducted using synthetic data.

We compare the results obtained with our guiding approach (*hybrid*) to a baseline guiding method that selects the object with the highest uncertainty to seek feedback (*baseline*):

$$select(O) = \arg\max_{o \in O} H(o)$$

Our *hybrid* approach is different from the baseline as it further considers the consequences of validation in addition to the mutually reinforcing relations between the reliability of workers and assignment correctness.

Effects of the number of labels. Many crowdsourcing applications are designed with multiple-choice questions, i.e., applications show different numbers of possible assignment labels. To evaluate the effect of the number of labels on the performance of our approach, we rely on a synthetic dataset containing 50 objects and 20 workers. The worker reliability is set to 0.65 for non-spammers and the spammers are simulated to follow the distribution discussed in Appendix A. We report the precision while varying the expert efforts (until precision reaches 1.0) and the number of labels ($m = 2$ and $m = 4$).

Figure 17 shows that our approach consistently outperforms the baseline method. Interestingly, the difference between the two approaches is very large in the setup with 4 assignment labels. For example, our approach (*hybrid*) increases precision to 1.0 using only 40% of expert input. However, the precision of 0.99 is already achieved using 10% of expert feedbacks. We can explain this observation as follows: with a higher number of labels, crowd workers are less likely to choose the correct answer by chance. Hence, our approach is able to detect reliable workers faster, which leads to a better overall performance. To test for the most challenging case, we fix the number of labels to 2 in the remaining experiments.

Effect of the number of workers. The idea behind crowdsourcing is that individual crowd answers complement each other. Thus, the aggregation of answers should be closer to the truth as more workers participate [45]. To verify this hypothesis, we vary the number of workers k from 20 to 40 that assign one of two labels to 50 objects. Figure 18 illustrates an important finding that our approach leads to better results for any number of workers. Taking a fixed amount of expert input, precision increases if more workers are employed. The reason is the widely quoted 'wisdom of the crowd' [45], which eventually leads to better precision. Another finding is that the precision improvement with the same amount of

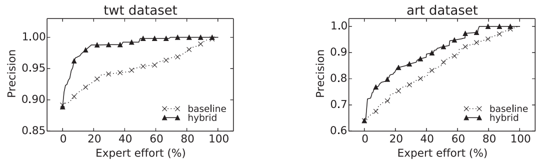

Figure 16: Effects of question difficulty

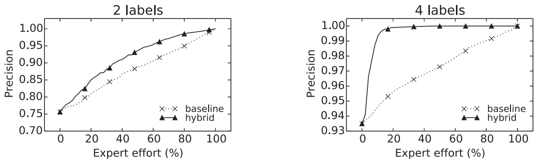

Figure 17: Effect of number of labels

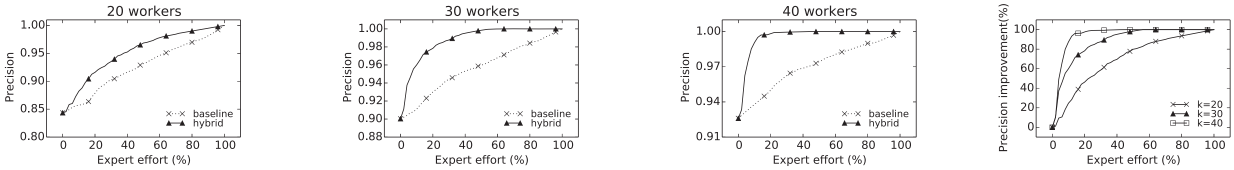

Figure 18: Effect of number of workers

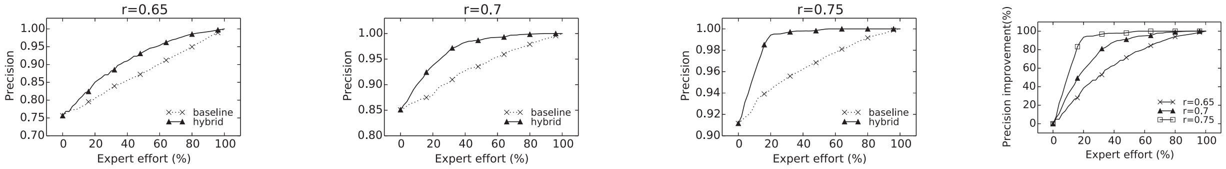

Figure 19: Effect of worker reliability

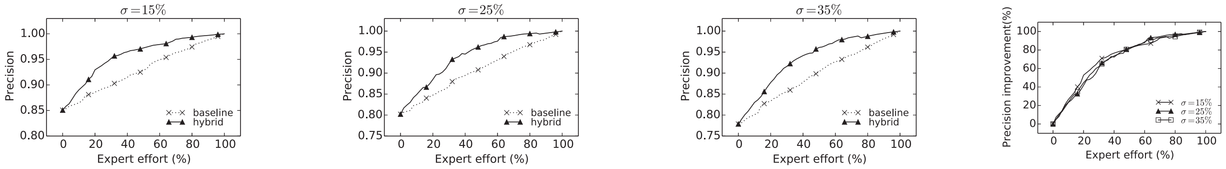

Figure 20: Effect of spammers

expert input is higher if we have more workers (most right plot in Figure 18). This is expected since, by having more workers, we acquire more answers for the same question, which results in better estimates of assignment probabilities and worker reliabilities. Our approach, thus, has a higher chance to select the objects that lead to a large gain in correctness.

In sum, the two findings suggest that increasing the number of workers is beneficial not only for computing assignment probabilities, but also for guiding answer validation. For the remaining experiments, we fix the number of workers to be the smallest tested value ($k = 20$), which is the most challenging scenario.

Effect of worker reliability. We further explored the effects of the worker reliability r on the effectiveness of our approach. As above, we used a dataset of 20 workers assigning one out of two labels to 50 objects. We then varied the reliability of the non-spammer workers from 0.65 to 0.75.

Figure 19 illustrates a significant improvement in precision using our approach (*hybrid*) compared to the *baseline* method. For instance, if the average worker reliability is 0.7, to achieve a precision of 0.93, our approach requires expert input for 20% of the objects, whereas the baseline method requires input for 40% of the objects. In other words, the amount of efforts the baseline method requires is twice that of our approach. Also, with the same amount of feedback, precision is increased if the average reliability of the workers is higher (most right plot in Figure 19). This is because an answer set provided by reliable workers requires less validation than an answer set coming from unreliable workers.

Effect of spammers. In this experiment, we studied the robustness of our guiding approach to spammers using the same dataset as in the previous experiment (20 workers, two labels, 50 objects). We varied the percentage of spammers σ in the worker population from 15% to 35% to analyze the effect of these spammers.

Independent of the percentage of spammers, our approach (*hybrid*) outperforms the *baseline* method, see Figure 20. The largest difference between the two approaches is observed when the percentage of spammers is 15%. In that case, to achieve a precision of 0.95, our approach needs 35% of expert input, while the baseline method requires 70%. Regarding the precision improvement (right most plot in Figure 20), the results are relatively similar across different percentages of spammers. For instance, using 40% of expert input, we are able to increase the precision of the deterministic assignment by 70%, independent of the percentage of spammers. Hence, our approach is indeed robust to the presence of spammers.

Effects of question difficulty. Beside worker reliability, another factor that can affect the performance of our method is the question difficulty. For hard questions, even reliable workers may give incorrect answers. As a result, there is a need to analyze the effects of question difficulty on the performance of our approach. We compared our approach with the baseline approach using two datasets: *twt* and *art*, where the questions in the *art* dataset is harder than the other. The experimental results are shown in Figure 16, where the x-axis depicts the expert efforts while the y-axis illustrates the precision of the deterministic assignment.

We observe that our approach is able to outperform the baseline approach for both datasets, meaning that the approach is robust against question difficulty. For instance, for the *twt* dataset with easy question, our approach needs only 15% of expert effort to achieve a precision of 0.95 while the baseline approach needs over 50% of expert efforts. Also, the performance of our approach when the questions are easy is better than in the setup with hard questions. This is expected and can be explained as follows. In the dataset with easy questions, most of the workers are able to give the correct answers, which makes the uncertainty in the dataset low. As

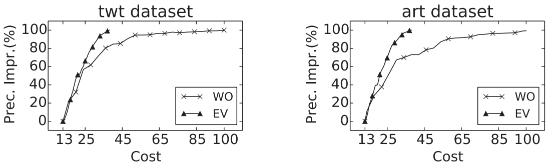

Figure 21: Effect of question difficulty on cost

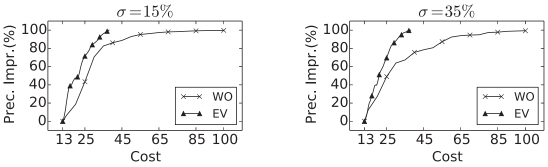

Figure 22: Effects of spammers on cost

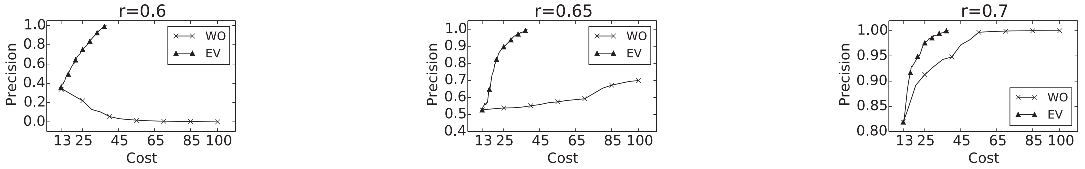

Figure 23: Effects of worker reliability on cost

a result, with the same amount of feedbacks, we can improve the precision higher than when the questions are hard.

D. COST TRADE-OFFS (CONT'D)

We complement the experiments reported in §6.8 by studying the effects of question difficulty, spammers, and worker reliability when comparing the *EV* approach with the *WO* approach.

Effects of question difficulty. In this experiment, we compare our EV approach with the WO approach with respect to the difficulty of the questions. We remove the answers from the answer matrix randomly such that 13 answers remain per question ($\phi_0 = 13$). Then, to simulate the addition of answers for the WO approach, we add the answers back to the questions. We fix the expert-crowd cost ratio to $\theta = 25$ and average the results over 100 experiment runs.

The experimental results are shown in Figure 21 where the X-axis depicts the normalized cost and the Y-axis measures the precision improvement of the deterministic assignment. The precision improvement of the EV approach is always higher than that of the WO approach, indicating that our EV approach is robust against the effects of question difficulty.

Effects of spammers. In this experiment, we analyze the effects of spammers by varying the percentage of spammers in the dataset from 15% to 35%. The experiment is conducted on the synthetic dataset with $\phi_0 = 13$, $\theta = 25$.

The results illustrated in Figure 22 show the benefits of using our approach with different percentages of spammers. The EV approach is able to achieve high precision improvement with a small amount of cost. For instance, when $\sigma = 35\%$, to improve the precision by 90%, a cost of 30 is required for the EV approach while the WO approach needs twice the amount. Also, the more spammers are part of the population, the better becomes the performance of the EV approach regarding the WO approach. For example, the difference in cost to achieve 90% precision improvement is about 10 when the percentage of spammers is 15%, but this increases three times to 30 as the percentage of spammers increases to 35%. Again, the reason is that as the percentage of spammer increases, the WO suffers from adding more answers as they are more likely to come from unreliable workers.

Effects of worker reliability. Worker reliability can affect the quality of crowd answers, thus also affects the cost model. If the worker reliability is high, the expert can spend less effort to give feedbacks as most of the answers are already correct. On the other hand, when the worker reliability is low, more feedbacks from the expert is required to achieve the same amount of precision. In this experiment, we analyze the effects of worker reliability on the cost of validating the crowd answers by varying the reliability of the normal workers from 0.6 to 0.7. Similar to the above experiment, we fix the following parameter: $\phi_0 = 13$, $\theta = 25$ and the workers population is simulated as discussed in §6.1.

The obtained results are illustrated in Figure 23, which highlights the relation between the cost normalized over each question and the precision of the deterministic assignment. Interestingly, when the reliability of the workers is 0.6, the precision of the deterministic assignment using the WO approach converges to 0 as we add more answers. The reason is that as we decrease the worker reliability, the average worker reliability becomes less than 0.5, which makes the precision converge to 0. This shows that adding more answers to the answer set may not improve but reduce the quality due to unreliable workers. When the reliability of the workers is 0.65, the precision of the deterministic assignment using the WO approach improves very slowly as the average reliability of the whole population is about 0.5. On the other hand, when the reliability of the workers is 0.7, the precision of the WO approach converges to 1. Yet, it requires higher cost to reach the same amount of precision as the EV approach. In summary, this experiment shows that our approach is robust against the reliability of the workers.

E. HARDNESS OF THE EFFORT MINIMIZATION PROBLEM

Now we consider a restricted version of Problem 1 in which we narrow down some problem conditions. First, instead of finding the minimal sequence, we consider the effort minimization as the identification of a minimal set $D = \{d_0, d_1, \ldots, d_n\} \in \mathcal{R}(\Delta, b)$. In other words, different sets of validation inputs will lead to different uncertainty values of the output. Second, we consider the reconciliation goal Δ as a thresholding condition defined on the joint entropy of the validated objects; i.e. $H(D) \geq \delta$. In other words, we select for validation the objects that lie on the edge of right and wrong, which is the key point to the effort minimization, rather than treating every object equally. As such, the effort minimization problem can be reformulated as:

$$\underset{D \subseteq O, |D| \leq k}{\arg\max} \; H(D) \tag{16}$$

This formulation can be interpreted as follows. We find a set of objects (with size as small as possible) for validation such that their joint entropy is maximal. In general, this problem is known to be NP-hard when the random variables of D are not independent [30], which is the case of our objects. Consequently, Problem 1 is harder than Equation16, given that finding the minimal sequence is difficult than finding the minimal set. Moreover, the goal Δ could be defined more complex. Due to the dependency between the objects via the crowd answers, satisfying Δ might need to look-ahead not only possible validation inputs of the objects but also the consequences of different orders of the validation.