Cost-Efficient Querying Strategies for the Crowd

Anja Gruenheid
ETH Zurich
anja.gruenheid@inf.ethz.ch

Besmira Nushi
ETH Zurich
nushib@inf.ethz.ch

Donald Kossmann
ETH Zurich
donaldk@inf.ethz.ch

ABSTRACT

To enhance data processing, crowdsourcing is a mechanism that has evolved recently and has been picked up by companies that refine large amounts of data through so-called crowd workers. Generally, there are two challenges to crowdsourcing: First, the crowd is an uncertain input source because a worker not necessarily knows the right answer or simply answers wrongly. Second, crowdsourcing comes at a monetary cost. As a result, finding ways to minimize the number of times the crowd is accessed is an essential line of work to make crowdsourcing a realistic option in the context of big data. To address these problems, crowd querying strategies have been devised which determine the task order and budget spent per worker task. Existing querying strategies focus on either improving certainty in the current result or optimizing the information gain by resolving the most uncertain parts of the dataset first. In this work, we first compare and evaluate both strategies after which we combine their core ideas into a hybrid querying strategy. We then show experimentally that it is a robust and cost-efficient alternative for querying the crowd.

1. INTRODUCTION

In the context of big data, data cleaning and data integration have become essential steps to provide analysis tools with useful input. Though automated methods can perform these tasks efficiently for many use cases, the state-of-the-art is still to tailor them to their respective use case. If no domain knowledge is available, crowdsourcing has been suggested as a way to obtain information for tasks such as entity resolution or picture classification which are intuitive for humans to solve but are hard problems for a computer. Figure 1 shows an example for such a task: Here, the crowd worker is asked to determine which animals belong to the same breed, i.e., sea lions and seals. For the full-grown animals (records r_2 and r_4) the task is easier than for the baby animals which have similar appearance characteristics (records r_1 and r_3).

Using the crowd as a decision mechanism poses two challenges: First, the quality of crowd responses has to be ensured and second, the cost (in $) of the crowd has to be minimized without lowering result quality. Quality is an obvious concern as the crowd is by far more unpredictable than an automated algorithm. Even unintentionally, workers can negatively influence the result through a) guessing or b) faulty answers. In this work, we will therefore look at querying strategies that repeatedly ask the crowd for an answer on (possibly the same) piece of data until it has been decided upon. To form a decision, we use a pair-wise, fault-tolerant decision function called MINMAX, [5], that applies a path-based decision mechanism on all information obtained from the crowd. To minimize cost, it leverages inference as a way to reduce the decision space of the crowd querying strategies. For instance, if records r_1 and r_3 in Figure 1

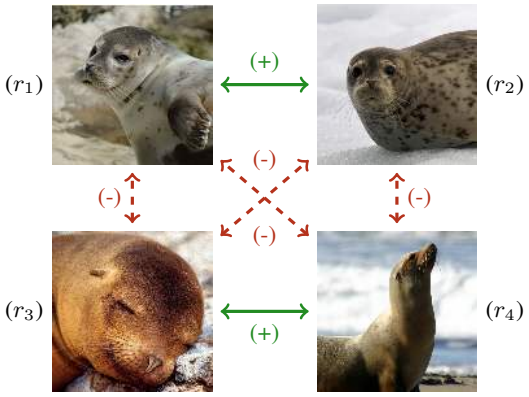


Figure 1: Animal classification problem.

are put into the same entity, it becomes unnecessary to ask twice for the connection between $\{r_1, r_3\}$ and r_2. Instead, the answer is inferred for r_1 if candidate pair (r_2,r_3) is solved and vice versa. To enable entity resolution on top of all pair-wise decisions, we then apply an adapted version of correlation clustering, [2], on the available decisions. Both these mechanisms are applied to all evaluated querying strategies equally and thus enable a fair comparison.

To determine the right order in which to ask the crowd, previous work has focused on two different crowd querying strategies: The first one is to reduce errors within the probed part of the dataset, the second strategy is to reduce uncertainty in the whole dataset, i.e., to retrieve more information on the uncertain parts of the dataset. In this work, we compare both strategies in terms of their cost and quality trade-off as well as how they resolve uncertainty within the dataset. More specifically, we make the following contributions:

- **Analysis**: Both, the *error reduction* and the *uncertainty reduction* strategy, have been proposed as querying strategies for crowdsourcing problems. We highlight and contrast their behavioral characteristics and show their (dis-) advantages in an experimental evaluation.
- **Hybrid strategy**: We propose a hybrid strategy based on our observations that leverages the ideas contained in both strategies. Its robustness and cost-efficiency is shown in the evaluation section.

This paper is organized as follows: We first discuss the two existing strategies and establish an alternative hybrid strategy. Afterwards, we evaluate the behavior of all three strategies given a perfect and noisy crowd. We will then introduce related work to compare our ideas with existing research after which we discuss future work.

2. QUERYING STRATEGIES

We will now discuss two different existing querying strategies as well as a new hybrid strategy that combines the benefits of the existing strategies to provide high quality results for low cost.

Error reduction strategy (ERS). The first strategy that has been established in prior research reduces the amount of noise in the part of the dataset that has been previously evaluated. Thus, it provides as correct as possible intermediate results at any point in time. The underlying idea of this strategy is that any algorithm using the crowd has to make certain that the decisions that are propagated into the entity resolution solution are in fact correct. It therefore favors correctness over completeness.

EXAMPLE 1. *Take our motivating example (Figure 1) and assume that there exists no information about any record pair. If the algorithm randomly picks (r_1,r_2) for crowdsourcing first, it then asks for this pair as often as it needs to be certain that this pair has been resolved correctly (i.e., it asks multiple times, until a quorum is reached etc.). If the decision is positive, i.e., the r_1 and r_2 are combined into one entity {r_1,r_2}, the candidate space of record pairs is adjusted accordingly because inference allows for effective search space reduction in that case. Alternatively, a negative decision is propagated to both entities that are affected by the new decision. This process is repeated until all record pairs have been resolved.*

Uncertainty reduction strategy (URS). The second technique that we evaluate here pursues the opposite objective from ERS: It will always ask those questions first for which it has the least confidence, continuing to ask until it has successfully removed all uncertainty for all candidate pairs. This strategy favors completeness over correctness, allowing for uncertain results to have an impact on the result entity resolution.

EXAMPLE 2. *In contrast to the previous approach, this method will first ask for a record pair exactly once, for example (r_1,r_2). The answer is then integrated into the clustering solution, thus a positive decision has direct impact on the clustering (here {r_1,r_2}). At this point, the algorithm has no knowledge about either ({r_1,r_2},r_3), ({r_1,r_2},r_4), or (r_3,r_4). Either one is asked next until the algorithm obtained some information on all pairs which are then further refined in descending order of uncertainty.*

Hybrid strategy (HS). Both previously explained techniques have advantages: While ERS guarantees high precision, URS is opportunistic and favors completeness which increases the quality gain if the crowd is knowledgable. To combine the best of both worlds, we propose a third strategy which is a hybrid between both. As positive decisions have the highest impact on the entity resolution solution (they impact both precision and recall), the hybrid strategy resolves positive decision candidates cautiously, following ERS. Negative decision candidates on the other hand are processed according to URS to avoid spending budget on a potentially redundant request.

EXAMPLE 3. *If record pair (r_1,r_2) is again selected as initial crowdsourcing pair and the crowd returns one positive vote,* HS *will continue asking for this pair until it is either resolved or not a positive candidate anymore (i.e., the number of positive matches the number of negative votes). When {r_1,r_2} is merged into one entity, and the next candidate pair is randomly selected as ({r_1,r_2},r_3), the algorithm decides to crowdsource (r_1,r_3) once. If the decision is negative, it will then proceed by asking the next pair in the queue until all uncertain pairs are resolved in the same manner.*

STRATEGY 1. Hybrid querying strategy.

```
1  Queue Q_p                                  // queue for positive pairs
2  Queue Q_n                                  // queue for negative pairs
3  ---------------------------------------------------------------
4  Function: ADD(ρ_ij, s_ij)                  // inserts a record pair
5  if s_ij then
6  |  Q_p.add(ρ_ij)
7  else
8  |  Q_n.add(ρ_ij)
9  ---------------------------------------------------------------
10 Function: NEXT : record pair               // returns the next record pair
11 if Q_p ≠ ∅ then
12 |  return Q_p.first
13 return Q_n.first
```

Implementing HS is straightforward as shown in Strategy 1. Both ERS and URS can be represented as queues that are ordered according to their respective objective. In HS, ERS is then used as queue for positive pairs between records (Q_p) while URS is used for the negative queue Q_n. Whenever a record pair $\rho_{ij} = (r_i, r_j)$ is now inserted into this hybrid queuing system (Function ADD, Line 4), it is inserted into either the positive or negative queue based on its current boolean similarity value s_{ij}. The similarity value of a record pair is obtained by evaluating ρ_{ij} with a decision function and it represents the certainty that those records belong to the same entity given the current voting state of ρ_{ij}. To extract the next pair to crowdsource, the hybrid NEXT function (Line 10) first searches for a match in Q_p. If that request is unsuccessful, it returns an unresolved candidate pair from Q_n.

3. EXPERIMENTS

In this section, we will evaluate the different querying strategies with a real-world dataset that allows us to show both their strengths and weaknesses. It contains pictures of 13 landmarks (i.e., squares, buildings etc.) in Paris, France, and Barcelona, Spain. We observe similar though less skewed behavior when running experiments on synthetic datasets. Part of future work is to verify our findings with further real-world datasets.

Dataset & crowd. The dataset used in this evaluation consists of 266 pictures that show 13 different landmarks in Paris and Barcelona[1]. The maximum number of pictures per landmark is 43, while the smallest entity is described by 7 pictures. To enable repeatable experiments, we established an exhaustive dataset, asking the crowd[2] for every possible picture combination exactly 10 times. The success of the different strategies is evaluated by comparing their final entity resolution solution to its gold standard solution by computing precision, recall, and f-measure, [1]. Every experiment is repeated multiple times to obtain averaged values.

MINMAX decision function. To enable a fair comparison between all three querying strategies, they all use the same decision function to decide about the similarity of two records resp. entities. MINMAX is a path-based mechanism that computes the certainty of a record pair as the difference between the positive and negative path connecting it, [5]. Thus, a record pair is uncertain if the difference between these path scores is small while certainty increases

[1]http://www.systems.ethz.ch/research/researchareas/dbs/crowdDB

[2]The platform used for these experiments is Amazon Mechanical Turk.

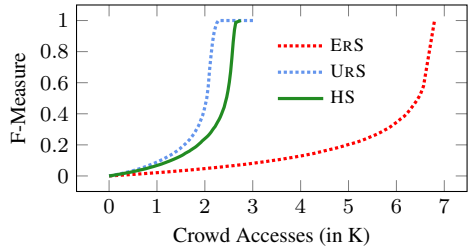


Figure 2: Landmarks dataset, evaluated with a perfect crowd for varying querying strategies.

if the difference increases. In all shown experiments, we assume certainty if the crowd returns either answer three times more than its alternative.

3.1 Perfect crowd experiments

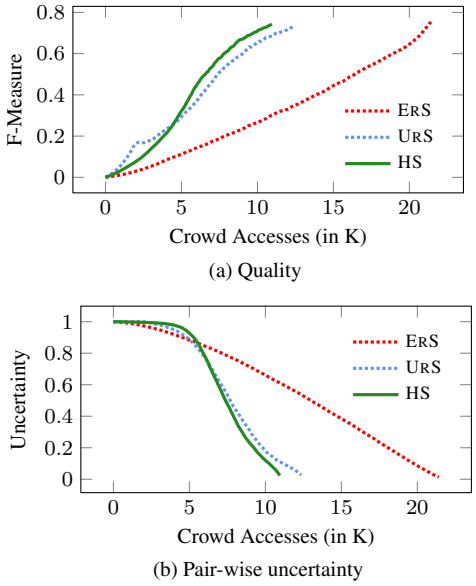
This experiment serves as baseline to compare the varying strategies in a noise-free environment. As shown in Figure 2, URS consistently achieves the best cost-quality trade-off which is as expected. Through uncertainty reduction, this strategy spreads the budget over all pairs rather than focusing on one pair specifically like ERS (or HS for positive candidate decisions). Votes are immediately propagated into the entity resolution solution resulting in the best overall quality gain per cost unit. The exact trade-off between the strategies is influenced by the dataset structure: An increase in the number of records per entity enables a better performance of ERS as its investment per comparison becomes more cost-effective. Similarily, a decrease leads to more negative decision candidates which benefits URS, lowering the quality gain per cost unit for ERS. As hybrid solution, HS will incorporate the behavior of both ERS and URS and will always be at least as good as ERS and at most as good as URS if the crowd is perfect.

3.2 Noisy crowd experiments

Figure 3 shows experiments on the same dataset obtained by using answers from a real-world instead of the synthetic, perfect crowd. Similar to the baseline experiment, we observe that the quality of the entity resolution solution improves if more cost is invested into resolving it (Figure 3a). The quality gain of ERS is rather constant while both URS and HS show a better cost vs. quality trade-off with HS dominating URS if more than 5,000 cost units are spent. The reason for the increased performance of HS lies in the higher correctness that is achieved for this approach by requiring positive candidates to be certain. Therefore, the intermediate results contain less noisy decisions which leads to a better overall quality. In Figure 3b, we observe that ERS reduces pair-wise uncertainty predictably. In comparison, the other approaches first gather information before refining the entity resolution solution, which is the cause for the slightly lower initial uncertainty reduction. It increases when the positive candidate pairs have been successfully resolved because they allow for an efficient propogation of negative decisions.

3.3 Crowd reliability

Comparing experiments with a synthetic crowd done on the same dataset with the experiments shown here, we observe that the synthetic crowd may help to identify tendencies of real-world crowds but that answers obtained from real crowd workers are rarely follow a pre-defined distribution. More specifically, we observe that given the context of crowd entity resolution, crowd errors can be categorized into two different types: First, the crowd makes false positive mistakes, i.e., they answer positively even though the ground truth



(a) Quality

(b) Pair-wise uncertainty

Figure 3: Landmarks dataset, evaluated with a real-world crowd for varying querying strategies.

is a negative decision. Second, the crowd makes false negative mistakes. In this example dataset, a false positive rate of 5.16% and a false negative rate of 26.4% can be observed. Obviously, both of these error types may vary according to the dataset and the crowd answering the tasks.

3.4 Dataset variations

We generally observe that similarily to the experiments with the perfect crowd, ERS will benefit from an increase of records per entity in noisy environments. It remains stable in terms of quality and uncertainty development but its cost comparatively decreases, even allowing for a lower termination budget than URS in certain cases. Due to more positive decision candidates and thus a better use of inference, all strategies improve quality gain in such a setup. Nevertheless, HS continues to dominate both baseline strategies.

4. RELATED WORK

Crowdsourcing is a technique integrating human input into the information retrieval process to solve tasks that cannot be answered easily by computers. It has gained momentum over the last years and has been studied in the context of entity resolution, grouping, and other tasks that are typically performed by database or data integration systems. As a result, a variety of systems combining classic database management with crowdsourcing have been proposed in the past, [3, 6, 7], alongside with respective follow-up work on database operators. This work is related to research done on crowdsourcing entity resolution in the database community, [9, 10], where ERS and URS are proposed respectively. Whereas this line of work assumes the crowd to be perfect, we acknowledge that it is not and evaluate our querying strategies accordingly. As we have shown in our experimental evaluation, dropping this assumption has a strong impact on the performance of these strategies.

Alternative approaches to crowdsourced entity resolution in the machine learning community accept noisy information from the crowd but do not pursue incremental crowdsourcing strategies. Instead, crowdsourcing is used on a small portion of the dataset to approximate the remaining tasks [4, 11]. Incremental improvement

of information is discussed in information theory in the context of active learning [8] in which both of our base lines have also been presented. Here, the strategies base their decision on a labeled subset of the data whereas the query strategies that we represent react online to the (false) input from the crowd and incrementally request more information based on their observations. Note that in the context of this work, these strategies are thus compared in an unsupervised learning situation.

5. CONCLUSION & FUTURE WORK

We have shown in this work shortly how traditional querying strategies behave in perfect and uncertain environments. The strategies present in current literature follow two opposite approaches: One aims to cautiously evaluate the candidate space, verifying its results before proceeding. In contrast, the second strategy asks for the most uncertain pairs first to optimize its information gain. We show that combining both approaches in fact benefits both quality gain and uncertainty reduction per unit of cost in noisy environments.

For future work, we will further optimize the proposed hybrid strategy. As we have shown in our experimental evaluation, it is in fact less good than a pure strategy if the crowd is noiseless. One idea is therefore to explore dynamic ways to measure and to learn about the quality of the crowd, integrating that knowledge into the querying strategy. Furthermore, certainty is achieved through asking the crowd multiple times for verification. As a result, minimizing the underlying questions per entity pair in an effective manner is also a relevant line of work.

6. REFERENCES

[1] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[2] N. Bansal, A. Blum, and S. Chawla. Correlation Clustering. In *Machine Learning*, pages 238–247, 2002.

[3] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. CrowdDB: Answering Queries with Crowdsourcing. In *SIGMOD Conference*, pages 61–72, 2011.

[4] R. Gomes, P. Welinder, A. Krause, and P. Perona. Crowdclustering. 2011.

[5] A. Gruenheid, D. Kossmann, S. Ramesh, and F. Widmer. Crowdsourcing Entity Resolution: When is A=B? Technical Report, ETH Zurich, 2012.

[6] A. Marcus, E. Wu, D. R. Karger, S. Madden, and R. C. Miller. Demonstration of Qurk: a Query Processor for Human Operators. In *SIGMOD Conference*, pages 1315–1318, 2011.

[7] H. Park, H. Garcia-Molina, R. Pang, N. Polyzotis, A. Parameswaran, and J. Widom. Deco: A System for Declarative Crowdsourcing. *Proc. VLDB Endow.*, 5(12):1990–1993, Aug. 2012.

[8] B. Settles. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[9] J. Wang, G. Li, T. Kraska, M. J. Franklin, and J. Feng. Leveraging Transitive Relations for Crowdsourced Joins. In *SIGMOD Conference*, pages 229–240, 2013.

[10] S. E. Whang, P. Lofgren, and H. Garcia-Molina. Question Selection for Crowd Entity Resolution. *PVLDB*, 6(6):349–360, 2013.

[11] J. Yi, R. Jin, A. K. Jain, and S. Jain. Crowdclustering with Sparse Pairwise Labels: A Matrix Completion Approach. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.