# BATC - A Benchmark for Aggregation Techniques in Crowdsourcing

Nguyen Quoc Viet Hung    Nguyen Thanh Tam    Lam Ngoc Tran    Karl Aberer
École Polytechnique Fédérale de Lausanne
1015 Lausanne, Switzerland
{quocviethung.nguyen,tam.nguyenthanh,ngoc.lam,karl.aberer}@epfl.ch

## ABSTRACT

As the volumes of AI problems involving human knowledge are likely to soar, crowdsourcing has become essential in a wide range of world-wide-web applications. One of the biggest challenges of crowdsourcing is aggregating the answers collected from crowd workers; and thus, many aggregate techniques have been proposed. However, given a new application, it is difficult for users to choose the best-suited technique as well as appropriate parameter values since each of these techniques has distinct performance characteristics depending on various factors (e.g. worker expertise, question difficulty). In this paper, we develop a benchmarking tool that allows to (i) simulate the crowd and (ii) evaluate aggregate techniques in different aspects (accuracy, sensitivity to spammers, etc.). We believe that this tool will be able to serve as a practical guideline for both researchers and software developers. While researchers can use our tool to assess existing or new techniques, developers can reuse its components to reduce the development complexity.

**Categories and Subject Descriptors:** H.3.3 [Information Search and Retrieval]: Selection process
**General Terms:** Algorithms, Design, Experimentation.
**Keywords:** benchmark, crowdsourcing, aggregate technique.

## 1. INTRODUCTION

Today, crowdsourcing becomes a promising methodology to overcome various problems that require human knowledge such as image labeling, text annotation, and product recommendation [1]. A wide range of applications (e.g. ESP game, reCaptcha, and Freebase [2]) have been developed on top of more than 70 crowdsourcing platforms [1] such as Amazon Mechanical Turk and CloudCrowd. The rapid growth of such crowdsourcing applications opens up a variety of technical and social challenges [2].

One of the most critical issues of crowdsourcing is to aggregate different answers given by *crowd workers*. This is a challenging task because of two reasons: (i) the workers might have wide ranging levels of expertise and (ii) the questions may vary in different levels of difficulty. While the former leads to high contradiction and uncertainty in the answer set, the latter renders some difficulties in distinguishing between truthful workers and malicious workers. To fully tackle this challenge, a rich body of research on answer aggregation has developed different techniques.

However, each work often reported its superior performance generally using a limited variety of data sets of evaluation methodologies. As a result, understanding the performance implications of these techniques, for a given type of application, is difficult to comprehend. Therefore, we present the Benchmark for Aggregate Techniques in Crowdsourcing (BATC) with three functionalities:

- **Choose well-suited techniques.** Each technique has distinct performance characteristics and there is no absolute winner that outperforms the others in every case. BATC will serve as a practical guideline for how to select well-suited techniques on particular application scenarios.
- **Guide to select appropriate parameters.** BATC also allows users to vary configurable parameters and visualize their effects. Through empirical observations, the users can select an appropriate parameter configuration for their applications.
- **Reduce the development complexity.** Since crowdsourcing platforms rarely support the answer aggregation, application developers have to re-implement existing aggregate techniques. However, it might be challenging for them to understand those techniques. Using BATC as a reusable framework, the developers can reduce the development time.

To support these functionalities, we design our tool with three main features: (i) simulate the crowd, (ii) re-implement state-of-the-art aggregate techniques within a common framework, and (iii) evaluate these techniques with different metrics. To the best of our understanding, BATC is the first system to provide these attractive features. In the following, we first describe the system overview and implementation details in Section 2. Next, Section 3 presents some demonstrations. Finally, Section 4 summarizes the paper.

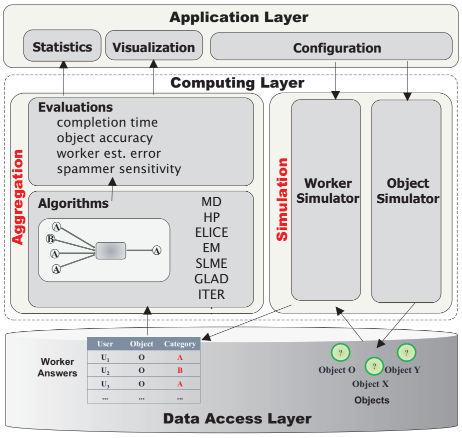
## 2. SYSTEM DESIGN

Figure 1 illustrates the simplified architecture of our framework—which is built upon three layers: *data access layer, computing layer* and *application layer*. The data access layer abstracts underlying data objects, which could be synthetic or real data. The application layer provides an interactive GUI to users. The computing layer consists of two important modules:

- **Aggregation module:** runs the aggregate techniques implemented and plugged into the framework. The aggregation module is divided into two components: (i) *algorithm component*—already implemented most representative algorithms, including MD [1], HP [6], ELICE [5], EM [3], GLAD [9],

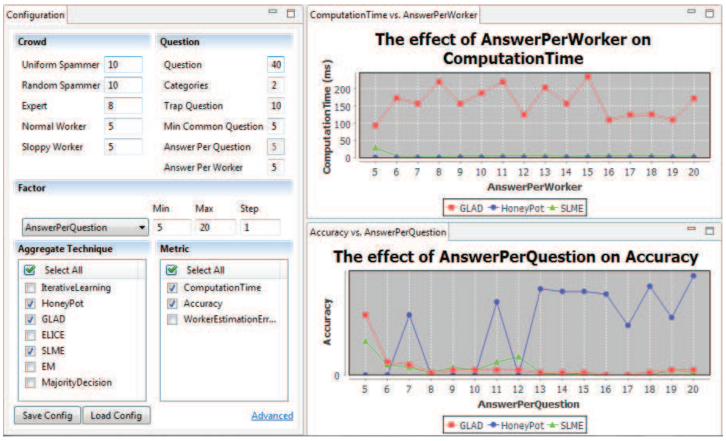
[1] http://www.crowdsourcing.org

*SIGIR'13,* July 28–August 1, 2013, Dublin, Ireland.
ACM 978-1-4503-2034-4/13/07.



| User | Object | Category |
|---|---|---|
| $U_1$ | O | A |
| $U_2$ | O | B |
| $U_3$ | O | A |
| ... | ... | ... |

**Figure 1: Benchmarking framework**



Crowd

Uniform Spammer 10
Random Spammer 10
Expert 8
Normal Worker 5
Sloppy Worker 5

Question

Question 40
Categories 2
Trap Question 10
Min Common Question 5
Answer Per Question 5
Answer Per Worker 5

Factor

AnswerPerQuestion ▾ Min 5 Max 20 Step 1

Aggregate Technique

☑ Select All
☐ IterativeLearning
☑ HoneyPot
☑ GLAD
☐ ELICE
☑ SLME
☐ EM
☐ MajorityDecision

Metric

☑ Select All
☑ ComputationTime
☑ Accuracy
☐ WorkerEstimationErr...

**Figure 2: User Interface**

SLME [7], ITER [4] and (ii) *evaluation component*—defined various performance metrics to evaluate aggregate techniques. Note that with this component-based architecture, new techniques and new metrics can be easily plugged in.

- **Simulation module:** simulates the crowdsourcing process in which each worker answers a set of various questions. This module contains two components: (i) *worker simulator*—simulates five types of workers according to the classification in [8], including *expert*, *normal*, *sloppy*, *random spammer*, *uniform spammer* and (ii) *question simulator*—simulates the process of generating answers of worker for two types of questions: binary-choice and multiple-choice.

Our benchmarking tool is developed as an Eclipse Rich Client Platform (RCP) application. The runnable file and the demo video of this tool are publicly available at our website [2].

## 3. DEMONSTRATION

We will demonstrate the benchmarking capabilities of BATC as described in Section 1. Users are able to simulate crowdsourcing process that involve different types of workers, questions, and aggregate techniques. They can also choose real datasets. To provide in-depth analysis, we characterize the aggregate techniques evaluated in the benchmark using five measures:

- **Computation time:** is an important aspect, as various crowdsourcing applications often have constraints on computing speed, or limitations in using server resources.
- **Accuracy:** is defined as the percentage of questions which are correctly aggregated. The higher accuracy, the higher power of aggregate technique.
- **Sensitivity to spammers:** In reality, spammers always exist in online community, it is important for crowdsourcing applications to know how each aggregate technique performs when the worker answers are not trustworthy.
- **Compatibility to multi-labeling:** In the literature, many applications are designed for multiple-choice questions. Therefore, it is important to know the compatibility of aggregate techniques and their performance behaviors in this setting.
- **Worker estimation error:** This measurement is important in some applications such as worker profiling. To reflect this aspect, we represent the estimation precision of worker expertise using mean absolute error. The lower error, the better estimation of worker quality.

BATC visualizes benchmarking results in several views, allowing users to compare multiple settings and choose the best-suited technique for their applications. Figure 2 depicts the interactive GUI of BATC that supports several operations such as zooming, panning, and dragging&dropping. In the BATC's interface, users may explore different parameter configurations and choose appropriate values for their application requirements.

## 4. SUMMARY

We have developed a benchmarking tool that focuses on providing in-depth analyses and practical guidelines. The target users (researchers and developers) can use BATC to select and configure well-suited aggregate techniques for a potential application. This tool is built upon a component-based architecture, in which new techniques and new measurements can be easily plugged. As the source code as well as the demonstrations are publicly available, we expect that our reusable framework will be refined and improved by the research community, in particular when more data become available, more experiments are performed, and more techniques are integrated into the framework in the future.

## Acknowledgement

This research has received funding from the NisB - European project (FP7/2007-2013) under grant agreement number 256955.

## References

[1] L. von Ahn. "Human computation". In: *Design Automation Conference*. 2009.

[2] A. Doan et al. "Crowdsourcing systems on the World-Wide Web". In: *Commun. ACM* (2011).

[3] P. G. Ipeirotis et al. "Quality management on Amazon Mechanical Turk". In: *HCOMP*. 2010.

[4] D. Karger et al. "Iterative learning for reliable crowdsourcing systems". In: *NIPS* (2011).

[5] F. Khattak et al. "Quality Control of Crowd Labeling through Expert Evaluation". In: *NIPS 2nd Workshop* (2011).

[6] K. Lee et al. "The social honeypot project: protecting online communities from spammers". In: *WWW*. 2010.

[7] V. Raykar et al. "Supervised learning from multiple experts: Whom to trust when everyone lies a bit". In: *ICML* (2009).

[8] J. Vuurens et al. "How much spam can you take? an analysis of crowdsourcing results to increase accuracy". In: *SIGIR*. 2011.

[9] J. Whitehill et al. "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise". In: *NIPS* (2009).

[2] https://code.google.com/p/benchmarkcrowd