Crowdsourced Query Processing on Microblogs
Course Project Report

Zhou Zhao
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Hong Kong, China
{zhaozhou}@cse.ust.hk

Abstract—Databases often offer poor answers with respect to judgemental queries such as asking the best among the movies shown in recent months. Processing such queries requires human input for providing missing information in order to clarify uncertainty or inconsistency in queries. Nowadays, it is common to see people seeking answers on micro-blogs through asking or sharing questions with their friends. This can be easily done via smart phones, which diffuse a question to a large number of users through message propagation in microblogs. This trend is important and known as Crowdsourcing. To tackle these issues, we propose a new problem of minimizing the cost of a crowdsourced query processing on microblogs. We propose a new problem of minimizing the cost of a crowdsourced query processing on microblogs, given the specified accuracy threshold. However, we prove that this problem is NP-hard in this paper. To tackle this problem, we propose a greedy algorithm for mining the crowdseed set for query diffusion on microblogs while cost of the set can be minimized. We validate the effectiveness and efficiency of our algorithm using real datasets.

I. INTRODUCTION

In recent years, Crowdsourcing databases [1],[4],[2],[5] have attracted substantial interest in the database research community. Many fundamental infrastructures are proposed to support various kinds of query processing on the crowd. Amazingly, the wisdom of crowds has been proved to outperform computer programs at various kinds of tasks, especially for image tagging, natural language processing and so on. Crowdsourcing relies on human workers to complete, at least partially, the job of query processing.

However humans are prone to error, which may provide extremely poor quality crowdsourcing results. To address the above problems, crowdsourcing applications often enroll a number of workers to process the replicated queries. If the collected results from workers are conflicting, the majority vote is adopted to determine which is correct. Currently, crowdsourcing applications duplicate the issued queries and publish them on designed platforms, such as Amazon MTurk1, CrowdFlower2 and the like.

In this paper, we formulate a new problem of CrowdSeed Selection that enables crowdsourced queries to be processed on microblogs. Typically, we focus on addressing the main issues:

- **Conflicting Answers.** The replication of queries may not fully solve this problem. If the number of replicated queries are few, we may not have enough confidence to infer a reliable answer. However, if we duplicate too many queries, we may have to suffer high cost [3].
- **Query Cost.** Some voters may not be reluctant to process the query until they can receive some reward. So, we aim to minimize the cost of processing the crowdsourced query on microblogs, given a specified accuracy threshold.
- **Query Diffusion.** The word of mouth effect is an important factor of microblogs. We aim to utilize the user influence to diffuse the crowdsourced query on microblogs.

There have already been some papers addressing the problem of conflicting answers [2],[5]. However, none of them resolves all the above issues satisfactorily. Furthermore, we mainly study the problem of crowdsourced decision-making query processing on microblogs.

The underlying idea of our approach is as follows. Given a crowdsourced decision-making query $Q$ and the specified accuracy threshold $\alpha$, we study to process this query on the graph of microblogs $G = (V,E)$. First, we estimate the number of voters is needed to process this query such that the aggregate answers is able to satisfy the specified accuracy threshold $\alpha$. We denote the minimum number of voters as $\tau$. Next, we devise a method to diffuse the crowdsourced query $Q$ on microblogs in order to reach $\tau$ votes. In this paper, we aim to find a crowdseed set of voters as the seed for query diffusion while monetary cost is minimized. We consider this set as $S_{\text{min}}$. Then, we “tweet” the query to the voters in set $S_{\text{min}}$ as well as give them the incentive rewards. As a result, the query $Q$ will be “retweet”ed from these voters to others on microblogs. Finally, we collect the answers from the crowd. We aggregate the conflicting answers and choose the correct one.

We illustrate an example of crowdsourced query processing in Figure 1. We issue a decision-making query “Is the City of Light Paris?” with the specific accuracy threshold to 0.9. Some voters are selected in the crowdseed set to give votes as well as diffuse this query to others. We demonstrate the aggregation of answers of this query in Figure 2.

**Contributions.** We mainly tackle the problem of crowdsourced query processing on microblogs. Specifically, we make

---
1https://www.mturk.com/mturk/welcome
2http://crowdflower.com/
The formula of majority voting is given by the following contributions.

1. We first formulate the problem of crowdsourced query processing on microblogs.
2. We prove the problem of CrowdSeed Selection to be NP-hard and the computation of Query Diffusion is #P-hard.
3. We propose a greedy algorithm to tackle the problem of CrowdSeed Selection. Then, we devise a sampling algorithm to estimate the Query Diffusion that can provide an error bound.
4. We validate the efficiency and effectiveness of our algorithm using real datasets.

This paper is organized as follows. Section 2 introduces the crowdsourcing models. Section 3 formulates the problem. Section 4 then presents our algorithm while Section 5 presents the experimental results. We conclude the paper in Section 6.

II. CROWDSOURCING MODEL

In this section, we propose some models for crowdsourced query processing on microblogs used in our subsequent discussion of query processing on microblogs.

A. Majority Voting Rule

Given the same query, the attitudes among crowd may be different. As a result, the human answers for the query posed on microblogs may be conflicting. To resolve this problem, we employ an aggregation schema called majority voting rule. The formula of majority voting is given by

\[ f(V) = \begin{cases} 1 & \text{if } \sum_{v_i \in C} v_i \geq \frac{\tau + 1}{2} \\ 0 & \text{otherwise} \end{cases} \]

where the voter \( v_i \) is a binary random variable (i.e. either 0 or 1) and \( V \) represents a collection of voters with size \( \tau \). Among the conflicting answers from the voters, we choose the one that is supported by more than half voters.

However, the output of majority voting rule may not be reliable if the number of voters is few. On the other side, the cost may be very huge if we enroll too many voters. To tackle this problem, we propose a probabilistic model to estimate the accuracy of majority voting rule.

Suppose the accuracy of voters that have processed the query are \( \{ a(v_1), \ldots, a(v_n) \} \), where \( a(v_i) \) is the probability of voter \( v_i \) giving the correct answer. The output answer by majority voting is correct only when at least half of the voters provide the correct answers.

Given a set of voters \( V \), we can estimate the correctness probability of majority voting rule, denoted as \( Pr(f(V)) \). We denote that the set \( \mathcal{S}_A \) consists of the sets of users who give the correct answer. Suppose that we have \( \tau \) voters giving the votes, then the correctness probability of majority voting rule is given by

\[
Pr(f(V)) = Pr(|\mathcal{S}_A| \geq \frac{\tau + 1}{2})
= \sum_{k=\frac{\tau+1}{2}}^{\tau} \sum_{S_A \in F_k} \prod_{v_i \in \mathcal{S}_A} a(v_i) \prod_{v_j \notin \mathcal{S}_A} (1 - a(v_j))
\]

(2)

where \( F_k \) is a set of subsets with size \( k \). For example, given that we have a collection of votes from three voters, then \( F_2 = \{ \{ v_1, v_2 \}, \{ v_1, v_3 \}, \{ v_2, v_3 \} \} \).

However, it may be infeasible to compute the correctness probability of majority voting rule by Equation 2 since it is hard to estimate the accuracy of each voter in a microblog. To tackle this problem, we turn to utilize the professionalism of the microblog to compute the expectation of Equation 2. We consider the professionalism of a microblog using the average accuracy of the voters in it, denoted as \( \mu \).

Then, the minimum number of voters is given by

\[
\tau \geq -\frac{\ln(1 - \alpha)}{2(\mu - \frac{1}{2})^2}
\]

(3)

where \( \alpha \) is the specified expected correctness of majority voting rule.

Proof omitted for brevity.

B. Crowdsourced Query Cost

However, humans are not always altruistic voters. Sometime, the voters need some incentive reward to process the crowdsourced query \( Q \). The existing crowdsourcing platforms pay each user a fixed amount of money for completing a query. However, we can find that the expected cost of each voter \( v_i \) to process and diffuse the crowdsourced query may be different. In the real life, the voters process and “tweeting” the crowdsourced query only when their expected reward has been
satisfied. Thus, the monetary cost of a crowdsourced query $Q$ is given by

$$\text{cost}(Q) = m(S) = \sum_{v_i \in C} m(v_i)$$

(4)

where $m(v_i)$ is the price set by voter $v_i$ and $S$ is the crowdseed set for diffusing the query. We only pay the voters in the seed set $S$ and make the query diffuse by the influence of the seeds (i.e. set $S$).

C. Crowdsourced Query Diffusion

We study the model for crowdsourced query diffusion on microblogs based on word of mouth effect. We consider that the voter $v_j$ votes and further diffuses the decision-making crowdsourcing query when the voter $v_i$ is able to influence $v_j$ ($I(v_i, v_j)$). The probability of the influence between two voters is estimated by the similarity of their friends. The friend similarity is estimated by the weighted Jaccard Distance, given by

$$\Pr(I(v_i, v_j)) = \frac{|N(v_i) \cap N(v_j)|}{|N(v_i) \cup N(v_j)|}$$

(5)

where $N(v_i)$ is the set of voter $v_i$’s friends and $\lambda$ is a weight depending on the type of the microblogs. If two voters share a lot of friends, they may be close friends. Furthermore, it is easier to diffuse query from one voter to another.

To model the crowdsourced query diffusion on microblogs, we consider a probabilistic graph $G = (V, E)$ where each edge is associated with an influence probability of two voters (i.e. $\Pr(e_{ij}) = \Pr(v_i, v_j)$). Given a crowdseed set $S$, the query diffusion probability on voter $v_k$ is given by

$$\Pr(I(v_k) | S) = \sum_{X} \Pr(X) R_X(v_k, S)$$

(6)

where $X$ is one flipping of all the edges in the probabilistic graph $G$. The $R_X(v_k, S)$ is the reachability between the voter $v_k$ and the set $S$. Then, the expected query diffusion over graph $G$ is given by

$$\delta(G | S) = \sum_{i=1}^{n} \Pr(I(v_i) | S)$$

(7)

where $n$ is the number of nodes in graph $G$.

III. PROBLEM DEFINITION

Using the proposed models above, we define the problem of CrowdSeed Selection below.

Problem 1: Given a probabilistic graph $G(V, E)$ of the microblog and a crowdsourced query $Q$, we want to find a crowdseed set for a crowdsourced query diffusion such that the query cost is minimized and the expected accuracy is larger than $\alpha$.

However, we find that the complexity of this problem is NP-hard and we claim that

Theorem 1: The problem of CrowdSeed Selection on microblogs is NP-hard.

Proof omitted for brevity.

Algorithm 1 CrowdSeed($G$, $\alpha$, $\mu$)

Input: $G = (V, E)$: a probabilistic graph;
Input: $\alpha$: specified accuracy threshold;
Input: $\mu$: professionalism of the Microblog;
Output: $S$: a crowdSeed set;

1: number of diffused queries: $\tau \geq \frac{-\ln(1 - \alpha)}{2(\mu - \frac{1}{2})^2}$
2: crowdseed set $S \leftarrow \emptyset$
3: while $\delta(G | S) < \tau$ do
4: \hspace{1em} $v_i = \arg \max_{v_i} (\Delta_{S \cup \{v_i\}} - \Delta_S)$
5: \hspace{1em} $S \leftarrow S \cup \{v_i\}$
6: end while
7: return $S$

However, we also find that even the computation complexity of the expected query diffusion of a crowdseed set $S$ is #P-hard and we claim that

Theorem 2: The computation complexity of expected query diffusion of a crowdseed set $S$ is #P-hard.

Proof omitted for brevity.

In this paper, we aim to mine the crowdseed set $S$ from the microblog such that: (1) we could have at least a collection of $\tau$ votes. (2) The monetary cost of the seed set $S$ is minimized.

IV. ALGORITHM

In this section, we first introduce a greedy algorithm to tackle the problem of CrowdSeed Selection on microblogs. Then, we propose a sampling-based algorithm to estimate the expected query diffusion of the set $S$.

V. EXPERIMENT

We evaluate the effectiveness and efficiency of our algorithm using four real datasets: DBLP, Epinions, HEP and Amazon, which can be obtained from our website. The statistics of these datasets can also be found in our website. We also propose three baseline algorithms such as Random selection algorithm, Degree-based selection algorithm and Centrality-based selection algorithm.

We demonstrate the performance of our algorithm on different specified error rate thresholds by $10^{-1}, 10^{-2}, \ldots, 10^{-6}$ (i.e. $1 - \alpha$). Next, we synthetically generate the monetary cost of each voter $v_i$ on these social graphs using a uniform distribution $\mathcal{U}(0, 100)$.

Number of Seeds. We study the size of the crowdseed set by comparing the number of seeds mined by different algorithms in Figures 3(a), 3(b), 3(c) and 3(d). To mitigate the unreliability of majority voting rule, we have to enroll many voters to improve its accuracy. Furthermore, the query diffusion of a crowdseed set is proportional to its size. Thus, the output number of seeds by all the algorithms increases when the error rate threshold decreases. We notice that the performance of the Greedy algorithm is stable on all datasets.

Query Cost. We illustrate the quality of different algorithms by comparing the estimated query cost in Figures 4(a),

3http://www.cse.ust.hk/~zhaozhou/data/crowdseed.zip
4(b), 4(c) and 4(d). We are able to observe that the query cost of our algorithm is much smaller than other algorithms on all datasets. This experimental result validates that the Greedy algorithm can minimize the cost of the crowdsourced query effectively on microblogs.

**Exec Time.** We study the efficiency of different algorithm by comparing their running times in Figures 5(a), 5(b), 5(c) and 5(d). We vary the error rate threshold and record the running time for each algorithm. By taking the advantages of the efficient implementation, we avoid the time cost of re-sampling deterministic graph at iteration such that the Greedy algorithm is very efficient even on very small error rate.

**Memory Cost.** To evaluate the space utility of our algorithm, we conduct a set of experiments in Figures 6(a), 6(b), 6(c) and 6(d). We consider the memory occupied by each algorithm for the comparison. We are able to observe that our algorithm has the least memory cost among these experiments. The memory cost of algorithm is mainly for the storage of a collection of sampled deterministic graphs at the beginning of the algorithm.

**VI. CONCLUSIONS**

In this paper, we explore a new issue of crowdsourced query processing on microblogs. We aim to minimize the cost of the crowdsourced query processing on microblogs while the aggregated answer satisfy the specified accuracy threshold \(\alpha\). We first study the query diffusion model and explain the problem of CrowdSeed Selection on microblogs. However, we prove that this problem is NP-hard and the computation of query diffusion given a crowdseed set \(S\) is \#P-hard. Next, we propose a greedy algorithm for the problem of CrowdSeed Selection. Then, we devise a sampling algorithm to compute the query diffusion of the selected crowdseed set. We also derive an error bound for the proposed sampling algorithm. We validate the performance of our algorithm using real datasets. The experimental results demonstrate that our algorithm can minimize the cost of the crowdsourced query effectively.

**REFERENCES**


