2017 IEEE 33rd International Conference on Data Engineering

CrowdFusion: A Crowdsourced Approach on Data Fusion Refinement

Yunfan Chen, Lei Chen
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Hong Kong SAR, China
{ychenbx, leichen}@cse.ust.hk

Chen Jason Zhang
Hong Kong University of Science and Technology and
Shandong University of Finance and Economics, China
czhangad@cse.ust.hk

***Abstract*—Data fusion has played an important role in data mining because high quality data is required in a lot of applications. As on-line data may be out-of-date and errors in the data may propagate with copying and referring between sources, it is hard to achieve satisfying results with merely applying existing data fusion methods to fuse Web data. In this paper, we make use of the crowd to achieve high quality data fusion result. We design a framework selecting a set of tasks to ask crowds in order to improve the confidence of data. Since data are correlated and crowds may provide incorrect answers, how to select a proper set of tasks to ask the crowd is a very challenging problem. In this paper, we design an approximation solution to address this challenge since we prove that the problem is at NP-hard. To further improve the efficiency, we design a pruning strategy and a preprocessing method, which effectively improve the performance of the proposed approximation solution. We verify the solutions with extensive experiments on a real crowdsourcing platform.**

I. Introduction

Obtaining true information from Web data is pivotal to the success of applications in making data-driven decisions. However, the Web data involve high inconsistency and false/out-of-date information. As a matter of fact, even for a simple question such as “Height of Mount Everest”, we get conflicting answers from search engines, such as $29,002$ feet, $29,035$ feet and $29,029$ feet. Essentially, our task is to identify the true values from the false ones. Such task is typically referred to as *data fusion* [4].

Data fusion is inherently challenging - the amount of information available on the Web has been growing rapidly and the Web data are being altered from time to time. Thus data fusion has drawn much attention from researchers. Existing works followed a general principle - they perform a weighted aggregation of multiple data sources based on the estimated source trustworthiness [7], assuming that the data generated from the same source is equally reliable. Many works have designed methods of estimating source quality and integrating results based on such principle. In addition, there are works trying to model the relationships between sources.

Though there exist many attempts to address the data fusion problem as discussed above, these machine-based approaches cannot achieve high accuracy. Therefore, we would like to leverage the power of the crowd to refine data fusion result obtained from pure machine-based methods. Besides, crowd-only system is too expensive to afford. With help of computer, we can leverage crowd power efficiently to refine the data fusion result.

In this paper, we develop a novel crowdsourcing-based machine-crowd hybrid system, namely *CrowdFusion*, to improve result of exiting *data fusion* methods. Figure 1 shows the general work flow of our system. Compatible with the traditional data fusion models, the input of our system is a set of fact observations and a prior probability distribution over all possible results, i.e., probability distribution calculated by existing data fusion models. There are two outstanding features of CrowdFusion - (1) it does not hold any strong assumption as the existing data fusion methods do; and (2) CrowdFusion is able to make use of the inherent correlations among facts to work efficiently. Since crowd workers are not always able to give correct answers especially when the tasks are complex, we take judgment of one fact as our task to get higher accuracy. The challenge is how we can ask the crowd tasks efficiently to get more accurate results with a restricted budget - we formulate this challenge as an optimization problem. Furthermore, we show the computational hardness of the problem, and design an approximate solution. Besides, we extend the problem to a new scenario with a query given by users as an extension and propose a task selection method to ask crowds effectively under such condition.

To summarize, we have made the following contributions. In Section II, we formally define crowdsourcing-based data fusion problem, which is the first attempt to address Web data fusion with the help of the crowd. We design a system called CrowdFusion to handle data fusion problems, prove that the finding the optimal set of crowdsourcing tasks is NP-hard, and consequently propose several approximation solutions in Section III. Finally, we conduct an extensive experimental study on a real crwodsourcing platform to demonstrate the effectiveneess of CrowdFusion. The results are discussed in Section IV.

II. Background and Problem Definitions

In this section, we briefly introduce the data model and the crowdsourcing model. We will give the formal definition of the problem that we are going to address in this work.

Let $\mathcal{F}$ be a set of facts and $|\mathcal{F}| = n$. A fact f_i is represented as a triple of {subject, predicate, object} and its value is either true or false. It is quite possible that multiple facts with the same subject and the same predicate are true; e.g. fact {Barack

2375-026X/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDE.2017.56

Obama, Daughter, Malia Obama} and fact {Barack Obama, Daughter, Sasha Obama} are both true facts. Given n facts, we consider each fact as a Bernoulli random variable, and the dependencies among the facts can be naturally depicted as their joint distribution, which has 2^n possible outputs. We represent probabilities of all possible outputs by $P(o_i), i = 1, 2, ..., 2^n$. Output Set $\mathcal{O}$ consists of all possible outputs.

One output o_i is a set of true-or-false judgments, i.e. $o_i = \{(f_i, state)|i = 1, ..., n, state \in \{true, false\}\}$. Let O_k be a set of outputs having f_k true, that is $O_k = \{o_i|(f_k, true) \in o_i\}$.

Definition 1: Given a fact set $\mathcal{F}$, the estimation of quality of $\mathcal{F}$, denoted by $Q(\mathcal{F})$ as an utility function, is the negative value of Shannon Entropy, that is

$$Q(\mathcal{F}) = -H(\mathcal{F}) = \sum_{i=1}^{2^n} P(o_i) \log P(o_i),$$

where $\sum_{i=1}^{2^n} P(o_i) = 1$.

We ask the crowd whether each fact is true or false independently to keep relatively high crowd reliability and set the probability of correctness of a worker to no less than 0.5. We will take crowd result as a sample of Bernoulli distribution.

Definition 2: Given a crowd, the probability that answer given by the crowd is correct is $P_c \in [0.5, 1]$. We assume all the tasks completed by crowds are independent from each other, i.e. whether we get correct answer for task t_i does not affect whether we can get correct answer for task t_j as long as $i \neq j$. We define entropy of crowd $H(Crowd)$ as

$$H(Crowd) = -P_c \log (P_c) - (1 - P_c) \log (1 - P_c). \quad (1)$$

After defining the data model and the crowdsourcing model, we can now formally define our system goal of CrowdFusion.

Definition 3: Given a fact set $\mathcal{F}$, possible outputs $\mathcal{O}$ with probability joint distribution and a crowd with accuracy P_c, our goal is to maximize the utility $Q(\mathcal{T})$ by selecting a size-k set of facts to ask the crowd.

III. CrowdFusion

The *CrowdFusion* system architecture is demonstrated in Figure 1. Explicitly, CrowdFusion can be initialized by any existing probability-based data fusion method, or simply set to uniform distribution. We call a selection-collection-updating cycle as a *round* in CrowdFusion. After that, the system executes the data improvement process for multiple rounds. In each round, we select a set of tasks, publish them to a crowd, and then use the crowdsourced answers to improve the data quality. The whole procedure terminates when the budget runs out, and generates the fusion results as output.

First, we concentrate on how the crowd answers will affect the confidence of the outputs, or say how to update $P(\mathcal{O})$ after we receive answers $Ans_j^{\mathcal{T}}$ from crowds. Considering a specific output o_i, the probability of o_i to be true is updated to $P(o_i|Ans_j^{\mathcal{T}})$. Then we can modify the probability of o_i to

$$P(o_i|Ans_j^{\mathcal{T}}) = P(o_i)P(Ans_j^{\mathcal{T}}|o_i)/P(Ans_j^{\mathcal{T}}). \quad (2)$$

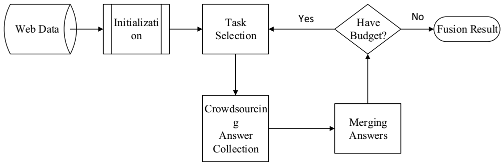


Fig. 1. System Flowchart

Definition 4: Given a task set $\mathcal{T}$, the utility of data after asking is:

$$Q(\mathcal{F}|\mathcal{T}) = H(\mathcal{T}) - H(\mathcal{F}, \mathcal{T}),$$

where $\mathcal{T}$ stands for task set $\mathcal{T}$ is selected to be asked.

To maximize $Q(\mathcal{F}|\mathcal{T})$ is as same as to maximize $\Delta Q(\mathcal{F}) = Q(\mathcal{F}|\mathcal{T}) - Q(\mathcal{F}) = H(\mathcal{F}|\mathcal{T}) - H(\mathcal{F})$. By properties of entropy, we have:

$$\Delta Q(\mathcal{F}) = H(\mathcal{F}) - H(\mathcal{F}|\mathcal{T}) = H(\mathcal{T}) - H(\mathcal{T}|\mathcal{F})$$

Where $H(\mathcal{T}|\mathcal{F}) = kH(Crowd)$ which is a constant given k. Thus, the task is simplified to select k facts tasks set $\mathcal{T}$ to maximize $H(\mathcal{T})$. And $H(\mathcal{T}) = H(\{Ans_i^{\mathcal{T}}\})$. Formally, we have the following optimization goal:

$$\mathcal{T}_{best} := arg \max_{\mathcal{T}} H(\mathcal{T}) \quad (3)$$

We have proved that given n possible tasks, selecting k of them to reach the highest value of utility function is an NP-hard problem. However, it is known that the conditional entropy is a submodular function [5], and the problem of selecting a k-element subset maximizing a monotone submodular function can be approximated with a performance guarantee of $(1 - 1/e)$, by iteratively selecting the most uncertain variable given the ones selected so far. Similarly, we can select k tasks iteratively with a modified greedy algorithm to get a $(1 - \frac{1}{e})$-approximate solution.

We select tasks based on the pre-defined k by greedy algorithm. We have proved that utililty will be improved whenever an uncertain fact exists to be selected to ask. Such that, in each round, we select as many tasks as we could to achieve better performance.

By considering the highest possible total gain value after selecting a fact in each iteration, we can prune some facts and do not need to consider them any more in the following iterations of the approximation solution.

Pruning f_j for all following selections is safe if $H(\mathcal{T} \cup \{f_j\}) + \log (k - |\mathcal{T}| - 1) < \max_t H(\mathcal{T} \cup \{f_t\})$.

Besides, we also designed a preprocessing to significantly accelerate the algorithm. For any of choice, we have $O(2^k)$ items in the marginal distribution and for each item we need $O(k|\mathcal{O}|)$ to calculate the probability as we need time k for counting the number of the same and the different judgments between the output items with the answer. Thus, we need $O(2^k nk|\mathcal{O}|)$ time for a single step which is a huge cost as it will repeat k times and finally cost $O(2^k nk^2|\mathcal{O}|)$ in total. The preprocessing is the calculation of Answer Joint Distribution. For details, please refer to our technical report [1].

TABLE I. ONE ROUND AVERAGE RUNNING TIMES OF FIVE APPROACHES

k	*OPT*	Approx.	Approx.&Prune	Approx.&Pre.	Approx.&Prune&Pre.
1	37.78	32.60	33.44	**1.08**	1.15
2	1475.66	94.73	40.94	2.10	**1.34**
3	75359.26	242.22	56.66	3.10	**1.40**
4		598.15	74.93	4.08	**1.53**
5		1401.05	74.32	4.96	**1.54**
6		3230.22	76.09	5.86	**1.48**
7		7005.02	74.35	6.67	**1.63**
8		14611.53	74.88	7.80	**1.53**
9		29476.42	74.87	8.37	**1.59**
10		57198.67	74.34	9.39	**1.82**

Note: Data listed in this table are time costs for each conditions in second.

IV. EXPERIMENTAL EVALUATION

We have conducted extensive experiments to evaluate our proposals with real-world datasets on gMission [2], which is a public crowdsourcing platform. We focus on investigating three issues. First, we examine the efficiency of our techniques. Second, we verify the effectiveness of our approaches, by evaluating the utility and F_1-score.

Crowdsourcing Platform: We conduct our experiments on *gMission* - a real crowdsourcing platform. [1]

Dataset: We adopt the *Book* [2] dataset, which is widely used in the field of data fusion [8], [3]. This dataset is particularly appropriate for crowd workers as it is about general real world knowledge and information that can be easily obtained manually. We manually label the ground truth for all items related to the *Gold standard* provided by the dataset for the purpose of evaluation.

As the data are statements of books' author list but not author of books, we define our facts triple as $\{$book, complete full name author list, statement$\}$ rather than $\{$book, author, statement$\}$.

Since our CrowdFusion system is general enough to be initialized by any "machine-only" fusion model with probabilistic results, we adopt the modified *CRH framework*[6] for initialization in this experiment.

We treat information about each book independently and set a budget of $B = 60$ tasks for each book. In each round, we set the number of tasks to k, so there are $\lceil B/k \rceil$ rounds. If a book has $n \geq k$ facts, we will ask k tasks in every round except for the last one. Otherwise, we will ask n tasks in each round instead.

Operating Environment: The experiments are run on a 10-nodes Linux cluster. Each note contains 4×10-core Intel Xeon E5-2650v3 (2.3 GHz) processors and 196 GB Physical memory. The Linux distribution installed is CentOS 6, x86_64 edition. For each condition, the programs are run for three times to get an average time cost on a single node of the cluster.

A. Efficiency Evaluation

In this subsection, we show the time cost of the following competing algorithms: (1) **OPT**: selecting exact optimal algorithm by brute-force method; (2) **Approx.**: the approximation algorithm (3) **Approx.&Prune** the approximation algorithm with the pruning; (4) **Approx.&Pre.** the approximation algorithm with the preprocessing strategy and (5) **Approx.&Prune&Pre**: the approximation algorithm with both the pruning and the preprocessing strategies. Please note that the books with a small number of facts usually stop getting better early and cannot show the efficiency clearly. Therefore, in order to distinguish the performances of above algorithms, we focus on books with facts more than 20. We test average running time for the books in one round. The details of the experimental results are demonstrated in Table I.

The time cost of *OPT* method increases exponentially, which is not affordable in real application. With $k = 4$, we had been waiting for more than 5 days and the algorithm was still running. From the experimental results, one can see that the pruning strategy is powerful - the time cost is almost constant w.r.t. the increase of k, no matter we adopt the preprocessing or not. Even though approximation method is a linear algorithm, the time cost increases rapidly and does not scale well. With the preprocessed data, we can significantly decrease the time consumption for task selection. Please note that we need to do the preprocessing for each round of selection. The approximation method with preprocessing is still a linear algorithm, but becomes much faster than that without preprocessing.

B. Quality Evaluation on gMission

We verify the correctness of our CrowdFusion system, by evaluating the accuracy of proposed algorithms with different settings of k and P_c.

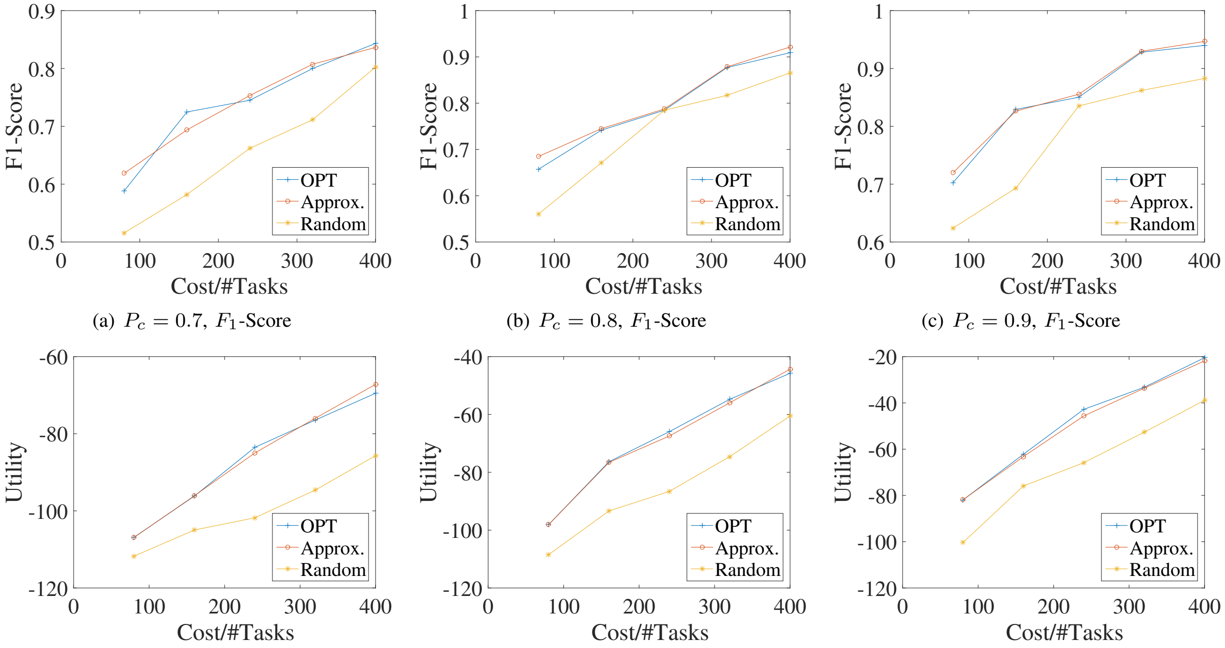
We use two different measurements to evaluate quality of our method. The first one is the utility defined in section II, which is our optimization goal. This measurement indicates how well our algorithm approximates the optimization goal. We simply sum up the utility scores of all data instances for the evaluation. The second measurement is F_1 score, which is calculated based on the ground-truth labels.

Budget is an essential issue in our experiment, thus we focus on the quality improvement with the increase of budget. Our budget allows CrowdFusion to ask at most 60 tasks for each book and we have 100 books in total.

Due to that the *OPT* solution is not scalable, we need to scale down k, B and n to conduct comparison with the *OPT* solution. Please note that *OPT* with $k = 1$ would have the result exactly the same as that given by the approximation algorithms - they all select the very best task at each round. We compare performance of *OPT* solution and our greedy solution only in condition that $k = 2$ and budget $B = 10$ with

[1] http://gmission.github.io/

[2] http://lunadong.com/fusionDataSets.htm



(a) $P_c = 0.7$, F_1-Score (b) $P_c = 0.8$, F_1-Score (c) $P_c = 0.9$, F_1-Score

(d) $P_c = 0.7$, Utility (e) $P_c = 0.8$, Utility (f) $P_c = 0.9$, Utility

Fig. 2. Quality Improvement of Three Methods, Cost as Number of Total Tasks

a small subset of data with 40 books, which contains the least number of statements n in the whole data set.

Figure 2 shows comparing between *OPT* results, approximation algorithm results and a random selection results in both F_1-Score measurement and utility measurement. Our approximation strategy performs as good as the *OPT* method not only in reaching optimization target but also in getting accurate results. And our approximation algorithm is significantly better than random selecting method. Additionally, *OPT* results are not always better than the approximated results, which is because that the crowdsourced answer can be incorrect and hereby reduce the utility and F1-score. In other words, the quality is not absolute monotonic w.r.t the number of crowd sourced answers recieved.

We also compared different k and P_c settings for our algorithm and further analyzed the errors. Please refer to our technical report [1].

V. Conclusions

In this paper, a crowdsourced data fusion refinement method is proposed and utilized to improve data fusion results of existing machine-only methods. Since different tasks may lead to different amount of benefit, we design an approximate algorithm with pruning and preprocessing strategies for this task selection problem which is NP-hard if we want exact the best task set. Empirical study shows that CrowdFusion achieves high accuracy at finding true facts and at the same time computational cost can be reduced by the approximation algorithm or heuristic solution without losing much effectiveness.

Our work is just an initial solution for data fusion with crowdsourcing technique. Further research on measuring the relationships between facts is a direction, which may be related with natural language processing, image processing and audio processing.

Acknowledgment

The work is supported partially by Hong Kong RGC Project N HKUST637/13, NSFC Guang Dong Grant No. U1301253, National Grand Fundamental Research 973 Program of China under Grant 2014CB340303, Microsoft Research Asia Gift Grant, the Fostering Project of Dominant Discipline and Talent Team of Shandong Province Higher Education Institutions.

References

[1] Y. Chen, L. Chen, and C. J. Zhang. Crowdfusion: A crowdsourced approach on data fusion refinement. *arXiv:1702.00567*.

[2] Z. Chen, R. Fu, Z. Zhao, Z. Liu, L. Xia, L. Chen, P. Cheng, C. C. Cao, Y. Tong, and C. J. Zhang. gmission: a general spatial crowdsourcing platform. *Proceedings of the VLDB Endowment*, 7(13):1629–1632, 2014.

[3] X. L. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: the role of source dependence. *Proceedings of the VLDB Endowment*, 2(1):550–561, 2009.

[4] X. L. Dong, E. Gabrilovich, G. Heitz, W. Horn, K. Murphy, S. Sun, and W. Zhang. From data fusion to knowledge fusion. *Proceedings of the VLDB Endowment*, 7(10):881–892, 2014.

[5] A. Krause and C. Guestrin. A note on the budgeted maximization of submodular functions. 2005.

[6] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 1187–1198. ACM, 2014.

[7] Y. Li, J. Gao, C. Meng, Q. Li, L. Su, B. Zhao, W. Fan, and J. Han. A survey on truth discovery. *arXiv preprint arXiv:1505.02463*, 2015.

[8] X. Yin, J. Han, and P. S. Yu. Truth discovery with multiple conflicting information providers on the web. *Knowledge and Data Engineering, IEEE Transactions on*, 20(6):796–808, 2008.