This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2954087, IEEE Transactions on Knowledge and Data Engineering

Improving the Quality of Web-based Data Imputation with Crowd Intervention

Binbin Gu, Zhixu Li, An Liu, Jiajie Xu, Lei Zhao and Xiaofang Zhou *Fellow, IEEE*

Abstract—Data incompleteness is a common data quality problem in databases. Recent work proposes to retrieve missing string values from the World Wide Web for higher imputation recall, but on the other hand, takes the risk of introducing web noises into the imputation results. So far there lacks an effective way to control the quality of web-based data imputation, given the complexity of the quality model and lacking of enough ground truth data. In this paper, an EM-based quality model is firstly built for web-based data imputation which investigates three key factors jointly, i.e., precision of web sources, correlation among web sources, and precision and recall of the employed extractors. However, the accuracy of the EM-based quality model could be harmed when the EM (Expectation Maximization) assumption that "the majority agree on the truth" does not hold in some cases. To solve this problem, we introduce crowd intervention to help improve the quality model. While a straightforward but expensive way is to let the crowd to identify all these undesirable cases and provide the right imputation values for these blanks, a most crowd-economic way is to select a small set of blanks for crowd-based imputation, whose results could help to adjust the EM-based quality model towards a better one. To achieve this, an adaptive blank selection strategy is proposed to select a sequence of blanks for crowd-based imputation. Also, we work on finding a proper time to stop further crowd intervention for the balance of crowd efficiency and quality improvement. Our experiments performed on three real world and one simulated data collections prove that the proposed quality model can effectively help improve the quality of the web-based imputation results by more than 15%, while our crowd cost saving strategy saves more than 75% crowd cost.

Index Terms—Data Imputation, Web, Crowd

1 Introduction

Missing data is a common issue in almost every large data collection, and the process of filling in the missing values is well known as data imputation [25], [39]. Traditional imputation methods to non-quantitive string data mainly rely on some local data constraints (such as FD/CFDs) [1], [34] or prediction models [39], [46] to infer substitutes or estimations for the missing string values. However, these methods always fail to get the right missing values due to the limitation of relevant information and knowledge for filling in these missing values. To reach higher imputation precision and recall, some recent work turn to outsource the task to crowd workers [14], [45] when the traditional methods are not capable of filling the missing values. However, crowd-based imputation could be expensive given that it pays money for every human input.

A rising class of approaches propose to get missing string values from external sources such as online encyclopedia [42] and the world wide web [25], [26], [19]. While some work proposes to get the required values from web tables and web lists [19], a more recent work proposes a web-based data imputation framework called *WebPut*, which finds the missing values with indicative context information from free-text web pages on the web. It leverages traditional information extraction methods together with the capabilities of web search engines towards the goal of imputing missing values in relational tables [25], [26]. The primary two steps of WebPut are issuing queries to the web and extracting the missing values from the returned web pages of various web sources as described in Figure 1.

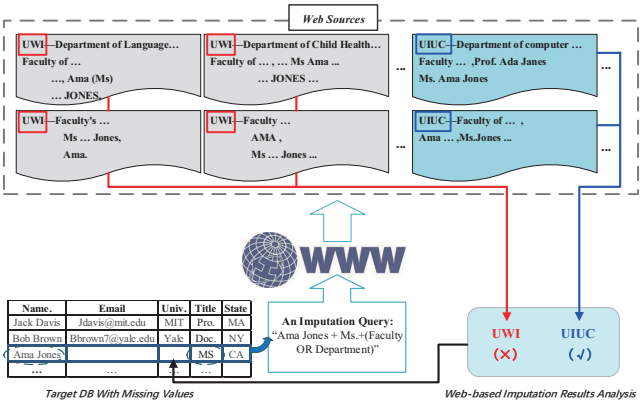


Fig. 1. A Bad Example of WebPut

Although reaching a much higher imputation recall than the previous approaches, due to various reasons such as the noisy data on retrieved web pages, or incorrect extraction of the target value from the web pages, WebPut is very likely to introduce incorrect imputation results into the objective database. Consider an imputation case described

- *B. Gu, Z. Li, A. Liu, J. Xu, L. Zhao are with the Institute of Artificial Intelligence, School of Computer Science and Technology, Soochow University, China. Email: gu.binbin@hotmail.com, {zhixuli, anliu, xujj, zhaol}@suda.edu.cn. Z. Li is the corresponding author of the paper.*
- *Z. Li is also with the IFLYTEK Research, Suzhou, China, and the State Key Laboratory of Cognitive Intelligence, IFLYTEK, Hefei, China.*
- *X. Zhou is with the School of Information Technology and Electrical Engineering, The University of Queensland, Brisbane QLD 4072, Australia. He is also an Adjunct Professor with the School of Computer Science and Technology, Soochow University, China. Email: zxf@itee.uq.edu.au*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2954087, IEEE Transactions on Knowledge and Data Engineering

in Figure 1, where a wrong answer (v_1) to a blank is supported by 4 web sources while the correct one (v_2) is only supported by 2. If we simply count on the number of websites supporting an answer, the wrong answer v_1 will be filled into the blank as a right one. According to our observations, the possible reasons behind this phenomenon including: low accuracy of the answers provided by sources, the plagiarism of the content between sources, and the failure of the extraction ways in identifying the target values from the retrieved web pages.

To control the quality of the web-based data imputation results for WebPut, building a suitable quality model is essential. According to our analysis above, this quality model should consider at least the following three groups of factors: (1) *Precision of Every Web Source*, which reflects the average correct likelihood of the answers provided by each web source; (2) *Correlation among Web Sources* is non-trivial in deciding the final answer to a blank provided by WebPut, given that sources may not be independent since one source may copy data from the other sources; (3) *Precision and Recall of Every Extractor* also greatly affects the quality of each extracted answer from the retrieved web pages, given that extractors may also make mistakes in extracting the correct answers from the sources. Note that the three groups of factors do not work independently, but mutually determine our judgement to the correctness of an imputation value, which brings great challenges to the construction of the quality model.

This paper mainly works on building a proper quality model for WebPut. Particularly, this quality model would consider the three factors above jointly in deciding the correctness of every retrieved imputation value from the Web. To this end, three groups of variables corresponding to the three factors respectively need to be set up for building the model. Nonetheless, given no priori knowledge about the correct imputation answers to all the blanks in hand, it is difficult to tackle the correlation among Web sources. As an alternative, we initially propose to build the quality model by employing an unsupervised EM-based approach to set up proper values for the three groups of variables, based on the simple EM (Expectation Maximization) assumption [28] that "the majority agree on the truth". However, the accuracy of the EM-based quality model could be harmed when the EM assumption that "the majority agree on the truth" does not hold in some cases. To solve this problem, we introduce crowd intervention to help improve the quality model later.

Given the complexity of the three factors' relations in jointly deciding the quality of the extracted values, estimating proper values for the three groups of variables at once is intractable. As an alternative, we first estimate the precision and recall of each extraction way with EM by fixing the precision of sources and the correlation among web sources, and then estimate the precision of each source by fixing the correlation among web sources. Finally, we discuss on setting up the correlation between web sources. We repeat this value estimation process for variables iteratively until the values of these variables become stable.

Most of the time, the EM-based quality model could help get the right answers for blanks in the objective database, but it still makes mistakes when its assumption does not hold in some specific cases. For instance, the ubiquitous copy operations among some sources will make the incorrect answer provided by these sources have a pretty high confidence to be taken as a correct answer by WebPut. Also, the EM-based quality model may be paralyzed for some blanks where all the candidate values have small correctness probabilities. As no ground truth is available, our inference to the three groups of variables in the quality model greatly depend on the EM assumption. As a result, when the assumption does not fit the actual case, the EM-based quality model may not be able to reach a good performance.

To tackle the problems, we introduce crowd interventions to help improve the EM-based quality model. In other words, we translate the unsupervised model into the semi-supervised model with some crowd interventions. A baseline approach could ask the crowd to help solve every problematic case by providing the right imputation results, but apparently this baseline way would be very costly when the data set is large. As an alternative, we novelly propose to do active crowd intervention to the EM-based quality model. The purpose of active crowd intervention is to adjust the EM-based model with as less blanks for crowd-based imputation as possible. To achieve this, a set of "informative" and "diverse" blanks are expected to be selected into a sampling set for crowd intervention, where "informative" means having a small correctness probability of the imputation result in WebPut while "diverse" blanks can help to improve the precision and correlations of all the sources of the EM-based quality model. Based on the crowd intervention results on the sampling set, we then work on adjusting the precision of web sources as well as the correlation among sources. In addition, to reach a balance between accuracy and crowd efficiency, we also need to figure out when to stop further crowd intervention.

To summarize, our contributions are as follows:

- We propose to build a quality model to control the quality of web-based data imputation results, which takes several important factors into account including the precision of web sources, the correlation among web sources, and the precision and recall of the extractors.
- Given no priori knowledge on the right imputation answers to those blanks, we propose to employ an iterative EM approach to estimate proper values for the three groups of factor variables in the quality model.
- To address the problems of the EM-based quality model, we introduce crowd intervention to help improve the model. Besides, we find ways to minimize the crowd cost as much as possible while guaranteeing the quality of the model. And our method is based on a general unsupervised method and can be easily extended to other similar models (Probabilistic Models).

Experiments : Our experimental study conducted on sev-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2954087, IEEE Transactions on Knowledge and Data Engineering

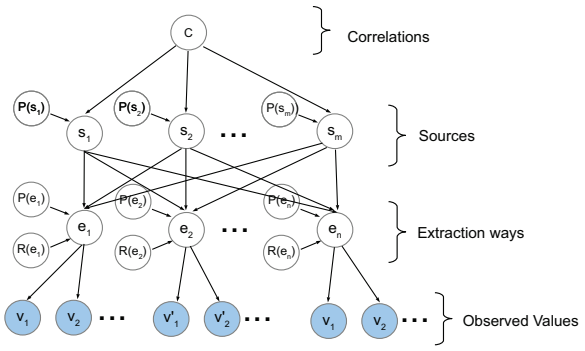


Fig. 2. The Key Factors in the Quality Model for WebPut

eral real data collections proves that the EM-based quality model improves the imputation quality by 10%, while the crowd intervention further enhances the quality by 5%. Besides, our proposed method could save 75+% crowd cost without hurting the imputation quality.

RoadMap : The rest of the paper is organized as follows: We first state our problem in Sec. 2. We introduce our EM-based probabilistic quality model for WebPut in Sec. 3, and then present how we modify the EM-based probabilistic model with crowd intervention in Sec. 4. After reporting the experiments in Sec. 5, the related work is covered in Sec. 6. We conclude in Sec. 7.

2 Problem Statement

Given a database with missing values, the Web-based Data Imputation (WebPut for short) approach issues imputation queries to web search engines for every blank in the database, with an expectation to obtain the right missing value from relevant web sources for the blank. More detailed description to WebPut could be found in the literature [26]. However, given several factors that may interfere the accuracy of the web-based data imputation results, it is necessary to establish a proper quality model for controlling the quality of WebPut results.

Key Quality Factors. To establish a proper quality model, three groups of key quality factors need to be considered jointly in deciding the correctness of imputation results as depicted in Fig. 2:

- Precision of Each Web Source: The precision of a web source is the average correct likelihood of the answers provided by the web source. It is the most direct factor that need to be considered in the quality model.
- Correlation among Web Sources: The correlation among web sources is non-negligible. If multiple web sources with correlation provide the same wrong answer to a blank, the wrong answer would have a much larger opportunity to be taken as the final answer to the blank by WebPut if we do not consider their correlation.
- Precision and Recall of Each Extractor: The employed extractors may also bring noises into the imputation results when they failed to identify the right answer from the retrieved web pages by WebPut. Please also note that an extractor is formulated according to a certain pattern for extracting *(subject, predicate, object)* triples from a web source. For instance, one extractor can extract the triple $(\$A, country, \$B)$ according to the pattern "A, the president of B". However, this is not correct when A is the president of a university, not a country. In this case the pattern "the nationality of A, B" may be more effective. So we need more than one extractors to guarantee the high accuracy of WebPut.

Basic Assumptions. Several basic assumptions are given below: (1) Although different web-based data imputation query may get different numbers of search results from the Web, only the top-k search results of a query are employed for imputation, where k = 100 is a proper value as reported in the literature [26]; (2) since different kinds of missing values usually require different extractors to extract the values from the web pages and the chosen process is usually made by human experts, thus the dependencies between different extractors can be negligible;

Notations. Some notations used in this paper are in Table 1.

NOTATION	DESCRIPTION
$s \in S$	A web source s in the web source set S
$e \in E$	An extraction way e in all the employed extraction ways E
$b \in B$	A blank b in the set of all blanks B in the objective database
$v \in V^b$	A value v in a set of candidate imputation values V^b to blank b
v_c^b	The correct imputation value for blank b
$v^b(s)$	An imputation value provided by a source s to b
$V^B(s)$	An imputation value set provided by s for B
$x \rightarrow y$	A source (or/and) extraction way x returns a value y

TABLE 1
Notations Used in the Rest of The Paper

Given all the above, the quality control task for web-based data imputation can be informally stated as follows:

Problem Statement. Given an objective table T with a set of absent values (or blanks) B, assume WebPut retrieves missing values for blanks in B from a set of web sources $S = \{s_1, s_2, ..., s_m\}$, by using a set of employed extraction ways in $E = \{e_1, e_2, ..., e_n\}$ to detect all kinds of objective missing values from the web pages in S, the quality control task for WebPut is to build a proper quality model below:

$$Model_Q(S, E) = f_Q(Corr(S), \{P(s_1), P(s_2), ...P(s_m)\}, \{P(e_1), P(e_2), ...P(e_n)\}, \{R(e_1), R(e_2), ...R(e_n)\})$$

where $f_Q(\cdot)$ denotes a function which could predict the quality of each candidate value extracted from the web sources and $Corr(S)$ is the correlation among sources in S, $P(s_i)$ $(1 \le i \le m)$ denotes the precision of a web source s_i, and $P(e_j)$ and $R(e_j)$ $(1 \le j \le n)$ are the precision and recall of an employed extractor e_j respectively. With this quality model, we expect to help identify the correct imputation values from the incorrect ones effectively, such that we can improve the quality of the web-based imputation results.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2954087, IEEE Transactions on Knowledge and Data Engineering

3 EM-BASED QUALITY MODELING

In this section, we introduce an EM-based probabilistic quality model for WebPut, which considers the three groups of factors presented in last section jointly. Given no priori knowledge on the right imputation values to any of these blanks, this EM-based quality model tries to estimate proper values for the three groups of variables (corresponding to the factors) based on the basic EM assumption that "the majority agree on the truth".

However, directly analyzing the three groups of variables together with EM approach is intractable. As an alternative, we first present how we estimate the correct likelihood of each value by assuming that all web sources are independent in Sec. 3.1, and then introduce how we do it with source correlation in Sec. 3.2. We finally analyze the drawback of the model in Sec. 3.3.

3.1 Inferring with Independent Sources

By temporarily ignoring the dependencies between sources, we can consider a source s and an extraction way e in combination, i.e., (s,e), in providing the correct imputation value for a specific blank b as follows:

$$Pr((s,e) \rightarrow v|b) = \begin{cases} P(s,e) & \text{if } v_c^b = v \\ \lambda(s,e,b,v) \cdot (1 - P(s,e)) & \text{otherwise} \end{cases} \quad (1)$$

where $P(s,e)$ is the precision of the (s, e) combination, v_c^b is the correct imputation value for the blank b, $\lambda(s,e,b,v)$ is the probability that (s,e) provides a certain wrong value v for b among all the wrong values (we adopt the uniform distribution to these wrong values in our experiments).

However, we do not know a priori knowledge about which value is the right answer which makes Eq. 1 incalculable. In this case, we employ an iterative EM [28] approach to estimate the precision of each source-extraction combination, i.e., $P(s,e)$ for each (s,e), such that we can further calculate the correct probability of each imputation value v returned by any of these source-extraction combinations.

Let $\vec{V}^b = [v_{1,1}, ..., v_{i,j}, ..., v_{m,n}]$ denote the vector of the observed values returned for imputing the blank b from all the combinations of source-extractor pairs, where $v_{i,j}$ is the value returned by the source-extractor pair (s_i, e_j) $(1 \le i \le m,\ 1 \le j \le n)$, then we have the joint distribution of the observed imputation values for b as follows:

$$Pr(\vec{V}^b|v_c^b = v, b) = \prod_{(s_i,e_j) \in S \times E} Pr((s_i,e_j) \rightarrow v_{i,j}|v_c^b = v, b) \quad (2)$$

Basically, the object of the EM approach is to find out the v that can maximize the value of $Pr(\vec{V}^b|v_c^b = v, b)$. Given that we have no prior knowledge about the confidence of each source, we give them a same score (say 0.7 in our experiments) initially. Specifically, we do E-step and M-step as follows:

E-step: At an E-step, given the observation value vector $\vec{V}^b$ for blank b, we can calculate the probability of $v_c^b = v$ by the Bayesian rule as follows:

$$Pr(v_c^b = v|\vec{V}^b, b) = \frac{Pr(v_c^b = v, b) \cdot Pr(\vec{V}^b|v_c^b = v, b)}{\sum_{v' \in \{\vec{V}^b\}} Pr(v_c^b = v', b) \cdot Pr(\vec{V}^b|v_c^b = v', b)} \quad (3)$$

where $\{\vec{V}^b\}$ is the set of all distinct values in $\vec{V}^b$. Note that at the first E-step, we use a uniform prior for all the $Pr(v_c^b, b)$ since we do not know any prior knowledge of the true values in the beginning.

M-step: At a M-step, we update v_c^b with the value v that can maximize $Pr(v_c^b = v|\vec{V}^b, b)$ among all the observed values, i.e.,

$$\hat{v_c^b} = arg\max Pr(v_c^b = v|\vec{V}^b, b), \quad (4)$$

and then we can update all $P(s,e)$ as follows:

$$P(s,e) = \frac{\sum_{((s,e) \rightarrow v_c^b | b \in B)} Pr(v_c^b = v|\vec{V}^b, b)}{\sum_{((s,e) \rightarrow v_c^b | b \in B)} 1} \quad (5)$$

which means we estimate the precision of (s,e) with the probability of (s,e) in providing the correct answers for blanks in B.

We perform the E-step and M-step alternately until all $P(s,e)$ become stable.

3.2 Inferring with Source Correlation

Given the correlation among sources, we need to consider every source and its corresponding extraction ways independently (instead of the two in combination). With the correlation among sources, we rewrite Eq. 1 as follows:

$$Pr((s,e) \rightarrow v|b) = \begin{cases} P(s) \cdot R(e) & True(b) = v \\ \lambda(s,e,b,v)P(s)(1- & \\ P(e)) + \lambda(s,e,b,v) & \\ R(e)(1 - P(s)) & \text{otherwise} \end{cases} \quad (6)$$

where $P(s)$ is the precision of the the source s, $P(e)$ is the precision of the extractor e, and $R(e)$ is the recall of the extractor e.

As illustrated in Fig. 2, three different kinds of variables need to be analyzed in this model. However, directly analyzing the three variables together is intractable because we do not know whether a web source actually provides a value v for a blank b and which is the true value v_c^b for a blank b. In other words, there are two groups of latent variables. Therefore, we first estimate the precision $P(e)$ and recall $R(e)$ of each extraction way e with the EM approach by fixing $P(s)$, and then estimate the precision $P(s)$ of each source s.

1) Estimating $P(e)$ and $R(e)$: Denote $\vec{V}^b(s) = [v_1, ..., v_j, ..., v_n]$ the vector of observed imputation values from s returned by the n extractors for the blank b, where v_j is the value returned by e_j $(1 \le j \le n)$. Based on the assumption that all the extractors are independent, we may have the joint distribution similar to Eq. 2 as follows:

$$Pr(\vec{V}^b(s)|v^b(s) = v, b) = \prod_{e_j \in E} Pr((s,e_j) \rightarrow v_j|v^b(s) = v, b) \quad (7)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2954087, IEEE Transactions on Knowledge and Data Engineering

where $v^b(s)$ is the value that truly provided by the source s for b (although it may not the correct imputation value for b).

Basically, the object of the EM approach here is to find out the v that can maximize the value of $Pr(\vec{V}^b(s)|v^b(s) = v, b)$. Specifically, we do E-step and M-step as follows:

E-step: At an E-step, given the observed value vector $\vec{V}^b(s)$ for blank b, we can calculate the probability of $v_c^b(s) = v$ by the Bayesian rule as follows:

$$Pr(v^b(s) = v|\vec{V}^b(s), b) = \frac{Pr(v^b(s) = v, b) \cdot Pr(\vec{V}^b(s)|v^b(s) = v, b)}{\sum_{v' \in \{\vec{V}^s\}} Pr(v^b(s) = v', b) \cdot Pr(\vec{V}^b(s)|v^b(s) = v', b)} \quad (8)$$

where $\{\vec{V}^b(s)\}$ is the set of all distinct values in $\vec{V}^b(s)$. Note that at the first E-step, we use a uniform prior for all the $Pr(v^b(s), b)$ since we do not know any prior knowledge of the true values in the beginning.

M-step: At a M-step, we update $v_c^b(s)$ with the value v that can maximize $Pr(v_c^b(s) = v|V^b(s), b)$ among all the observed values, i.e.,

$$\hat{v^b(s)} = arg\max Pr(v^b(s) = v|\vec{V}^b(s), b),$$

and then we can update all $P(e)$ and $R(e)$ as follows:

$$P(e) = \frac{\sum_{((e,s) \to v_c^b(s)|s \in S, e \in E, b \in B)} Pr(v^b(s) = v|\vec{V}^b(s), b)}{\sum_{((e,s) \to v_c^b(s)|s \in S, e \in E, b \in B)} 1} \quad (9)$$

$$R(e) = \frac{\sum_{((e,s) \to v_c^b(s)|s \in S, e \in E, b \in B)} Pr(v^b(s) = v|\vec{V}^b(s), b)}{\sum_{v' \in \{\vec{V}^b(s)\}} Pr(v^b(s) = v'|\vec{V}^b(s), b)} \quad (10)$$

where the meaning of Eq. 9 is similar to Eq. 5, and Eq. 10 calculates the recall of e according to the percentage of correct values among all the values provided by the sources in S.

We perform the E-step and M-step alternately until all $P(e)$ and $R(e)$ become stable.

2) Estimating $P(s)$: After deciding the precision and recall (i.e., $P(e)$ and $R(e)$) for all extractors, we now estimate the precision (i.e., $P(s)$) of all web sources with the correlation among these sources. In statistics, the *inclusive-exclusive principle* [3] is usually applied to calculate the joint probability, but it often needs much prior knowledge (correlations among various incidents). In our case, we actually seek to find the joint true and false probability of values. Inspired by this, we use the joint distribution of values to represent correlation factor of sources which could fit into our EM-model. Specifically, let $Corr(S)$ denote the correlation factor among the sources in S that all provide a correct value. Basically, $Corr(S)$ can be estimated by the joint distribution over the source set S and the precision of each single source in S. Formally,

$$Corr(S) = \frac{P_{joint}(S)}{\prod_{s \in S} P(s)} \quad (11)$$

where $P_{joint}(S)$ is the joint distribution of values provided by S. Note that the correlation factor of the sources in S that all provide an incorrect value is sort of different with the above equation. Instead, the value should be calculated by replacing the denominator as $\prod_{s \in S} P_{inco}(s)$ in Eq. 11 in this circumstance, where $P_{inco}(s)$ is the probability that s provides a specific incorrect value. The challenge here lies on how we compute the joint distribution (accuracy) $P_{joint}(S)$. As we do not have the ground truth about the imputation values for all the blanks, we can not calculate $P_{joint}(S)$ exactly. On the other hand, even having the ground truth, it would be very costly and impractical to estimate the joint distribution for each of the $2^{|S|-1}$ different possible combinations of sources in S.

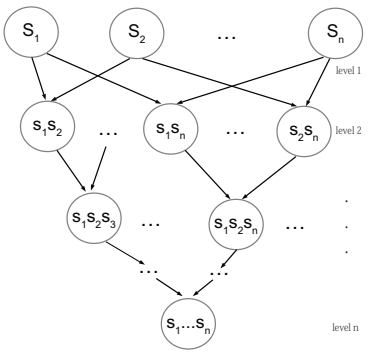


Fig. 3. An Example Lattice of Source Set

As an advisable way, we propose an approximate and efficient way to estimate $P_{joint}(S)$. Inspired by the existence of social communities in social network [12], [13], we suppose that there are also *Source Communities* on the web, where there are relatively strong dependencies within a community, but much less dependencies across communities. Based on this assumption, we would like to get source communities by identifying and merging similar sources that always provide similar values into a community. Approximately but efficiently, we would take these communities as independent ones, such that we could essentially synthesize the dependency between the sources within a source community by giving an expert confidence of them rather than addressing those sources independently. Although this method can not address the correlation of sources thoroughly, we will use the crowd to help us tackle the correlation of sources more effectively and exactly in the next section based on the above strategy.

The key challenge here lies on how we measure the similarity among sources. In the following, we present how we measure the similarity among sources and then introduce how to further reduce the computational complexity of merging sources into communities.

1) Similarity Measure for Sources: Basically, we decide whether two (sets of) sources should be merged or not based on whether the values provided by the two (sets of) sources for the same set of blanks are similar to some extent. Naturally, we can calculate the similarity between two (sets of) sources S_1 and S_2 as follows:

$$sim(S_1, S_2) = \frac{Overlap\{S_1, S_2\}}{|B|} \quad (12)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2954087, IEEE Transactions on Knowledge and Data Engineering

where $Overlap\{S_1, S_2\}$ is the number of overlapping values provided by S_1 and S_2, and B is the blank set which needs to be imputed.

For more than two sources, given a set of blanks B in the local database, we define the proportion of the overlapped values in S w.r.t. B as:

$$closeness(S) = \frac{\sum_{b \in B} D(S,b)}{|B|} \tag{13}$$

where $D(S,b)$ is to distinguish if the values provided by every source in S are the same. Formally, it can be calculated as follows:

$$D(S,b) = \begin{cases} 1 & \text{if } |V(S,b)| = 1 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

where $V(S,b)$ is the set of distinct values provided by all sources in S, and $|\cdot|$ returns the number of elements in a set.

Theorem 1: For two source sets S' and S'', if $closeness(S') < \theta$ and $S' \subset S''$, then $closeness(S'') < \theta$, where $\theta \in (0, 1]$.

Example 1: Assume that $B = \{b_1, b_2, b_3, b_4, b_5\}$ and sources s_1, s_2, s_3, s_4, s_5 provide values for $B = \{b_1, b_2, b_3, b_4, b_5\}$ as $\{a, b, c, d, e\}$, $\{a, b, c, d, f\}$, $\{a, b, c, d, f\}$, $\{a, b, c, d, f\}$, $\{a, g, c, d, e\}$ respectively. Then we have $closeness(s_1, s_3) = 0.8$, $closeness(s_1, s_5) = 0.8$, and $closeness(s_1, s_3, s_5) = 0.6$, $closeness(s_1, s_2, s_3, s_4, s_5) = 0.6$, $closeness(s_2, s_3, s_4) = 1$.

2) Generating Source Communities: We now merge sources into a number of source communities. Besides the similarity measurement for sources, we also need a similarity threshold θ to judge the closeness of all sources in a source set. In other words, a set of sources will be taken as a possible source community only when its closeness is no less than θ.

To merge similar sources, a lattice structure can be used to enumerate the list of all possible source set which is illustrated in Fig. 3. Generally, the $|S|$ sources can potentially generate up to $2^{|S|-1}$ source sets excluding the empty one. Since $|S|$ can be very large, the search space of source sets that need to be explored is exponentially large. To reduce the number of candidate source sets, our strategy is guided by the following intuition. According to Theorem. 1, for two source sets S' and S'', if $closeness(S') < \theta$ and $S' \subset S''$, then S'' should not be a qualified source community, where θ is the minimum closeness threshold we set for source communities. In other word, a qualified source community should guarantee that its members are similar enough to each other. If the source set S is not a candidate source community (whose similarity is less θ), then all the supersets of S can not be candidate source communities. As can be seen from Fig. 3, initially, we make comparisons between any two sources in S and calculate their similarity (the results are illustrated at level two). For those sources whose similarity is less than θ, all its supersets will be pruned from this search space. Next, we iteratively search the candidate source set based on the qualified source set (whose similarity is no less than θ) on current level.

However, comparing the source set between any two sources is quite expensive because a large number of source sets must be examined. We notice that there are many duplicate candidates generated with this method. Inspired by the apriori-gen function [36] used in frequent item set generation, we could use a similar method to further reduce the number of source set comparison. Let $S^a = \{s_1^a, s_2^a, ..., s_k^a\}$ and $S^b = \{s_1^b, s_2^b, ..., s_k^b\}$ denote two qualified source sets at level k in Fig.3, if and only if S^a and S^b satisfy the following conditions:

$$s_i^a = s_i^b \ (for\ i = 1, 2, ..., k-1)\ and\ s_k^a \neq s_k^b \tag{15}$$

we then examine them by calculating their similarity.

We repeat the merging process from level 2 to level n in turn. After the search space has been scanned once, we recheck the remaining source sets from level n to level 2 in turn. If there are more than two qualified source sets which have the same subset at the same level, we just remain that one which has the largest similarity for best source merging performance. If a source set is identified as a qualified source community, all its subsets will be pruned from the search space. Finally, all the remaining source sets are the source merging results, i.e., the qualified source communities.

Two issues should be specified after source merging: (1) Note that there might be conflicts between sources in the same community in providing answers to the same blank. In this case, we take the value owning the largest number of support sources as the answer provided by the community to the blank. (2) After merging similar sources into communities, we need to reassign different confidence to these communities instead of using the same initial confidence (precision) for each source as we did in Sec. 3.1. Specifically, we tend to give a higher confidence to a community if it contains more sources. For approximating the real distribution of the precision of sources, we employ the Gaussian distribution to help us set the initial confidence of these communities. Finally, similar to the computation process in Sec. 3.1, we compute the correct probability of each value by considering s as the (s, e) pattern and then output the results.

Example 2: Continue with Example 1, given $P(s_i) = 0.7$ for every $i \in [1, 5]$ initially, and the precision and recall of extractors are all 1 , we assume that s_3 and s_4 copy all the data from s_2, then we have $P_{joint}(s_1, s_2, s_3) = 0.7 * 1 * 1 = 0.7$ and $Corr(s_1, s_2, s_3) = (0.7 * 1 * 1)/(0.7 * 0.7 * 0.7) = 2.04$. Assume the five sources are independent, then we have $P(s_2) = P(s_3) = P(s_4) = 0.99$, $P(s_1) = 0.8$ and $P(s_5) = 0.6$ according to Eqs. (1-5). In this scenario, the correctness probability of the value e nearly 0. However, it is usually not the truth in real scenarios. If we consider s_2, s_3 and s_4 as one big source, denoted by s_{234}, and assign 0.9 as its initial precision, then we have the correctness probability of the value e approximately equals to 0.5.

For better explain the modularity and logics of the EM-based quality model for WebPut, we also have a meta

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2954087, IEEE Transactions on Knowledge and Data Engineering

algorithm to specify and how we call each module/formula in the process of the algorithm. Due to the space limitation, the Algorithm is put in the appendix.

3.3 Problems Analysis

In most cases the EM-based quality model could help get the right answer for the blanks in the objective database, but it still makes mistakes when its assumption does not hold in some specific cases. For instance, the ubiquitous copy operations among some sources will make an incorrect answer provided by these sources has pretty high confidence to be taken as the correct answer by WebPut. Also, the EM-based quality model may be paralyzed for some blanks where all the candidate values have small correctness probabilities.

4 Crowd-Aided Quality Control

To overcome the disadvantages of the EM-based quality model, we propose to use crowdsourcing as a complement. Different from the existing crowdsourcing-based data imputation approaches [29] which rely heavily on crowdsourcing to give answers to every unsolved imputation task. In this paper, we propose a crowd-economic way to let the crowd help adjust the quality model by imputing those values that the WebPut model are paralyzed for and diagnosing the potential biases of the EM-based quality model. Our target here is to improve the quality of imputation results with the proper crowd intervention cost.

To this end, we actively select a number of blanks for crowd imputation, and the imputation results can be used for two issues: *Precision of Sources Adjustment* and *Correlation Adjustment among Sources*. In the rest of this section, we first introduce how we do active selection for crowd intervention in Sec. 4.1, and then present when to stop further crowd guidance in Sec. 4.2.

4.1 Active Crowd Intervention

The purpose of active crowd intervention is to adjust the EM-based model with as few blanks for crowd-based imputation as possible. To achieve this, a set of "informative" and "diverse" blanks are expected to be selected into a sampling set for crowd intervention, where "informative" means having a small correctness probability of the imputation result in WebPut while "diverse" blanks should be able to help improve the precision and correlations of all the sources of the EM-based quality model without a bias.

Here we adopt Shannon Entropy to measure the "informativeness" of a blank b, denoted by $E(b)$ as follows:

$$E(b) = -\sum_{v_i \in V^b} Pr(v_c^b = v_i|b) log Pr(v_c^b = v_i|b) \quad (16)$$

where V^b contains all candidate imputation values provided by all the sources in S to a blank b. The higher the entropy value, the more informative the blank b is. If the distribution of the value is skewed, then the associated entropy value is low. On the contrary, if it is close to a uniform distribution, then the associated entropy value is high. For example, the entropy value reaches maximal when $Pr(v_c^b = v_i|b) = 1/|V^b|$ for each $v_i \in V^b$. WebPut outputs imputation result whose correctness probability is maximal among all the candidate values, thus a high entropy value often means a small correctness probability of the imputation result in WebPut.

Given the above, our blank selection strategy can be described below: Given all the blanks in the blank set B, we first select top-K blanks with the highest entropy values from B. Next, we select n_{sam} blanks from these top-K blanks into a sampling set B_{sam}, using weighted sampling, where the weighted sampling method could let the selected blanks be both "informative" and "diverse" as has been stated and proved in the literature [15]. Next, we let the crowd to provide the correct imputation answer to each of the n_{sam} blanks in the sampling set, such that we could make an estimation to the accuracy of WebPut over B_{sam} and then make proper adjustments to the precision of sources and the correlation among sources correspondingly.

The imputation values provided by a source s to the blanks in B_{sam} can be put into two categories: some of them would be accepted as the final imputation answers by WebPut according to the EM-based quality model, while the others would not. We call the former ones as *Model-Accept Values* denoted by $V^+(s, B_{sam})$ and the latter ones as *Model-Reject Values* denoted by $V^-(s, B_{sam})$. According to the correct imputation answers provided by the crowd, we could calculate the accuracy of the values in $V^+(s, B_{sam})$ as well as the accuracy of the values in $V^-(s, B_{sam})$, denoted by $P^+_{sam}(s)$ and $P^-_{sam}(s)$ respectively. For better understanding, we assume $P^+_{sam}(s)$ and $P^-_{sam}(s)$ are calculated by ignoring the possible wrong answers provided by the Crowd. But we take into account the accuracy of Crowd answers in our experiments and do the theoretical analysis in the appendix.

However, the estimated accuracy of a source s on the sampling set B_{sam} may not always be qualified to reflect its accuracy on the whole data set denoted by $P^+(s)$ and $P^-(s)$ respectively[1]. According to the literature [41], we could assume that using $P^+_{sam}(s)$ to estimate the value of $P^+(s)$ has an error margin ϵ^+ with a confidence δ , that is, $P^+(s) \in [P^+_{sam}(s) + \epsilon^+, P^+_{sam}(s) - \epsilon^+]$ with a confidence δ. Similarly, we assume using $P^-_{sam}(s)$ to estimate the value of $P^-(s)$ has an error margin ϵ^- with a confidence δ , that is, $P^+(s) \in [P^-_{sam}(s) + \epsilon^-, P^-_{sam}(s) - \epsilon^-]$ with a confidence δ. Then the values of ϵ^+ and ϵ^- can be computed as follows:

$$\epsilon^+ = Z_{1-\frac{\delta}{2}} \sqrt{\frac{P^+_{sam}(s)(1 - P^+_{sam}(s))}{n^+_{sam}} \cdot \frac{n^+ - n^+_{sam}}{n^+ - 1}} \quad (17)$$

$$\epsilon^- = Z_{1-\frac{\delta}{2}} \sqrt{\frac{P^-_{sam}(s)(1 - P^-_{sam}(s))}{n^-_{sam}} \cdot \frac{n^- - n^-_{sam}}{n^- - 1}} \quad (18)$$

1. The reason that we do the estimation on both model-accept values and model-reject values respectively is that: the samples on one of the two sets of values could be small, which requires a much smaller sampling set for the estimation than doing the estimation on the whole value set without using such a classification. Besides, we do not know the number of really true and false values of the whole data set but we know the number of mode-accept and model-reject value which makes our estimation strategy feasible.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2954087, IEEE Transactions on Knowledge and Data Engineering

where δ is a user-setting confidence (which is set to 95% in our experiments), n^+ and n^+_{sam} are the numbers of model-accept values provided by s on the whole data set and on the sampling set B_{sam} respectively, while n^- and n^-_{sam} are the numbers of model-reject values provided by s on the whole data set and on the sampling set B_{sam} respectively.

We use a threshold ϵ_{max} (which is set to $\epsilon_{max} = 0.1$ in our experiments) to find qualified estimations on sources, that is, if $\epsilon^+ \leq \epsilon_{max}$ (or $\epsilon^- \leq \epsilon_{max}$), we say $P^+_{sam}(s)$ (or $P^-_{sam}(s)$) is a qualified estimation to $P^+(s)$ (or $P^-(s)$), then we should adjust the precision of the source s accordingly. Otherwise, the estimation is unqualified and we take no further adjustment step to s.

To facilitate the Crowd guidance process, our method is based on the following two propositions which are about the adjustments to the *Precision of Sources* and *Correlation among Sources* respectively:

Proposition 1: If the EM-based quality model always takes the correct answers provided by a source s as incorrect ones, then the precision of s should be increased. The vice versa.

Proposition 2: If the EM-based quality model always takes the correct answers provided by a set of source communities S as incorrect ones, then the joint precision of all the source communities in S (w.r.t. correlation among source communities) which provide these correct answers should be larger than the value by assuming these source communities being independent. The vice versa.

Example 3: Continue with Example 1 and Example 2, suppose we find that our WebPut model always takes the correct values provided by s_1 and s_2 as incorrect ones with the help of crowd workers. If we assign a higher initial precision to s_1 and s_2, we could get a higher correctness probability for the value e (please refer to Example 2), such that e would be taken as a right answer by our WebPut Model. Likewise, we can use the same mechanism to adjust the correlation factor of sources.

In the following, we show how to improve the EM-based quality model according to $P^+_{sam}(s)$ and $P^-_{sam}(s)$.

1) Precision of Sources Adjustment: Given a threshold ω_1, if $P^+_{sam}(s) < \omega_1$, we say the EM-based quality model mistakenly predicts a large number of the answers provided by source s as correct ones. According to Proposition 1, the precision of s should be decreased to improve the weight of source s in the quality model. Basically, the value of $P(s)$ decreases with $P^+_{sam}(s)$. Denoting $P(s)$ the current prior precision of s, we adopt the following rule to update $P(s)$ into $P'(s)$.

$$P'(s) = P(s) - \alpha \cdot (\omega_1 - (P^+_{sam}(s))) \tag{19}$$

where α is a parameter to control the step size of our adjustment.

Likewise, given a threshold ω_2, if $P^-_{sam}(s) < \omega_2$, we say the EM-based quality model mistakenly predicts a large number of answers provided by source s as incorrect ones. Similar to Eq. 19, denote $P(s)$ the current prior precision of s, we adopt the following rule to update $P(s)$ into $P'(s)$.

$$P'(s) = P(s) + \beta \cdot (\omega_2 - (P^-_{sam}(s))) \tag{20}$$

where β is a parameter to control the step size of our adjustment.

2) Correlation Factor Adjustment: If we consider a source as source community, we then can address the scenario of source community (referring to source merging step in EM-based quality model) using the same method above which is also described in Proposition 2. However, we can not always do that for all the sources which may only provide a portion of same answers for imputing all the blanks. In this circumstance, we need to adjust the correlation factor of a source set S as described in Eq. 11. Proposition 2 provides us a good enlightenment for doing that. Recall Eq. 3, the correctness probability of a value v, $Pr(v^b_c = v|\vec{V}^b, b)$, is determined by the joint distribution (referring to $Pr(\vec{V}^b|v^b_c = v, b)$) of those sources that provide the value v. More clearly, the larger the $Pr(\vec{V}^b|v^b_c = v, b)$ value, the larger the value of $Pr(v^b_c = v|\vec{V}^b, b)$ is. We now substitute Eq. 11 into Eq. 2, then we can rewrite Eq. 2 as follows:

$$Pr(\vec{V}^b|v^b_c = v, b) = \prod_{S' \in S} Corr(S') \prod_{(s_i,e_j)\in SE} Pr((s_i, e_j) \to v_{i,j}|v^b_c = v, b) \tag{21}$$

Clearly, larger $Corr(S')$ value will lead to larger $Pr(v^b_c = v|\vec{V}^b, b)$ value. Since $Corr(S')$ is somewhat abstract for us, directly increasing or decreasing its value is intractable for us. Thus, we first translate it into an easy way for better control. Given a source set S in which all the sources provide the values for each blank, we update its precision similar to the adjustment method in the **Precision of Sources Adjustment**, that is, we update $P(S)$ into $P'(S)$ according to Eq. 19 and Eq. 20 as follows:

$$P'(S) = P(S) - \alpha \cdot (\omega_1 - (P^+_{sam}(s))) \tag{22}$$

$$P'(S) = P(S) + \beta \cdot (\omega_2 - (P^-_{sam}(s))) \tag{23}$$

Then we rewrite Eq. 3 as

$$Pr(v^b_c = v|\vec{V}^b, b) = \frac{Pr(\vec{V}^b|v^b_c = v, b)}{Pr(\vec{V}^b|v^b_c = v, b) + t} \tag{24}$$

where t is a constant. Then by transforming above equation, we can easily obtain that

$$Pr(\vec{V}^b|v^b_c = v, b) = \frac{t \cdot Pr(v^b_c = v|\vec{V}^b, b)}{1 - Pr(v^b_c = v|\vec{V}^b, b)} \tag{25}$$

For a source set S, assuming the other sources are independent, we then have the following equation by substituting Eq. 21.

$$Corr(S) = \frac{t \cdot Pr(v^b_c = v|\vec{V}^b, b)}{u \cdot (1 - Pr(v^b_c = v|\vec{V}^b, b))} \tag{26}$$

where $u = \prod_{(s_i,e_j)\in SE} Pr((s_i, e_j) \to v_{i,j}|v^b_c = v, b)$. But note that $Pr(v^b_c = v|\vec{V}^b, b)$ may have different values for different blanks in the samples, so here we use the average correct possibility of those values to substitute $Pr(v^b_c =$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2954087, IEEE Transactions on Knowledge and Data Engineering

$v|\vec{V}^b, b)$ for estimating $Corr(S)$. Besides, $Corr(S)$ has different values between the same correct answers of S, denoted by $Corr^+(S)$, and the the same incorrect answers of S, denoted by $Corr^-(S)$. Consequently, we can then update $Corr^+(S)$ into $Corr^+(S)'$ and $Corr^-(S)$ into $Corr^-(S)'$ according to the following two equations.

$$Corr^+(S)' = \frac{t}{u} \cdot \frac{(u \cdot Corr^+(S) + t)}{t + \alpha \cdot (\omega_1 - (P^+_{sam}(s)))(u \cdot Corr^+(S) + t)} - t \quad (27)$$

$$Corr^-(S)' = \frac{t}{u} \cdot \frac{(u \cdot Corr^-(S) + t)}{t - \beta \cdot (\omega_2 - (P^-_{sam}(s)))(u \cdot Corr^-(S) + t)} - t \quad (28)$$

4.2 When to Stop Crowd Intervention

An important issue of using Crowdsourcing is to decide when to stop further crowd guidance to save the crowd intervention overhead. In our case, we need to consider the stop conditions for two crowd intervention operations, that is, when to stop adjusting the precision of source and when to stop adjusting the correlation among sources with single sampling.

Intuitively, we stop using crowd to do further adjustment to the quality model when the performance of the EM-based quality model becomes stable i.e., the accuracy of imputation results holds steady. However, given that the change with a single round of adjustment is not that reliable, we only make our decisions on stopping further adjustment to the precision of a specific source (or the correlation among specific sources) when we observe a stable performance to the precision (or correlation) after a new round of adjustments.

As Kappa statistics is a commonly used statistic that takes into account the fact that observers will sometimes agree or disagree simply by chance, here we adopt the Kappa statistics [37] to measure the degree of stability of the quality model. Specifically, the kappa coefficient κ can be calculated as follows:

$$\kappa = \frac{A_o - A_e}{1 - A_e} \quad (29)$$

where A_o is an observed agreement and A_e is an agreement expected by chance. The Kappa statistic measures agreement expected by chance by modeling each quality model at a state. Formally, we calculate A_e as follows:

$$A_e = \sum_{c \in \{-1,1\}} Pr(c|M_t) \cdot Pr(c|M_{t+1}) \quad (30)$$

where M_t and M_{t+1} are the quality model at two different states t and $t+1$, and $Pr(c|M_t)$ is the precision that the Model M_t claims a value as correct or incorrect.The range of κ is between -1 and 1. When $\kappa = 1$, it means the prediction values of M_t and M_{t+1} are completely consistent.

We test the sensitivity of the kappa coefficient in our experiments, which show that κ (which is set as 0.8 in our experiment) is quite robust. One of the advantages of this method for stopping crowd-aided is that the users can control how aggressive our model performs by giving some constraints (such as the cost limitation).

In practice, the agreements between models of several consecutive states may fluctuate due to some casual factors, such as wrong crowd-based imputation to some blanks. Thus, we implement a smoothing window of size k ($k = 3$ is proper in our experiment) to average the agreements among the most recent k models.

5 Experiments

This section presents our experimental results. The experimental environment is a computer with four-core Intel Core i7 processor, 16GB memory, running Windows 8. All the approaches are implemented by Java.

5.1 Data Sets and Metrics

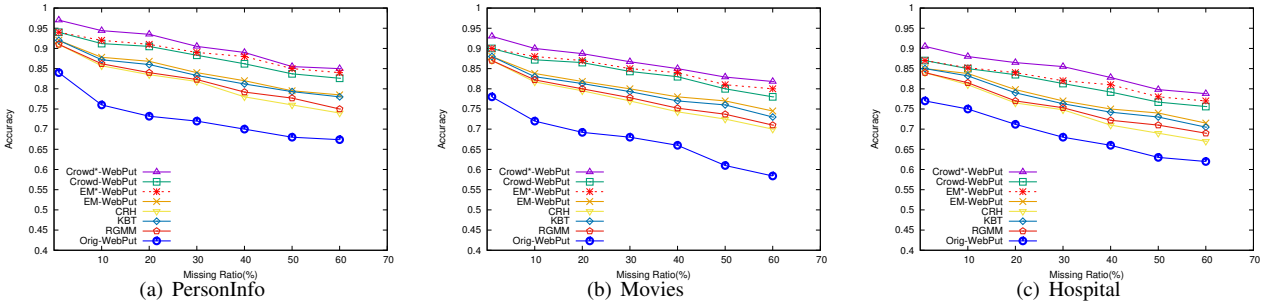
We experiment on three real and one simulated data sets.
Real Data Sets:

1) Personal Information Table (**PersonInfo**): This is a round 50k-tuples, 9-attributes table that has been used previously in WebPut [26] and TRIP [24], which contains contact information for academics including name, email, title, university, street, city, state, country and zip code. This information was collected from more than 1000 different universities in USA, UK, Canada and Australia.
2) Multilingual Movies Table[2] (**Movies**): This table contains names of about 15k movies in 5 different languages collected from Wikipedia and MovieLens.
3) Hospital Information Table[3] (**Hospital**): This data set has 25k tuples under 6 attributes. The table contains some basic contact information of hospitals located in USA and Canada, including their names, countries, states, zip codes, street addresses and phone number.

The three data sets above are complete relational tables. To generate incomplete tables for our experiments, we remove attribute values at random positions from the complete table, while making sure that at least one key attribute value will be kept in each tuple. Each reported result is the average of 3 evaluations, that is, for each missing value percentage (1%, 10%, 20%, 30%, 40%, 50%, 60%), 3 incomplete tables will be generated with 3 random seeds, and the experimental results we present are the average results based on the 3 generated incomplete tables. We perform data imputation to these generated incomplete tables and then evaluate the solution by using the original complete table as ground truth. We tested these three real-world datasets on Amazon Mechanical Turk (AMT), which is a widely used crowdsourcing marketplace. Each answer for a blank is awarded 0.03 USD for all the three datasets. We manually create qualification test by selecting 20 tasks, and each worker should pass the qualification test before he could work on our tasks.

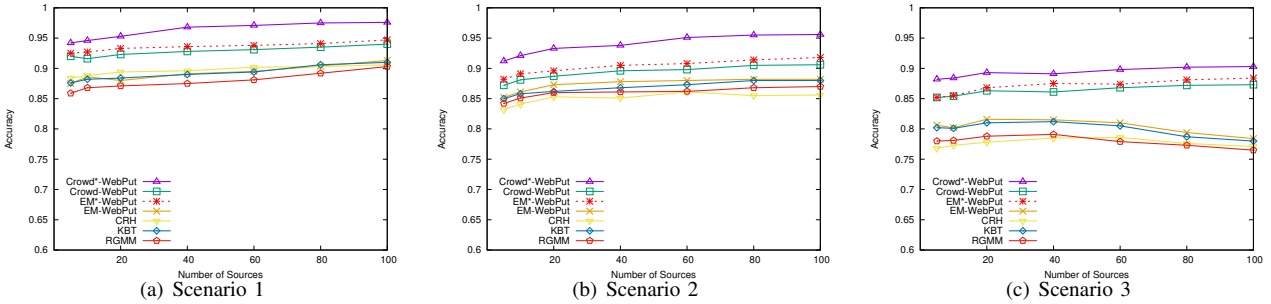
2. https://pan.baidu.com/s/1DOUHNwsn6FSYIVuyIxoyBA
3. https://pan.baidu.com/s/1anuLIuVGkxa8jS8PXNhN3A

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2954087, IEEE Transactions on Knowledge and Data Engineering



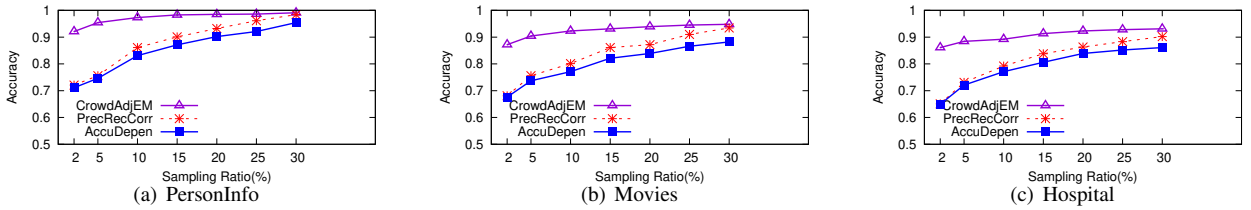
(a) PersonInfo (b) Movies (c) Hospital

Fig. 4. Comparing the Effectiveness of the Proposed Quality Models for WebPut on the Three Data Sets



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Fig. 5. Comparing the Effectiveness of the Proposed Quality Models for WebPut on the Simulated Data



(a) PersonInfo (b) Movies (c) Hospital

Fig. 6. Evaluating the Influence of the Sampling Set Size on the Three Data Sets

Simulated Data: For better evaluating the performance of our method, we simulate different truth finding scenarios to compare our method with the existing ones.

Data Generation. In each experiment, we generate 100 sources 5 extractors and 50000 blanks. For all extractors, their precisions and recalls are drawn from uniform distribution. For each source, its precision is drawn from a distribution which we will describe below. In order to generate the correlation among sources, we first generate 20 sources independently. Next, for the rest of 80 sources, we randomly copy some data (ranging from 10% – 90%) from one of the 20 sources and then randomly change 10% of these duplicated data. Each of the 80 sources only duplicates data from one other source once. But all the sources satisfy their precision limit.

Scenario 1: P(s) ~ Uniform (a, b). In this scenario, the precisions of sources are drawn from a uniform distribution with $a = 0.5$ and 0.9. The precisions of sources are uniformly distributed.

Scenario 2: P(s) ~ Normal (μ, σ^2). In this scenario, the precisions of sources are drawn from a Normal distribution with mean $\mu = 0.7$ and variance $\sigma^2 = 0.04$. The smaller the σ is, the more centralized the precisions of sources.

Scenario 3: P(s) ~ Student's $t - (\nu)$. In this scenario, the precisions of sources are drawn from a student's t-distribution with freedom $\nu = 2$. Student's t-distribution has heavier tails when the freedom ν is smaller. As a result, there are more sources with very low or high precision.

Metrics: We evaluate the effectiveness of the proposed quality models in improving the quality of the WebPut results with *Imputation Accuracy*, which is the percentage of the correctly imputed blanks among all the blanks in a data set. Besides, we use *the Number of Crowd Intervention Times*, denoted by N_{crowd}, to evaluate our efficiency optimization strategy.

5.2 Effectiveness of the Quality Models

We now evaluate the effectiveness of the quality models by comparing the imputation accuracy of the following several imputation approaches.

a) **Orig-WebPut**: The original WebPut approach, which simply do Web-based data imputation with no quality control model as described in the literature [26].
b) **EM-WebPut**: This is the WebPut approach with only the proposed EM-based quality control model.
c) **Crowd-WebPut**: This is the WebPut approach with the crowd-guided EM-based quality control model. In order

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2954087, IEEE Transactions on Knowledge and Data Engineering

to evaluate the effectiveness of the crowd guidance in adjusting the EM-based model, we do not directly take the blanks labeled by crowd into account in calculating the imputation accuracy.

d) **EM*-WebPut**: Also, to further evaluate the effectiveness of the crowd guidance in adjusting the EM-based model, we also design this "virtual" approach which adopts the true values of all the three groups of factors to operate the EM-based model.

e) **Crowd*-WebPut**: This is the WebPut approach with the Crowd guided EM-based quality control model, which takes the blanks labeled by crowd into account in calculating the imputation accuracy. Therefore, this is the final quality model we could reach in this paper.

f) **CRH** : CRH [23] formulates truth finding problem as an optimization problem, seeking the optimal truth and source reliability to minimize the weighted deviation between the truth and the observations.

g) **KBT** : [9] proposes a multi-layer probabilistic model to solve truth finding problem. In their framework, source reliability, extractor reliability and truth are model parameters to estimate.

h) **RGMM** : In [44], truth finding problem is modeled as seeking the maximum likelihood estimate of truth while incorporating source bias. Based on EM techniques, RGMM proposes population-based and sample-based solutions to solve the problem.

For the methods Orig-WebPut, CRH and RGMM, we only use one extractor with the highest precision since they do not consider the role of extractors. In other words, these methods take the data extracted by the extractor as the real data of sources. Some examples of the extractors for each dataset can be found in the appendix.

Results on Real Data: As can be observed in all the three sub-figures in Fig. 4, as the missing ratio increases, the performance of all the eight approaches decreases linearly, but the Orig-WebPut always reaches the lowest imputation accuracy. While EM-WebPut, KBT, CRH and RGMM reach similar performance without the help of Crowd, the Crowd-WebPut could further improve the EM-WebPut by $5 - 7.5\%$. Besides, The pretty closeness between the line of Crowd-WebPut and EM*-WebPut shows that the crowd guidance can adjust the EM-based model to approach the best state it could achieve. Finally, the Crowd*-WebPut model could still improve the performance of the Crowd-WebPut and EM*-WebPut by around $1.8 - 3.2\%$, which shows that Crowd*-WebPut can effectively pick up those can not be predicted correctly by EM-WebPut.

Results on Simulated Data: Since the missing ratio makes no sense on the simulated data, we report the experimental results with different source number $S = \{10, 20, ..., 100\}$ in terms of all methods. From Fig. 5, we can see that the performance of all the seven approaches increases approximately linearly as the number of sources increase in scenario 1 and 2. All the methods perform best in scenario 1 because there are more high-precision sources and many sources copy from them. In Scenario 2, Crowd-WebPut's improvement is not as prominent as that in Scenario 1 and 3 due to the reason that the precisions of most sources are similar. So even there are some complex correlation among the sources, it will not affect the imputation accuracy a lot. In scenario 3, we surprisingly find that the performance of EM-WebPut, KBT, RGMM and CRH decreases when the number of sources reach over 40. The reason is that there are a lot of low-precision sources and many sources copy data from them and change the data. Thus these methods without correlation detection can not keep good performance, while Crowd-WebPut and Crowd*-WebPut can always reach good performance even with complex correlations among sources.

PersonInfo(10%)	EM-WebPut	Crowd-WebPut	Crowd*-WebPut
NoExtractor	0.852	0.876	0.891
WithExtractor	0.878	0.912	0.954
Movie(30%)	**EM-WebPut**	**Crowd-WebPut**	**Crowd*-WebPut**
NoExtractor	0.783	0.795	0.826
WithExtractor	0.802	0.843	0.875
Hospital(50%)	**EM-WebPut**	**Crowd-WebPut**	**Crowd*-WebPut**
NoExtractor	0.711	0.724	0.762
WithExtractor	0.739	0.767	0.803

TABLE 2
Evaluating the Role of Extractors

5.3 Effect of the accuracy of Crowd Answers

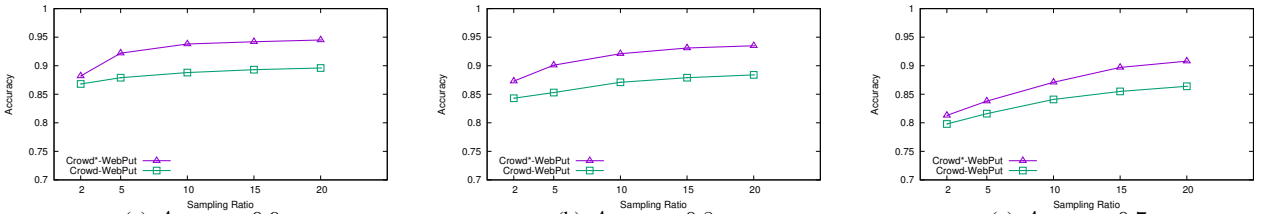
This section evaluates the effect of the accuracy of crowd answers to the performance of our approach. Here we use a simulated crowd in scenario 2 for evaluation. We denote the accuracy of Crowd Answers as A_{crowd}. We find that the higher A_{crowd} is, the faster the convergence is. This finding is supported by observing Fig. 7(a)(b)(c). In other words, we would need more answers for Crowd with lower accuracy. Moreover, regardless of the setting of A_{crowd}, the accuracy of our model is gradually increased with the reception of the Crowd Answers. This suggests that the crowd is conducive for our model in general.

5.4 Evaluating the Role of Extractors

In this section, we evaluate the role of extractors in **EM-WebPut**, **Crowd-WebPut** and **Crowd*-WebPut** under two different situations: (1) only using one extractor with the highest accuracy; (2) considering the precision and recall of extractors we proposed in this paper. Situation (1) takes the data extracted by extractors as the data of sources which means ignoring the errors of extractors. So that we call this method as **NoExtractor** and the method under Situation (2) as **WithExtractor**.

As can be seen from Table 2 (the missing ratio of PersonInfo, Movie and Hospital are respectively 10%, 30% and 50%), **WithExtractor** has better performance than **NoExtractor** with around $1.9\% - 2.8\%$. Beside, **WithExtractor** could provide better framework for Crowd-Aided Model since we can see that Crowd-WebPut and Crowd*-WebPut could only improve EM-WebPut by $1.2\% - 2.4\%$ and $3.3\% - 5.1\%$ with **NoExtractor**. While using **WithExtractor**, Crowd-WebPut and Crowd*-WebPut could improve EM-WebPut by $2.8\% - 4.1\%$ and $6.4\% - 7.6\%$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2954087, IEEE Transactions on Knowledge and Data Engineering



(a) $A_{crowd} = 0.9$ (b) $A_{crowd} = 0.8$ (c) $A_{crowd} = 0.7$

Fig. 7. Evaluating the Effect of the accuracy of Crowd Answers on the Stimulated Data Set

Dataset		CrowdImp	CrowdAdjEM	SmartCAdjEM
PersonInfo	Accuracy	0.973	0.954	0.951
	N_{crowd}	1143	456	**308**
Movies	Accuracy	0.931	0.904	0.896
	N_{crowd}	1498	482	**371**
Hospital	Accuracy	0.915	0.884	0.882
	N_{crowd}	1307	421	**346**

TABLE 3
Evaluating the Crowd Cost Saving Strategies

5.5 Evaluating the Crowd Cost Saving Strategy

To evaluate the effectiveness of our crowd cost saving strategy, we compare the number of crowd intervention times per 10000 blanks as well as the imputation accuracy of the following four methods on the three data collections by setting the missing ratio to 10%.

a) **CrowdImp**: This baseline asks the crowd to provide the correct missing value for every missing blank that can not be filled with a high-confidence value by WebPut.
b) **CrowdAdjEM**: This is the method we use the crowd to help adjust the EM model for web-based data imputation. This method sets a fixed size of sampling set (say 5%) for crowd intervention in order for improving the EM model.
c) **SmartCAdjEM**: On the basis of CrowdAdjEM, we also adopt the Kappa statistics to help find a proper time to stop doing further adjustment with crowd intervention for saving crowd cost.

As can be observed in Table 3, on all the three data sets, the CrowdImp always uses the maximum N_{crowd} to reach the highest imputation accuracy. Comparatively, the CrowdAdjEM uses only round 30% N_{crowd} of that used by CrowdImp by only sacrificing within 0.05 imputation accuracy, which proves the advantage of our crowd intervention way compared to the baseline one. Finally, the SmartCAdjEM can further reduce around 20% more N_{crowd} of that used by CrowdAdjEM by only slightly decreasing the imputation accuracy a bit. Overall, the SmartCrowdAidEM strategy can effectively save more than 75% crowd cost.

5.6 Evaluation with Correlation and Crowd Cost

In this section, we compare our methods with two state-of-the-art models which also take the correlation among sources in consideration.

d) **AccuDepen**: [49] takes into consideration the copying relationships between sources with a Bayesian truth detection model. AccuDepen penalizes the vote count of a source if the source is detected to be a copier of another source.
e) **PrecRecCorr**: [33] considers correlation between sources and applies the inclusion-exclusion principle for exact solution. This work relies on training data which uses the crowdsourcing platforms to facilitate the labeling process.

Here we need to point out that correlation detection inevitably involves prior knowledge as we need to know which values are really true or false. In data imputation system, it is unreasonable to assume that we always have some gold standard data which can correctly estimate the correlation among sources. So the cost of Crowd (or prior knowledge) is an important part to be evaluated.

As illustrated in Fig. 6, CrowdAdjEM only needs 5% of the data to reach a satisfying accuracy, while PrecRecCorr needs at least 20% of the data since PrecRecCorr requires a lot of data to estimate the quality of sources with small bias in prior. The performance of AccuDepen is similar with that of PrecRecCorr. However, since AccuDepen only considers the copying relationships between sources, the accuracy of Accudepen is a little less than that of PrecRecCorr.

6 Related Work

Data imputation has been studied for decades. While most of the previous efforts focus on recovering missing quantitive data, which is either continuous data or discrete data with a finite number of values [27], only a small portion of attention has been paid on recovering missing non-quantitive data with an infinite number of values.

Existing techniques for non-quantitive data imputation can be roughly divided into three categories, i.e., *Local-based approaches*, *Crowdsourcing-based approaches* and *External resource based approaches*. The local-based approaches mainly deduce missing values based on the complete part of data set [17], [16], [6], which can only reach limited imputation recall due to lacking of enough knowledge about the missing data. To reach higher imputation recall, some recent work turn to outsource the task to crowd workers [48], [45], [29] when the traditional methods are not capable of filling the missing values. However, crowd-based imputation could be expensive given that it pays money for every human input. Differently, the external resource based approaches resort to external sources such as existing domain databases, online encyclopedia [42] or the world wide web [25], [26], [19], [40] for answering

the missing values, which could reach a much higher imputation recall with much less human cost.

Much work has been conducted to harvest missing values from either web lists [11], [19], or web tables [18], [22], but a more general web-based imputation approach called WebPut was proposed to impute missing values from all kinds of web documents [25], [26]. Typically, WebPut formulates imputation queries based on the existing information of the original data set, and then combine the Information Extraction techniques to get more concrete and efficient queries for the purpose of getting more precise imputation results. WebPut has been proved to reach a higher recall than the local-based methods. However, there are also some kinds of missing values WebPut can not work well, since either too much noises on the Web or little information about the missing values.

To improve the quality of values retrieved from the Web, there has been a lot of work assessing the quality of web sources by PageRank [2], Authority analysis [21], web spams [4] and so on [20], [35]. Our work is relevant to the truth-finding problem [47], [10] whose goal is to find the truths from data provided by multiple sources. Most of the recent work in this field evaluates the confidence of sources based on link-based measures [30], [31], IR-based measures [43], accuracy-based measures [33], [23]. Li et al. [23] propose an optimization framework, seeking the optimal truth and source reliability to minimize the weighted deviation between the truth and the observations. Besides, some probabilistic methods [44], [47], [9], [32] were developed to address the truth-finding problem. The basic idea is to formulate source data as certain mixture of distributions and incorporate source reliability as some random variable into the probabilistic models. These approaches differ in the way of selecting proper distributions and evaluating sources. To find the truth, these approaches usually leverage the EM method which tends to maximize likelihood and iteratively update model parameters for inference. But most of them neglect the correlation among sources and assume that the sources are independent by default if they have no ground truth, thus they can not accurately find the truth in some cases. An experimental evaluation for truth-finding can be found in [38].

The correlation among sources has been studied previously. The most typical correlation detection is copy detection which has been studied in [7], [8] for structured data. Some recent work [33] proposes an exactly complete correlation detection including copying, positive correlation and negative correlation which could measure correlation among sources pretty accurately, but it often needs a lot of prior knowledge, i.e., a lot of manual labels. So far, no unsupervised methods could tackle the complete correlation among sources. [5] introduces an active-learning-based truth estimator for social networks, but it needs an influence network which we can not obtain in our problem.

7 Conclusions and Future Work

In this paper, we propose to build a quality model to control the quality of web-based data imputation results. An EM-based model is firstly built, which is then improved with crowd intervention. The experimental study conducted on several data collections proves that the EM-based quality model improves the imputation quality by 10%, while the crowd intervention further enhances the quality by 5%. Besides, our proposed strategies could save 75+% of the crowd cost.

This work only considers one possible way to improve the quality of WebPut with crowdsourcing, our future work would try some other ways such as taking the crowdsourcing as a sort of data sources to complement the drawbacks of the web sources. Besides, this paper does not discuss on the possibility that the employed crowd workers could also make mistakes in data imputation, and different workers may get different payment to accomplish the same task. Our ongoing work expects to find a more applicable way for crowd intervention.

Acknowledgements: This research is partially supported by the Natural Science Foundation of Jiangsu Province (No. BK20191420), National Natural Science Foundation of China (No. 61632016, 61572336, 61572335, 61772356), and the Natural Science Research Project of Jiangsu Higher Education Institution (No. 17KJA520003, 18KJA520010).

References

[1] P. Bohannon, W. Fan, F. Geerts, X. Jia, and A. Kementsietsidis. Conditional functional dependencies for data cleaning. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 746–755. IEEE, 2007.

[2] S. Brin and L. Page. Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer networks*, 56(18):3825–3833, 2012.

[3] R. A. Brualdi. *Introductory combinatorics*. New York, 1992.

[4] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri. Know your neighbors: Web spam detection using the web topology. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 423–430. ACM, 2007.

[5] H. Cui, T. Abdelzaher, and L. Kaplan. A semi-supervised active-learning truth estimator for social networks. In *The World Wide Web Conference*, pages 296–306. ACM, 2019.

[6] A. R. T. Donders, G. J. van der Heijden, T. Stijnen, and K. G. Moons. Review: a gentle introduction to imputation of missing values. *Journal of clinical epidemiology*, 59(10):1087–1091, 2006.

[7] X. L. Dong, L. Berti-Equille, Y. Hu, and D. Srivastava. Global detection of complex copying relationships between sources. *Proceedings of the VLDB Endowment*, 3(1-2):1358–1369, 2010.

[8] X. L. Dong, L. Berti-Equille, and D. Srivastava. Truth discovery and copying detection in a dynamic world. *Proceedings of the VLDB Endowment*, 2(1):562–573, 2009.

[9] X. L. Dong, E. Gabrilovich, K. Murphy, V. Dang, W. Horn, C. Lugaresi, S. Sun, and W. Zhang. Knowledge-based trust: Estimating the trustworthiness of web sources. *Proceedings of the VLDB Endowment*, 8(9):938–949, 2015.

[10] X. L. Dong, B. Saha, and D. Srivastava. Less is more: Selecting sources wisely for integration. *Proceedings of the VLDB Endowment*, 6(2):37–48, 2012.

[11] H. Elmeleegy, J. Madhavan, and A. Halevy. Harvesting relational tables from lists on the web. *Proceedings of the VLDB Endowment*, 2(1):1078–1089, 2009.

[12] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[13] S. Fortunato and D. Hric. Community detection in networks: A user guide. *Physics Reports*, 659:1–44, 2016.

[14] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. Crowddb: answering queries with crowdsourcing. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 61–72. ACM, 2011.

[15] C. Gokhale, S. Das, A. H. Doan, N. Rampalli, N. Rampalli, J. Shavlik, and X. Zhu. Corleone: hands-off crowdsourcing for entity matching. In *ACM SIGMOD International Conference on Management of Data*, pages 601–612, 2014.

[16] J. W. Grzymala-Busse, W. J. Grzymala-Busse, and L. K. Goodwin. Coping with missing attribute values based on closest fit in preterm birth data: A rough set approach. *Computational intelligence*, 17(3):425–434, 2001.

[17] J. W. Grzymala-Busse and M. Hu. A comparison of several approaches to missing attribute values in data mining. In *Rough sets and current trends in computing*, pages 378–385. Springer, 2001.

[18] R. Gummadi, A. Khulbe, A. Kalavagattu, S. Salvi, and S. Kambhampati. Smartint: using mined attribute dependencies to integrate fragmented web databases. *Journal of Intelligent Information Systems*, 38(3):575–599, 2012.

[19] R. Gupta and S. Sarawagi. Answering table augmentation queries from unstructured lists on the web. *Proceedings of the VLDB Endowment*, 2(1):289–300, 2009.

[20] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th international conference on World Wide Web*, pages 640–651. ACM, 2003.

[21] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[22] G. Koutrika. Entity reconstruction: Putting the pieces of the puzzle back together. *HP Labs, Palo Alto, USA*, 2012.

[23] 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.

[24] Z. Li, L. Qin, H. Cheng, X. Zhang, and X. Zhou. Trip: An interactive retrieving-inferring data imputation approach. *IEEE TKDE*, 27(9):2550–2563, 2015.

[25] Z. Li, M. A. Sharaf, L. Sitbon, S. Sadiq, M. Indulska, and X. Zhou. Webput: Efficient web-based data imputation. In *Web Information Systems Engineering-WISE 2012*, pages 243–256. Springer, 2012.

[26] Z. Li, M. A. Sharaf, L. Sitbon, S. Sadiq, M. Indulska, and X. Zhou. A web-based approach to data imputation. *World Wide Web*, 17(5):873–897, 2014.

[27] S. G. Liao, Y. Lin, D. D. Kang, D. Chandra, J. Bon, N. Kaminski, F. C. Sciurba, and G. C. Tseng. Missing value imputation in high-dimensional phenomic data: imputable or not, and how? *BMC bioinformatics*, 15(1):346, 2014.

[28] G. J. Mclachlan and T. Krishnan. *The EM Algorithm and Extensions, Second Edition*. Springer, 2007.

[29] H. Park and J. Widom. Crowdfill: Collecting structured data from the crowd. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 577–588. ACM, 2014.

[30] J. Pasternack and D. Roth. Knowing what to believe (when you already know something). In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 877–885. Association for Computational Linguistics, 2010.

[31] J. Pasternack and D. Roth. Making better informed trust decisions with generalized fact-finding. In *IJCAI*, volume 11, pages 2324–2329, 2011.

[32] J. Pasternack and D. Roth. Latent credibility analysis. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1009–1020. ACM, 2013.

[33] R. Pochampally, A. Das Sarma, X. L. Dong, A. Meliou, and D. Srivastava. Fusing data with correlations. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 433–444. ACM, 2014.

[34] J.-J. Shen, C.-C. Chang, and Y.-C. Li. Combined association rules for dealing with missing values. *Journal of Information Science*, 2007.

[35] A. Singh and L. Liu. Trustme: anonymous management of trust relationships in decentralized p2p systems. In *Peer-to-Peer Computing, 2003.(P2P 2003). Proceedings. Third International Conference on*, pages 142–149. IEEE, 2003.

[36] Tan, P. N. Steinbach, M. Kumar, and Vipin. *Introduction to data mining* =. Posts & Telecom Press, 2006.

[37] A. J. Viera, J. M. Garrett, et al. Understanding interobserver agreement: the kappa statistic. *Fam Med*, 37(5):360–363, 2005.

[38] D. A. Waguih and L. Berti-Equille. Truth discovery algorithms: An experimental evaluation. *arXiv preprint arXiv:1409.6428*, 2014.

[39] Q. Wang, J. Rao, et al. Empirical likelihood-based inference under imputation for missing response data. *The Annals of Statistics*, 30(3):896–924, 2002.

[40] R. Wang and W. Cohen. Iterative set expansion of named entities using the web. In *ICDM*, pages 1091–1096. IEEE, 2008.

[41] L. Wasserman. *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013.

[42] R. West, A. Paranjape, and J. Leskovec. Mining missing hyperlinks from human navigation traces: A case study of wikipedia. In *WWW*, pages 1242–1252, 2015.

[43] M. Wu and A. Marian. Corroborating answers from multiple web sources. In *WebDB*, 2007.

[44] H. Xiao, J. Gao, Z. Wang, S. Wang, L. Su, and H. Liu. A truth discovery approach with theoretical guarantee. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1925–1934. ACM, 2016.

[45] C. Ye and H. Wang. Capture missing values based on crowdsourcing. In *Wireless Algorithms, Systems, and Applications*, pages 783–792, 2014.

[46] S. Zhang. Parimputation: From imputation and null-imputation to partially imputation. *IEEE Intelligent Informatics Bulletin*, 9(1):32–38, 2008.

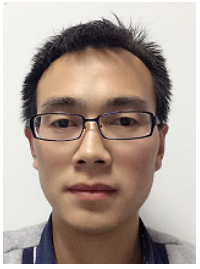
[47] B. Zhao, B. I. Rubinstein, J. Gemmell, and J. Han. A bayesian approach to discovering truth from conflicting sources for data integration. *Proceedings of the VLDB Endowment*, 5(6):550–561, 2012.

[48] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng. Truth inference in crowdsourcing: Is the problem solved? *Proceedings of the VLDB Endowment*, 10(5):541–552, 2017.

[49] 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.

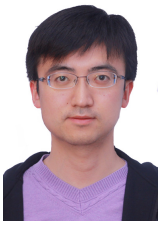
APPENDIX

Due to the limitation of space, we put the appendix online: https://pan.baidu.com/s/1Nvk2jmKmj_s6RZUJWHnPFA



Binbin Gu is a researcher at the Research Center on Advanced Data Analytics (ADA) in Soochow University, China. His research interests include Knowledge Fusion, Information Extraction and NLP. He has published several papers at TKDE, ICDE and DASFAA. He is also an external reviewer of several International Conferences such as CIKM, DASFAA, ADC, APWEB-WAIM etc.

Zhixu Li is an associate professor in the Department of Computer Science & Technology at Soochow University, China. He used to work as a research fellow at KAUST. He received his Ph.D. degree from the University of Queensland in 2013, and his B.S. and M.S. degree from Renmin University of China in 2006 and 2009 respectively. His research interests are Knowledge Graph, Question Answering, Data Quality and various Big Data Applications.



An Liu is a professor in the Department of Computer Science & Technology at Soochow University. Prior to that in 2014, he was a Senior Research Associate in the Joint Research Center of City University of Hong Kong (CityU) and University of Science & Technology of China (USTC). He received his Ph. D. from both CityU and USTC in 2009. His research interests include security, privacy, and trust in emerging applications; cloud computing; and services computing.

Jiajie Xu is an associate professor in the Department of Computer Science and Technology at Soochow University. He got his Ph.D. and Master degree from Swinburne University of Technology and University of Queensland in 2006 and 2011 respectively, and then worked in the Institute of Software, Chinese Academy of Sciences as assistant professor before joining Soochow University. His research interests are Spatio-temporal Database Systems and Big Data Analytics.

Lei Zhao is a Professor with the School of Computer Science and Technology at Soochow University. He received his Ph.D. degree in Computer Science from Soochow University in 2006. His recent research is to analyze large graph database in an effective, efficient, and secure way. He has published over 100 papers including more than 20 published in well-known journals and conferences such as ICDE, DASFAA, WISE, JCST.

Xiaofang Zhou is a Professor of computer science with The University of Queensland. He is the Head of the Data and Knowledge Engineering Research Division. He has been working in the area of spatial and multimedia databases, data quality, high performance query processing, Web information systems and bioinformatics, co-authored over 250 research papers with many published in top journals and conferences.