

Provenance-based Reconciliation In Conflicting Data

Duong Chi Thang

ABSTRACT

Data fusion is the process of resolving conflicting data from multiple data sources. As the data sources are inherently heterogeneous, there is a need for an expert to resolve the conflicting data. Traditional approach requires the expert to resolve a considerable amount of conflicts in order to acquire a high quality dataset. In this project, we consider how to acquire a high quality dataset while maintaining the expert effort minimal. At first, we achieve this goal by building a model which leverages the provenance of the data in reconciling conflicting data. Secondly, we improve our model by taking the dependency between data sources into account. In the end, we empirically show that our solution can significantly reduce the user effort while it can obtain a high quality dataset in comparison with traditional method.

1. INTRODUCTION

Many data management applications require integrating data from multiple sources, each source provides a set of values. However, different sources may provide conflicting values, some of them are true, the others are false. To provide quality data to the users, it is critical that data integration systems can resolve conflict and discover true values. Typically, we expect there is an authoritative source (an expert) who can provide the true value for all the data. However, having an expert to reconcile all the conflicting data is not feasible for two reasons: (1) the amount of data need reconciling is tremendous, (2) the cost, e.g., salary for the expert, is extremely high if we ask the expert to reconcile all the conflicting data. Therefore, we have to settle for an other option where the expert gives feedbacks for only a part of the data but we still want to resolve conflicting data as much as possible. In this report, we consider the following problem: can we devise a reconciliation process such that given a limited user interaction, we can get the maximal quality of the data possible ?

Our solution is based on two observations. Our first observation is that reliable data come from trustworthy data sources and trustworthy data sources provide reliable data. More precisely, if we know which data sources are trustworthy, we are able to identify reliable data and if we have reliable data, we can detect the trust-

worthy data sources. Our approach follows this observation where we elicit expert assertion to get reliable data. Then, our proposed model updates the data source reliability accordingly from these reliable data. Eventually, with enough user feedback, our model is able to distinguish trustworthy and untrustworthy data sources. Therefore, the quality of the dataset can be improved if we discard data from untrustworthy sources. In other words, by taking the data source reliability into account, we are able to detect more reliable data using a small amount of user feedbacks.

Our second observation is that a dataset may contain data sources that copy from each other. Copy data sources are problematic. For example, if the copy sources do not update its data regularly, its data become obsolete or incorrect. Moreover, if a data source copies an incorrect value, it is more critical to the system as the incorrect values may become to dominate the correct ones. Therefore, if we can exclude these copiers early, we can save user effort while acquire a high quality dataset.

The rest of this report is structured as follows. We formally define the problem in section ???. Then, we propose our solution to the problem in Section 3. Section 4 describes our solution when there are dependency between the data sources. Our experimental evaluation is presented in Section 5. Finally, we discuss some related work in Section ??? and conclude our report in Section ???.

2. MODEL AND PROBLEM DEFINITION

In this section, we first introduce our model of the massive data collection. Then, we give a formal definition of the problem we want to solve.

2.1 Massive data collection

We consider a set of n data items and a set of m data sources that provides values for the data items. We denote $D = \{T_1, T_2, \dots, T_n\}$ to be a set of data items and each data item T_i is a discrete variable with values in $C = \{c_1, c_2, \dots, c_t\}$. Each data item T_i has a correct value $g(T_i) \in C$ that is unknown to us. We denote $M = \{m_{ij}\}$ to be the set of values provided by the set of data sources R where $m_{ij} \in \{\emptyset, c_1, c_2, \dots, c_t\}$ is the value assigned to T_i by data source S_j . In other words, a data source may not assign values to all data items.

Combining all the above notions, we define a massive data collection to be a triple $E = \langle R, D, M \rangle$ where R is a set of data source, D a set of data items and M the set of values provided by the data sources.

2.2 Selective instance

The data collection has a set of instances $\Omega = \{I_i\}$ where an instance I_i is a set of n assignments. Each assignment $\langle T, v \rangle$ or $T = v$ is a tuple of a data item T and a value $v \in C$. Among

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...\$15.00.

the instances, there is a special instance, the ground truth instance $G = \{(T_i, g(T_i))\}$, which is the set of correct assignments for all data items. During the reconciliation process, we maintain a selective instance I_r that we consider it approximates the ground truth G the best. To measure how good an instance approximates the ground truth G , we define the precision value P_i of an instance I_i as follows:

$$P_i = \frac{|I_i \cap G|}{|I_i|} \quad (1)$$

2.3 Problem statement

As mentioned in Section 1, given the heterogeneity of the data sources, we need external knowledge from an expert to reconcile conflicting data. There is a tradeoff between the expert effort (i.e., the number of feedbacks) and the data collection quality (i.e., the precision of the selective instance). This tradeoff is depicted in Figure 1 by the lower curve in which the expert gives feedbacks for the data items randomly and the selective instance is generated by majority voting. In this report, we want to heighten this curve as much as possible. In other words, given an upper limit of expert interaction (e.g., 10% of data items), we aim to improve the precision of the selective instance as much as possible. Formally, the problem

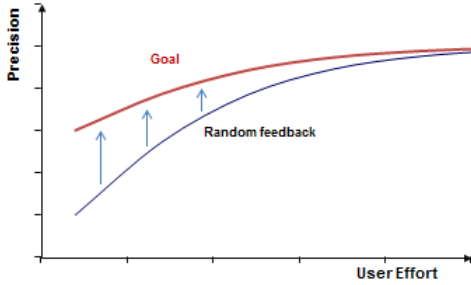


Figure 1: Optimization goal

we want to solve is as follows:

PROBLEM 1. Given a data collection $\langle R, D \rangle$ and a predefined number of user interactions N , find an instance I that has the maximal precision.

The above problem is basically about select a set of N user feedbacks F and build an instance I from the triple $\langle R, D, F \rangle$ such that I has the maximal precision possible.

3. PROVENANCE-BASED RECONCILIATION

In order to solve Problem 1, we follow an iterative approach where in each iteration, we select a data item that provides the most benefit in reconciling conflicting data in the massive data collection. The iterative approach has a key advantage that we can leverage the information provided by previous feedbacks to select the most useful data item to feedback next. The reconciliation process as shown in Algorithm 1 takes a data collection as input and iteratively improve its selective instance I_i through user feedback and feedback propagation. Finally, it returns an instance I_r that approximates G the best. In each iteration, we do the following steps: (1) select a data item $T_i \in D$, (2) elicit user input on the data item T_i (select the correct value for this data item), and (3) propagate the user feedback and update the selected instance I .

Initially (line 1), we initialize the set of feedbacked data items F and the counter i . Then, we generate an instance I_0 from the massive data collection $\langle R, D, M \rangle$. Then, we proceed as follows: First, we select a data item T which brings the most benefit in reconciling conflicting data in the data collection from the set of candidate

Algorithm 1: Reconciliation process with EM and IG

```

input : set of data sources  $R$ ,
        set of data items  $D$ ,
        values assigned by data sources  $M$ 
output: a selective set of categories  $I_r$ 

// Initialization
1  $F = \emptyset; i = 0$ 
2  $I_0 = \text{instantiate}(M, F)$ ;
3 while  $i < N$  do
   // In each user interaction step
   // (1) Select a data item
4    $T = \text{select}(D \setminus F)$ ;
   // (2) Elicit user input
5   Elicit user input on  $T$ ;
   // (3) Integrate the feedback
6    $F = F \cup \{T\}$ ;
7    $I_{i+1} = \text{instantiate}(M, F)$ ;
8    $i = i + 1$ ;
9 return  $I_i$ 

```

data items (line 4). Here, all data items for which we already have feedbacked (represented by F) are neglected. Second, we elicit user input for this data item (line 5). Then, we integrate the feedback by updating the set of feedbacked data items F (line 6), and instantiate a new instance I_{i+1} from the data collection leveraging the feedbacks F . The reconciliation process stops when we use all the budget of user interactions, i.e., $i = N$.

The reconciliation process has two main methods *instantiate* and *select*. In the following section, we first describe the *instantiate* method for instantiating a selective set using the expectation-maximization algorithm. Then, we discuss the *select* function which ranks data items based on information gain.

3.1 Generating the Selective Instance

Since data are collected from various data sources and the reliability of each data source is unknown, given the values assigned to a data item by multiple data sources, it is difficult to know which data source to trust or which value we should choose as the correct one. Therefore, we handle this uncertainty in reconciling conflicting data based on a probabilistic model. For each data item, we assign a probability of being correct for each possible value. Moreover, each data source is also assigned a reliability value. Based on the above ideas, we define a *probabilistic data collection* as follows:

DEFINITION 1. Let $\{T_1, \dots, T_n\}$ be a set of data items. Let $C = \{c_1, \dots, c_t\}$ be a set of values. A *probabilistic data collection* is a tuple $\langle U, V \rangle$ such that

$$U = \{\langle T_1, \{Pr_1(c_1), \dots, Pr_1(c_t)\} \rangle, \dots, \langle T_n, \{Pr_n(c_1), \dots, Pr_n(c_t)\} \rangle\}$$

$$V = \{\langle S_1, Pr(S_1) \rangle, \dots, \langle S_m, Pr(S_m) \rangle\}$$

where

- $Pr_i(c_j) \in [0, 1]$, and $\sum_{j=1}^t Pr_i(c_j) = 1$
- $Pr(S_i) \in [0, 1]$

We will describe how we assign probability based on the following example:

EXAMPLE 1. Table 1 shows the open prices of two stocks provided by 2 different data sources. Among the values provided by two sources for two stocks, we do not know which one is correct. Therefore, we follow the maximum entropy approach by assigning each possible value of AAPL and MSFT a probability of 0.5, i.e.,

$Pr(AAPL = 335.95) = Pr(AAPL = 335.94) = 0.5$ and $Pr(MSFT = 25.93) = Pr(MSFT = 25.95) = 0.5$. If we have a feedback on the AAPL stock saying the open price of AAPL is 335.95, we may need to update the correctness probabilities of the values of the MSFT stock. There are two approaches to do this:

- Without considering the provenance of the data: if we do not take into account the data sources, the feedback on AAPL does not affect MSFT since we see them as isolated. Therefore, the probabilities remain the same: $Pr(MSFT = 25.93) = Pr(MSFT = 25.95) = 0.5$
- Considering the provenance of the data: Since AAPL = 335.95 confirms that the value assigned by data source S_1 on AAPL is correct and by S_2 is incorrect, it is reasonable for us to trust data source S_1 more than S_2 . In other words, S_1 should have higher trustworthiness value than S_2 . Since S_2 is less reliable, the value it assigns to data item MSFT is more likely to be incorrect. Therefore, we should decrease the probability of MSFT = 25.95 to reflect this observation. In other words, we may update the probabilities as follows: $Pr(MSFT = 25.93) = 0.75, Pr(MSFT = 25.95) = 0.25$.

In order to assign the probabilities to the data items and the data sources, we consider these sources of probabilities:

- Data source provenance: since data come from different sources and the reliability of data sources are different, there is a tendency that data from reliable sources are more reliable and vice versa. Therefore, data from reliable sources should have a higher probability to be correct.
- User inputs: a user input reduces the uncertainty of not only the validated data item T but also other data items correlated to T through the same data source. Consequently, a data item consistent with user inputs is likely to be correct; whereas a data item inconsistent with user input is likely to be incorrect.

3.1.1 A quick reminder on Expectation-Maximization algorithm

Once we know the correct value of each data item, we can measure the data source reliability. However, since we do not know the correct values for all data items beside the values feedbacked by the user, we need to estimate the ground truth. Nevertheless, estimating the correct values requires knowing the reliability of the data sources, whereas computing the reliability of the data sources requires knowing of the ground truth. There is an inter-dependence between them and we solve the problem by concurrently estimating ground truth and source reliability. Therefore, the Expectation-Maximization algorithm is employed for its ability to simultaneously calculate the source reliability and estimate the ground truth. Given enough feedbacks, the reconciliation process can distinguish between reliable data sources and unreliable ones.

We take the same approach as [1] and model the trustworthiness of a data source S_i indirectly through a $t \times t$ latent confusion matrix F_i where $t = |C|$. Each row refers to the correct value and each column refers to a value assigned by a data source. Therefore, we have m confusion matrices for m data sources. A cell f_{kl}^w in the

	S_1	S_2
AAPL	335.95	335.94
MSFT	25.93	25.95

Table 1: The motivating example with two data sources provide information about OpenPrice of two stock symbols

confusion matrix of data source S_w represents the probability that data source S_w assigns category c_l to a data item given c_k is the correct category. f_{kl}^w can be estimated as follows:

$$f_{kl}^w = \frac{\sum_{i=1}^n \mathbb{1}_{m_{i,w}=c_l \wedge g(T_i)=c_k}}{\sum_{i=1}^n \mathbb{1}_{m_{i,w} \neq \emptyset \wedge g(T_i)=c_k}}$$

where $\mathbb{1}_{cond}$ equals 1 if $cond$ equals *true* and 0 otherwise and $\sum_{l=1}^t f_{kl}^w = 1 (\forall k \in [1..t], j \in [1..m])$.

From the confusion matrix, we can estimate the reliability of data source S_i by $Pr(S_i) = \frac{\sum_{j=1}^t f_{ij}^i}{\sum_{j,k} f_{jk}^i}$.

Let n_{il}^j be the number of times a data source S_j assigns value c_l to data item T_i . Since a data source only assigns value to a data item at most once, n_{il}^j can only takes two values 0 or 1. Consider a data item T_i , recall that f_{gl}^j is the probability of data source S_j assign the value c_l given c_g is the correct category for T_i . Then, the probability of doing this n_{il}^j times is $(f_{gl}^j)^{n_{il}^j}$. Therefore, the number of times a data source assigns a category c_1, c_2, \dots, c_t to a data item T_i given c_g is the correct value for T_i is distributed according to a multinomial distribution and its likelihood is proportional to:

$$Pr(n_{i1}^j, \dots, n_{it}^j; f_{g1}^j, \dots, f_{gt}^j | g(T_i) = c_g) \propto \prod_{l=1}^t (f_{gl}^j)^{n_{il}^j} \quad (2)$$

Since we assume that m data sources assign value independently to the data items, the likelihood of categories provided for data item T_i when c_g is correct is proportional to

$$\prod_{j=1}^m Pr(n_{i1}^j, \dots, n_{it}^j; f_{g1}^j, \dots, f_{gt}^j | g(T_i) = c_g) \propto \prod_{j=1}^m \prod_{l=1}^t (f_{gl}^j)^{n_{il}^j} \quad (3)$$

Since we do not know the ground truth, we compute the expectation of $Pr(n_{i1}^j, \dots, n_{it}^j; f_{g1}^j, \dots, f_{gt}^j)$ over all possible categories, i.e., we compute the marginal probability over all possible categories

$$\sum_{k=1}^t p_k \prod_{j=1}^m \prod_{l=1}^t (f_{gl}^j)^{n_{il}^j} \quad (4)$$

We also assume that all the data items are independent, the joint probability distribution over all n data items is

$$\prod_{i=1}^n \left(\sum_{k=1}^t p_k \prod_{j=1}^m \prod_{l=1}^t (f_{gl}^j)^{n_{il}^j} \right) \quad (5)$$

Equation 5 contains many multinomial distributions. In order to estimate the prior probability p_k of category c_k , the confusion matrix f_{gl}^j and the correct value for each data item g_i , we apply the expectation maximization (EM) algorithm. The algorithm has two steps: (1) the parameter estimation step where we estimate the parameters p_k, f_{gl}^j and g_i , (2) the maximization step where we maximize the likelihood function based on the estimated parameter. The EM algorithm is described in Algorithm 2.

The EM algorithm takes a data collection and the set of feedbacked data items at the moment as input and returns a probabilistic data collection. Initially, we estimate the ground truth of the set

of data items which have not been feedbacked randomly. For the feedbacked data items, the correct values are the values feedbacked by user. We combine these two sets to get the initial estimated ground truth \widehat{G} (line 1). Then, we proceed as follows. In the first step, given the new estimated ground truth, we compute the confusion matrix of each worker and the prior probability distribution of the values p_k . We also calculate the reliability of the workers and store them in V (line 7-8). In the second step, for each data item T_i and each category c_g , we calculate its probability of being correct $Pr(g(T_i) = c_g)$ using the confusion matrices and the prior probability distribution p_k (line 14). The probabilities of the data items U are updated accordingly (line 16, 26). For feedbacked data items, the correct value has probability 1 and the others have probability 0 (line 20-25) Then, we reestimate the ground truth \widehat{G} using the newly calculated probabilities. For each data item, the value with the highest probability is considered correct (line 27). We repeat these steps until the results converge. After the results converge, the probabilistic data collection $\langle U, V \rangle$ is returned.

3.1.2 Instantiating with Expectation-Maximization algorithm

In the previous section, we discuss the EM algorithm and how

it can concurrently estimating ground truth and the accuracy of the data sources. The output of the EM algorithm is a probabilistic massive data collection $\langle U, V \rangle$ such that each value of a data item and each data source is assigned a probability. From the probabilistic massive data collection $\langle U, V \rangle$, we can generate the selective instance as follows

$$I = \{\langle T_i, v_i \rangle \mid \forall v_j \in C \wedge v_j \neq v_i : Pr_i(v_i) > Pr_i(v_j)\} \quad (6)$$

Informally, the selective instance contains assignments of values which have the highest probability of being correct for each data item.

3.2 Ranking data items

In this section, we introduce the key concepts of our ordering approach—which ranks and displays data items. The system interacts with the user to get feedback on suggested data items. The task of ordering strategies is to devise how to best present the data items to the user, in a way that will provide the *most benefit* for improving the quality of the data repository. To this end, we apply the concept of information gain from information theory to choose a ranking in a principled manner.

Algorithm 2: Expectation maximization

```

input : a set of data sources  $R$ ,
        a set of data items  $D$ ,
        a set of feedbacked data items  $F$ .
output: a probabilistic data collection,  $\langle U, V \rangle$ 
// Initialization
1 Choose  $\widehat{G}$  randomly
2 while not converge do
3    $U = \emptyset, V = \emptyset$ 
   // (1) Estimate the confusion matrices and
   // category prior probability
4   for  $S_w \in R$  do
5     for  $c_k, c_l \in C$  do
6        $\widehat{f}_{kl}^w = \frac{\sum_{i=1}^n \mathbb{1}_{m_{iw}=c_l \wedge g(T_i)=c_k}}{\sum_{i=1}^n \mathbb{1}_{m_{iw} \neq \emptyset \wedge g(T_i)=c_k}}$ ;
7        $Pr(S_w) = \frac{\sum_{j=1}^t f_{jk}^w}{\sum_{j,k} f_{jk}^w}$ ;
8        $V = V \cup \{\langle S_w, Pr(S_w) \rangle\}$ ;
9   for  $c_k \in C$  do
10     $\widehat{p}_k = \frac{\sum_{i=1}^n \mathbb{1}_{g(T_i)=c_k}}{n}$ ;
   // (2) Re-calculate probabilities
11  for  $T_i \in D \setminus F$  do
12     $B_i = \emptyset$ ;
13    for  $g \in [1, t]$  do
14       $Pr(g(T_i) = c_g) = \frac{p_g \prod_{j=1}^m \prod_{l=1}^t (f_{gl}^j)^{n_{il}^j}}{\sum_{k=0}^t p_k \prod_{j=1}^m \prod_{l=0}^t (f_{kl}^j)^{(n_{il}^j)}}$ ;
15       $B_i = B_i \cup \{Pr(g(T_i) = c_g)\}$ ;
16     $U = U \cup \{\langle T_i, B_i \rangle\}$ ;
17  for  $T_i \in F$  do
18     $B_i = \emptyset$ ;
19    for  $g \in [1, t]$  do
20      if  $g(T_i) = c_g$  then
21         $Pr(g(T_i)) = c_g = 1$ ;
22         $B_i = B_i \cup \{Pr(g(T_i) = c_g)\}$ ;
23      else
24         $Pr(g(T_i)) = c_g = 0$ ;
25         $B_i = B_i \cup \{Pr(g(T_i) = c_g)\}$ ;
26     $U = U \cup \{\langle T_i, B_i \rangle\}$ ;
   // (3) Re-estimate  $\widehat{G}$ 
27   $\widehat{G} = \{\langle T_i, c_i \rangle \mid Pr(g(T_i) = c_i) > Pr(g(T_i) = c_j), \forall c_j \neq c_i\}$ 
28 return  $U, V$ 

```

We measure the uncertainty of the probabilistic data collection using Shannon entropy. First, we define the entropy of a data item:

$$H(T_i) = - \sum_{c \in \mathcal{C}} Pr_i(c) \times \log(Pr_i(c))$$

From the entropy of a data item, we can model the uncertainty of a probabilistic data collection $\langle U, V \rangle$ as follows:

$$H(\langle U, V \rangle) = \sum_{T_i \in D} H(T_i)$$

To acquire a maximal precision within a provided budget, we focus on heuristic strategies that exploit a ranking of data items for which user inputs shall be elicited. We design the selection function such that the elicited user inputs on the chosen data item could contribute the most in reducing the uncertainty. We measure the contribution of a user feedback on a data item by information gain. In order to introduce the notion of information gain, we need to define a conditional entropy measure. The conditional entropy measures the entropy of the probabilistic data collection conditioned on the user feedback feedbacks on data item T_i is:

$$H(\langle U, V \rangle \mid T_i) = \sum_{j \in [1..t]} q_{ij} \times H(I \mid T_i = c_j) \quad (7)$$

Equation 7 measures the expected entropy of the probabilistic data repository when the user asserts that c_j is the correct value for data item T_i . To make a decision on which data item to forward first to the user, we compare the uncertainty before and after the user give inputs on it. We can now define the information gain for a data item $T \in D$ by the change in the entropy. Thus the information gain score if user feedback on data item T is computed as:

$$IG(T) = H(I) - H(I \mid T). \quad (8)$$

In fact, information gain is a mean of quantifying the potential benefit of knowing the true value of an unknown object [18]. More precisely, information gain measures the (expected) amount of uncertainty reduction. Therefore, we suggest the data item that reduce the uncertainty most. In other words, the selected data item has highest information gain, i.e.,

$$T = \arg \max_{T_i \in D} IG(T_i)$$

4. RECONCILIATION WITH DEPENDENT DATA SOURCES

In the previous section, we discuss our solution to the maximal quality problem using information gain ranking and expectation maximization algorithm. However, the previous solution does not perform well if there are dependency between the data sources as shown in Section *experiment*. Therefore, we need a mechanism to remove these dependent data sources as early as possible. Removing dependent data sources also helps minimizing user efforts since the dependent data sources to be removed tend to contain incorrect or obsolete data.

As we want to remove dependent data sources early, the reconciliation process with dependency detection has two phases. In the first phase, we focus on detecting and removing as many dependent sources as possible and in the second phase, we focus on reducing the uncertainty of the data collection. There may be some dependent data sources that are not removed in the first phase that we need to remove them in the second phase as we get more feedbacks. The reconciliation process with dependency detection is

shown in Algorithm 3. Beside the data collection $\langle R, D, M \rangle$, it takes a threshold l (the number of user interactions that are used for detecting dependent sources) and a dependent threshold t_d as input. The reconciliation process with dependency detection iteratively improves its selective instance I through user feedback, dependency detection and feedback propagation. Finally, it returns an instance I_r that approximates G the best. In each iteration, we do the following steps: (1) select a data item $T_i \in D$, (2) elicit user input on data item T_i , (3) detect and remove dependent data sources and (4) propagate the user feedback and update the selected instance I .

Algorithm 3: Reconciliation process with dependency detection

```

input : set of data sources  $R$ ,
        set of data items  $D$ ,
        values assigned by data sources  $M$ ,
        a threshold  $l$ ,
        a dependent threshold  $t_d$ .
output: a selective set of categories  $I_r$ 
// Initialization
1  $F = \emptyset; i = 0$ 
2  $I_0 = \text{instantiate}(M, F)$ ;
3 while  $i < N$  do
    // In each user interaction step
    // (1) Select a data item
4   if  $i < l$  then
5      $T = \text{select\_depend}(D \setminus F)$ 
6   else
7      $T = \text{argmax}_{T \in D \setminus F} IG(T)$ 
    // (2) Elicit user input
8   Elicit user input on  $T$ ;
    // (3) Handling dependent sources
    // (3.1) Detect dependent sources
9    $R' = \text{detect}(R, D, F, t_d)$ ;
    // (3.2) Remove dependent sources
10   $\text{remove\_dependent}(R')$ ;
    // (4) Integrate the feedback
11   $F = F \cup \{T\}$ ;
12   $I_{i+1} = \text{instantiate}(M, F)$ ;
13   $i = i + 1$ ;
14 return  $I_i$ 

```

Initially (line 1), we initialize the set of feedbacked data items F and the counter i . Then, we generate an instance I_0 from the massive data collection $\langle R, D, M \rangle$. Then, we proceed as follows: For the first l iterations, we apply the data item ordering method *select_depend* which selects a data item T which provides the most benefit in detecting dependent sources (line 5). After l first iterations, we apply the information gain ordering as described in Section 3.2 (line 7). Here, all data items for which we already have feedbacked (represented by F) are neglected. Second, we elicit user input for this data item (line 8). Then, we detect the dependent sources in line 9 and remove them in line 10. Finally, we integrate the feedback by updating the set of feedbacked data items F (line 11), and instantiate a new instance I_{i+1} from the updated data collection leveraging the feedbacks F . The reconciliation process stops when we use all the budget of user interactions, i.e., $i = N$.

The reconciliation process with dependency detection has four main methods: *instantiate*, *select_depend*, *detect* and *remove_dependent*. Beside the *instantiate* method which was discussed in Section 3.1.2, we will discuss the other methods in the following section. First, we describe the *detect* method in which we use to detect the dependent sources. Then, we explain the *select_depend* function which orders the data items in a way that helps the most in detecting dependent sources. Finally, we show how we handle the dependent

data sources (the *remove_dependent* function).

4.1 Detecting data source dependency

In order to detect dependence between data sources, we apply the method mentioned in [3]. We assume that the data collection contains two types of data sources: independent sources and copiers. We denote $S_1 \sim S_2$ the dependency between two data sources S_1 and S_2 . In detecting dependence between data source S_1 and S_2 , we are interested in three sets of data items \overline{D}_t , the set of data items that S_1 and S_2 give the same correct category, \overline{D}_f , the set of data items that S_1 and S_2 give the same incorrect category, \overline{D}_d , the set of data items that S_1 and S_2 give different categories. We denote d_t, d_f, d_d the size of $\overline{D}_t, \overline{D}_f, \overline{D}_d$ respectively.

In order to calculate d_t, d_f, d_d , we need to know the ground truth of the data items. However, the only source of ground truth available is the feedbacks provided by the expert. Therefore, in each iteration, we calculate d_t, d_f, d_d based on the feedbacks provided by the user, i.e., $\overline{D}_t, \overline{D}_f, \overline{D}_d \subseteq F$ where F is the number of feedbacked data items at an iteration. The conditional probability that S_1 and S_2 are dependent given the observation of the data Δ is

$$Pr(S_1 \sim S_2 | \Delta) = \frac{1}{1 + \frac{1-\alpha}{\alpha} \left(\frac{1-\epsilon}{1-\epsilon+c\epsilon}\right)^{d_t} \left(\frac{\epsilon}{c(t-1)+\epsilon-c\epsilon}\right)^{d_f} \left(\frac{1}{1-c}\right)^{d_d}} \quad (9)$$

In Equation 9, we denote $\alpha = Pr(S_1 \sim S_2)$ to be the prior probability that two data sources are dependent, $c(0 < c \leq 1)$, the probability that a value provided by a copier is copied; and $\epsilon(0 \leq \epsilon < \frac{t-1}{t})$, the probability that an independently provided value is false.

For a set of data sources, we can detect dependent sources by calculating the probability in Equation 9 and set an accept threshold t_d for this probability. If the conditional probability that two data sources S_1 and S_2 are dependent is higher than t_d , they are considered dependent, otherwise, we consider them as independent sources.

4.2 Handling dependent sources

In the previous section, we describe the *detect* method which we use to detect the dependence between data sources. In this section, we first discuss our method of ordering the data items (the *select_depend* method). Then, we show how we handle the dependent data sources (the *remove_dependent* method).

4.2.1 Ranking data items for detecting source dependency

In this section, we discuss a data item ordering strategy where data items which contribute the most to detect dependent sources are feedbacked earlier. We apply a greedy ordering approach where in each iteration, we select the data item with the maximum expectation of common incorrect categories. The intuition behind this heuristic is that copy data sources tend to have more number of common false values and eliciting feedback on data items that have a larger number of common incorrect categories d_f is particularly beneficial for detecting dependent sources. Formally, we define the expectation of common incorrect categories of a data item T_i as follows:

$$E(T_i) = \sum_{c \in C} Pr_i(c) \sum_{S_1, S_2, S_1 \neq S_2} |\overline{D}_f^{S_1, S_2}| \quad (10)$$

Then, we select the data item with the highest expected number

of common incorrect categories, i.e.,

$$T = \arg \max_{T_i \in D} E(T_i)$$

4.2.2 Removing dependent sources

Since Equation 9 does not indicate the direction of dependence, after we detect two dependent sources, we may not know which data source to remove. Therefore, we apply a heuristic to select data source to delete. Since a copy source tends to be less reliable than an independent one, we remove data source with a lower trustworthiness value from a pair of dependent ones. In other words, if $Pr(S_1 \sim S_2 | \Delta) > t_d$ and $Pr(S_1) > Pr(S_2)$ then we remove data source S_2 from the data collection.

5. EXPERIMENTAL RESULTS

In this section, we now empirically show the effect of minimizing user efforts and data source dependence detection on many datasets. We tested the user effort minimization process on a real-world dataset and a synthetic one. However, since we want to test source dependence detection in numerous conditions which are not easily found in real data, we only experimented with synthetic data for the dependence detection part. Moreover, in real-world dataset, we do not know which sources copy from others since a data source never exposes the information whether it copies from others or not.

5.1 Experimental settings

5.1.1 Datasets

Our experiments will be conducted on two types of data: real data and synthetic data. While the real data provide a pragmatic view on real-world scenario, the synthetic data help to evaluate the performance with different settings.

- Real data¹: the dataset was extracted from 50 data sources in the Stock domain on 01-07-2011. We focus on the NASDAQ100 symbols which are the major stocks. For the purposes of evaluation, we use the gold standard provided by Nasdaq.com on the 100 symbols in the NASDAQ100 index. Among many attributes from the Stock domain, we focus on one attribute *Open price*. We use the ground truth to simulate expert's assertions. Among 50 data sources, we select 5 data sources that best represent the dataset.
- Synthetic data: we generate a synthetic dataset with 15 independent data sources providing values for 50 data items. From the independent data sources, we generate 11 copiers that copy from the same independent data source. For each value from the original source, the copiers may change it to another value with a probability of 0.4.

All experiments ran on an Intel Core i7 system (2.8GHz, 8GB RAM).

5.1.2 Metrics

Beside the precision metric described in Section ??, we measured the quality improvements achieved by reconciliation and the required human efforts as follows:

- **User effort**: is measured in terms of feedback steps. Since a user examines one data item at a time, the number of feedback steps is the number of asserted data items.

¹We thank authors of [11] for providing us the data set

- **Percentage of Precision Increased:** measure the relative quality improvement. If the precision of the data collection at an iteration is p and the initial precision is p_0 then the percentage of precision increased is

$$\frac{p - p_0}{1 - p_0} \times 100$$

5.2 Computational time

User participation is not only limited in cost but also in time. An expert can not wait for hours or days for the system to select a data item for her elicitation. As a result, the computational time to select a data item is an important aspect. In this experiment, we study the average response time of the system to select a data item with two different parameters: the number of values t and the number of data items n in the data collection. In each setting, we measure the average running time over 100 runs.

Table 2 shows the result of this experiment. A significant observation is that when the number of data items doubles, the running time increases nearly 8 times. This is because we need to inspect all data items to calculate the conditional entropy for each of them in order to select the one with the most information gain. Another observation is that the running time increases 4 times when the number of values doubles. The reason is that for each data item, we need to calculate the entropy of the data collection for each possible value. However, for a data collection with 100 data items, 4 possible values, taking 2 minutes to select a data item is reasonable since user elicitation is a one-time process.

		n	
		50	100
t	2	144538	1031567
	4	550740	4703390

Table 2: Effects of number of values and data items on running time (ms)

5.3 Evaluations on User Effort Maximization

In this set of experiments, we study the efficiency of our solution to the maximal quality problem. More precisely, we study the improvements in precision (Y-axis) with increased feedback percentage (X-axis, out of total number of data items) using two strategies: (1) Baseline: feedback on data item in random order, instantiate using majority voting, (2) IGEM: selection of data item based on information gain and instantiate using EM algorithm. In that, we first analyze the effects of data source reliability on the performance of two strategies. Based on the results, we show that our solution using expectation maximization algorithm and information gain ordering outperforms the baseline. Secondly, we evaluate the effects of data source reliability variance on our solution. From the experimental results, we show that our solution performs well if the reliability of the data sources vary considerably.

5.3.1 Effects of data source reliability

In this experiment, we study the effects of data source reliability on the efficiency of the IGEM model. More precisely, we generate a dataset where the data source reliability follow a normal distribution $N(\mu, 0.1)$. Then, we vary the mean μ to see its effects on the precision. Expert validation is simulated using the generated ground truth (constructed together with the dataset).

Figure 5.1.2 depicts the result in average over 50 experiment runs. This result shows a significant improvement in precision for

the IGEM strategy with respect to the baseline. For example, if the source reliability mean is 0.5, to reach a precision of 0.8, the IGEM strategy needs only 30% of user interactions while the BL takes 60%, saving about 30% of the user effort. Or equivalently, if the source reliability mean is 0.6, the IGEM strategy acquires the precision of 0.9 for only 20% of user interactions while the baseline requires 60% of user interaction.

5.3.2 Effects of data source reliability variance

In this experiment, we study the effects of the data source reliability variance. Our hypothesis is that if the reliability of the data sources vary considerably, the IGEM strategy works extremely well. Following the previous experiment, we also generate a dataset where data source reliability follow a normal distribution $N(0.5, \sigma)$. Then, we vary the standard deviation σ to see its effects on the performance of the IGEM strategy. Expert validation is also simulated using the generated ground truth.

Figure 4 depicts the result in average over 50 experiment runs. We only keep the IGEM curves in Figure 4 since all the baseline curves are below them. The result confirms our hypothesis that the variance of the data source reliability have a positive effect on user effort minimization. For example, to reach a precision of 0.9, we need 60% of user interactions if σ is 0.1 but we only need 50% of user interaction if σ is 0.15. The reason is that when the reliability of the data sources vary considerably, given enough feedbacks, the EM algorithm eventually distinguishes reliable data sources from unreliable ones. Hence, we can correctly identify the correct values since they are given by reliable sources.

5.4 Evaluations on Dependency Detection

In this set of experiments, we study the efficiency of our dependency detection method (the DEPEND strategy) and its effect on maximizing the data collection quality. In these experiments, we want to simulate the following scenario: the data collection contains some good data sources and some malicious ones (a data source is malicious if its reliability is below 0.5). To make it worse, the data collection also contains some copiers and the copiers copy from the malicious source. We focus on this scenario for its worst-case nature. Therefore, if our solution can cope with this scenario, it can work well with the other cases. In order to detect the dependent sources, we devote 10% of user feedbacks in both of the experiments.

In the following sections, we test the robustness of our solution to the malicious sources on two aspects: (1) the maliciousness of the malicious data sources, i.e., the reliability of the malicious sources, (2) the number of malicious data sources in the data collection. From the results, we show that our solution is robust to malicious sources in both aspects.

5.4.1 Effects of malicious data source reliability

In this experiment, we generate two clusters of 15 data sources: a cluster of good data sources which data source reliability follow a normal distribution $N(0.8, 0.01)$ and a cluster of malicious data sources which reliability follow a normal distribution $N(\mu, 0.01)$. Then, we generate the copiers from one of the malicious source. In this experiment, the number of reliable sources and malicious sources are equal. We examine the effects of varying μ on the reconciliation process.

The result of this experiment is shown in Figure 5.3. A noticeable observation is that the DEPEND strategy dramatically increases the precision of the selective instance despite the malicious source reliability and it outperforms the baseline strategy during the reconciliation process. We observe a surf in precision for the

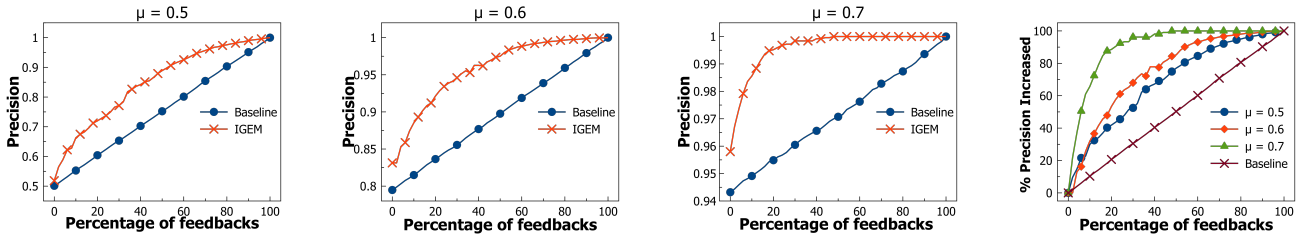


Figure 2: Effects of data sources reliability

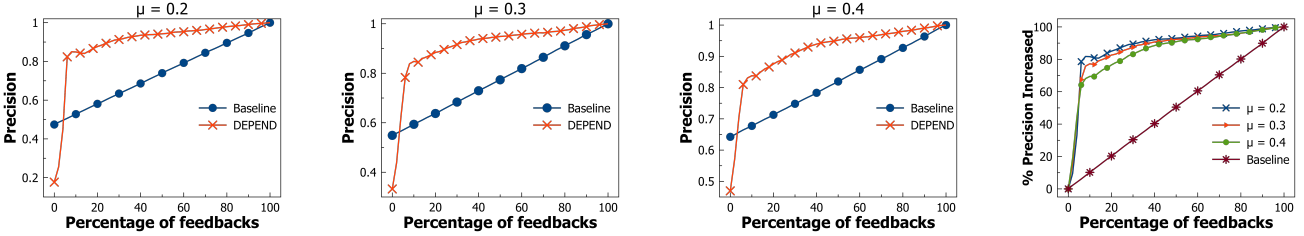


Figure 3: Effects of malicious data sources reliability

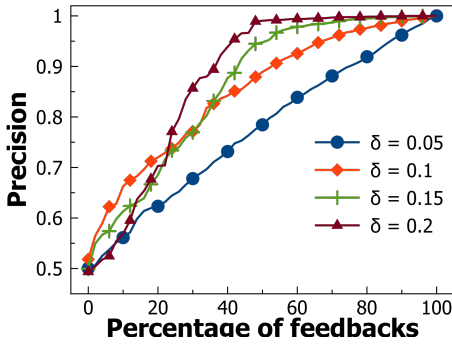


Figure 4: Effect of data source reliability variance on User Effort Minimization

DEPEND strategy in the beginning of the reconciliation process. In other words, we are able to get a very high precision (above 0.8) for only 10% of user interaction. This is expected as we devote the first 10% of user interaction for dependency detection. By removing the dependent sources, we are able to get a higher quality data collection. An interesting observation is that when μ increases, the amount of precision we can improve with the same amount of feedbacks decreases. The reason is that as we increase μ , the variance of the reliability between the data sources in the data collection decreases. However, as we show in previous experiment, our solution performs better if the variance between the data sources is high.

5.4.2 Effects of malicious data source quantity

In this experiment, we generate two clusters of 15 data sources: a cluster of good data sources which data source reliability follow a normal distribution $N(0.8, 0.01)$ and a cluster of malicious data sources which reliability follow a normal distribution $N(0.2, 0.01)$. Then, we generate the copiers from one of the malicious source. We vary the ratio between reliable and malicious sources to find out its effects on the reconciliation process.

Figure 5.4 shows the result of this experiment. An interesting observation is that in all cases, there is a slight decrease in precision after we use 10% of user interaction. The reason is that we may

remove some reliable sources in the dependency detection phase and it affects the precision when we move on to the second phase. Another interesting observation is that given the same amount of feedbacks, we can improve the precision the most when the number of reliable and malicious sources are equal. The phenomenon can be explained as follows: when most of the sources are reliable (75% are reliable), we are more likely to incorrectly remove reliable data sources. However, when most of the sources are malicious, the overall quality of the data collection is low, which means we need more feedbacks to get the same amount of precision. However, it can be clearly seen that the DEPEND method significantly outperforms the baseline strategy in all cases. For example, when 75% of the data sources is malicious, we only need about 10% of user interaction to get a precision of 0.6 while the baseline strategy need over 50% of user interaction. In other words, the DEPEND strategy is robust to the increased number of malicious data sources in the data collection.

6. RELATED WORKS

We now review salient work in user effort minimization and source dependency detection that are related to our research.

Data fusion The goal of data fusion is to resolve conflicts in data and acquire the true value [4, 12, 5, 14, 19, 17]. Our algorithm differs from theirs in that their approaches are fully unsupervised while ours is semi-supervised where user also takes a part in resolving conflicts.

Trust management There are many studies on source trustworthiness assessment [13]. The PageRank algorithm and the Authority-Hub model estimate the trustworthiness of a data source based on its connection with other sources. However, to the extent of our knowledge, our approach is the first to leverage user feedback to assess the trustworthiness of the data source.

User Feedback Further, guiding user efforts has been addressed for eliminating violations of integrity constraints [20] and for improving ETL processes. Recent works in crowdsourcing user input for data integration is related to our work [10, 8, 9, 15, 16, 7]. While we assume one input assertion per data item, different approaches have been presented for resolving conflicts among multiple input assertions per data item. There are some papers that use expectation-maximization as an aggregation method such as [6].

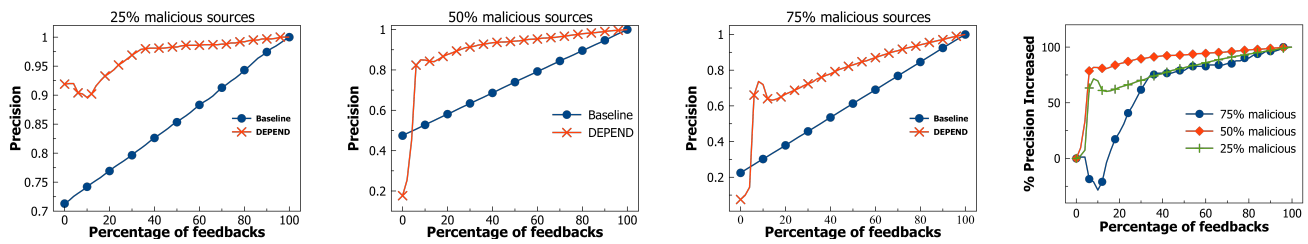


Figure 5: Effects of ratio of malicious data sources

In this paper, the authors argue that the aggregation method using expectation-maximization algorithm dominate the traditional majority voting. Beside our technique of probabilistic aggregation using Expectation-Maximization, there are some works that rely on Conditional Random Fields such as [2]. In this paper, the authors propose an aggregation technique using factor graph. Against this background, it was argued that data integration in the web setting has to follow the pay-as-you-go approach that is evolutionary, reducing uncertainty in a step-wise fashion.

7. CONCLUSION

In this project, we consider how to acquire a high quality dataset while maintaining the expert effort minimal. At first, we achieve this goal by building a model which leverages the provenance of the data in reconciling conflicting data. Secondly, we improve our model by taking the dependency between data sources into account. In the end, we empirically show that our solution can significantly reduce the user effort while it can obtain a high quality dataset in comparison with traditional method.

8. REFERENCES

- [1] A P Dawid and A M Skene, *Maximum likelihood estimation of observer error-rates using the EM algorithm*, J. R. Stat. Soc. (1979), 20–28.
- [2] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux, *Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking*, WWW, 2012, pp. 469–478.
- [3] Xin Luna Dong, Laure Berti-Equille, and Divesh Srivastava, *Integrating conflicting data: The role of source dependence*, PVLDB 2 (2009), no. 1, 550–561.
- [4] Avigdor Gal, Michael Katz, Tomer Sagi, Matthias Weidlich, Karl Aberer, Hung Quoc Viet Nguyen, Zoltán Miklós, Eliezer Levy, and Victor Shafraan, *Completeness and ambiguity of schema cover*, CoopIS, 2013, pp. 241–258.
- [5] Avigdor Gal, Tomer Sagi, Matthias Weidlich, Eliezer Levy, Victor Shafraan, Zoltán Miklós, and Nguyen Quoc Viet Hung, *Making sense of top-k matchings: A unified match graph for schema matching*, 2012, p. 6.
- [6] Mehdi Hosseini, Ingemar J. Cox, Nataša Milić-Frayling, Gabriella Kazai, and Vishwa Vinay, *On aggregating labels from multiple crowd workers to infer relevance of documents*, Proceedings of the 34th European conference on Advances in Information Retrieval (Berlin, Heidelberg), ECIR' 12, Springer-Verlag, 2012, pp. 182–194.
- [7] Nguyen Quoc Viet Hung, Nguyen Thanh Tam, Zoltan Miklos, and Karl Aberer, *On leveraging crowdsourcing techniques for schema matching networks*, DASFAA, 2013, pp. 139–154.
- [8] Nguyen Quoc Viet Hung, Nguyen Thanh Tam, Chau Vinh Tuan, Tri Kurniawan Wijaya, Zoltan Miklos, Karl Aberer, Avigdor Gal, and Matthias Weidlich, *Smart: A tool for analyzing and reconciling schema matching networks*, ICDE, 2015, pp. 1488–1491.
- [9] Nguyen Quoc Viet Hung, Duong Chi Thang, Matthias Weidlich, and Karl Aberer, *Erica: Expert guidance in validating crowd answers*, SIGIR, 2015, pp. 1037–1038.
- [10] Nguyen Quoc Viet Hung, Duong Chi Thang, Matthias Weidlich, and Karl Aberer, *Minimizing efforts in validating crowd answers*, SIGMOD, 2015, pp. 999–1014.
- [11] Xian Li, Xin Luna Dong, Kenneth Lyons, Weiyi Meng, and Divesh Srivastava, *Truth finding on the deep web: is the problem solved?*, Proceedings of the 39th international conference on Very Large Data Bases, PVLDB' 13, VLDB Endowment, 2013, pp. 97–108.
- [12] Hung Quoc Viet Nguyen, Tri Kurniawan Wijaya, Zoltán Miklós, Karl Aberer, Eliezer Levy, Victor Shafraan, Avigdor Gal, and Matthias Weidlich, *Minimizing human effort in reconciling match networks*, ER, 2013, pp. 212–226.
- [13] Quoc Viet Hung Nguyen, Son Thanh Do, Tam Nguyen Thanh, and Karl Aberer, *Privacy-preserving schema reuse*, DASFAA, 2014, pp. 234–250.
- [14] Quoc Viet Hung Nguyen, XuanHoai Luong, Zoltan Miklos, ThoThanh Quan, and Karl Aberer, *Collaborative schema matching reconciliation*, CoopIS, 2013, pp. 222–240.
- [15] Quoc Viet Hung Nguyen, Thanh Tam Nguyen, Ngoc Tran Lam, and Karl Aberer, *Batc: a benchmark for aggregation techniques in crowdsourcing*, SIGIR, 2013, pp. 1079–1080.
- [16] Quoc Viet Hung Nguyen, Tam Nguyen Thanh, Tran Lam Ngoc, and Karl Aberer, *An evaluation of aggregation techniques in crowdsourcing*, WISE, 2013, pp. 1–15.
- [17] Thanh Tam Nguyen, Quoc Viet Hung Nguyen, Matthias Weidlich, and Karl Aberer, *Result selection and summarization for web table search*, ICDE, 2015, pp. 231–242.
- [18] Stuart J. Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, Pearson Education, 2003.
- [19] Nguyen Thanh Tam, Duong Chi Thang, Nguyen Quoc Viet Hung, and Karl Aberer, *An evaluation of diversification techniques*, DASFAA, 2015, pp. 215–231.
- [20] Mohamed Yakout, Ahmed K. Elmagarmid, Jennifer Neville, Mourad Ouzzani, and Ihab F. Ilyas, *Guided data repair*, Proc. VLDB Endow. 4 (2011), no. 5, 279–289.