Staging User Feedback toward Rapid Conflict Resolution in Data Fusion

Romila Pradhan
Purdue University
West Lafayette, IN, USA
rpradhan@cs.purdue.edu

Siarhei Bykau*
Bloomberg L.P.
New York, NY
sbykau@bloomberg.net

Sunil Prabhakar
Purdue University
West Lafayette, IN, USA
sunil@cs.purdue.edu

ABSTRACT

In domains such as the Web, sensor networks and social media, sources often provide conflicting information for the same data item. Several data fusion techniques have been proposed recently to resolve conflicts and identify correct data. The performance of these fusion systems, while quite accurate, is far from perfect. In this paper, we propose to leverage user feedback for validating data conflicts and rapidly improving the performance of fusion. To present the *most beneficial* data items for the user to validate, we take advantage of the level of consensus among sources, and the output of fusion to generate an effective ordering of items. We first evaluate data items individually, and then define a novel decision-theoretic framework based on the concept of value of perfect information (VPI) to order items by their ability to boost the performance of fusion. We further derive approximate formulae to scale up the decision-theoretic framework to large-scale data. We empirically evaluate our algorithms on three real-world datasets with different characteristics, and show that the accuracy of fusion can be significantly improved even while requesting feedback on a few data items. We also show that the performance of the proposed methods depends on the characteristics of data, and assess the trade-off between the amount of feedback acquired, and the effectiveness and efficiency of the methods.

1. INTRODUCTION

With the advent of modern information systems and services, the amount and diversity of data have been growing at an unprecedented pace. Moreover, the number of sources that provide data has significantly increased, spanning well-known sources, such as top news agencies (e.g., CNN, BBC), to individual contributors of Wikipedia articles. Unsurprisingly, conflicts among such data sources arise often, e.g., financial firms publish different stock prices for the same company [21], sensors report conflicting measurements [36], online bookstores list different authors for identical books [41] and so on. Resolving such conflicts is important since inaccurate information may result in unfavorable consequences such as a missed flight or financial losses.

ID	Data Item	S_1	S_2	S_3	S_4
O_1	Zootopia		Howard*	Spencer	Spencer
O_2	Kung Fu Panda	Stevenson*		Nelson	
O_3	Inside Out		leFauve	Docter*	
O_4	Finding Dory				Stanton*
O_5	Minions	Coffin*	Renaud		
O_6	Rio	Jones		Saldanha*	

Table 1: A motivating example. Table shows four sources providing information about directors of six movies. Correct claims are marked with a (*).

Recently, a number of *data fusion* systems have been proposed to discriminate `true` and `false` claims of data items from multiple conflicting data sources (see [22] for a survey). Most of the existing fusion techniques automatically identify correct claims for data items. Although quite accurate, these automated fusion systems are not error-free; incorrect conclusions about the correctness of claims of a data item quickly trickle down to other data items. Particularly for crucial data items where it is imperative to distinguish correct claims from incorrect ones, we cannot solely rely on automated data fusion. Feedback should be integrated in the form of validation from an expert to ensure that the fusion system correctly identifies true claims for most items. Trusted validation of claims is expected to steer the system toward a state of higher efficacy.

1.1 Motivation

Consider an example of websites (sources) providing information on directors of certain animation movies (Table 1). Data fusion systems take the table of conflicting claims as input, and output the correctness of each claim (and, in some cases, the accuracy of each source, i.e., the probability that a claim provided by the source is correct).

Source S_2 provides `Howard` as the director for the movie `Zootopia` whereas sources S_3 and S_4 claim it to be `Spencer`. A data fusion system that outputs `Spencer` to be the true claim of `Zootopia` can benefit from the validation that `Howard` is instead correct. With this knowledge, the fusion system can reconsider the claims provided by sources S_2, S_3 and S_4 and improve its output on other data items.

Validation of claims per se is an expensive task; to guarantee effective conflict resolution, it assumes access to highly accurate feedback (e.g., domain experts). To judiciously uti-

*Work done while at Purdue University.

SIGMOD'17, May 14 - 19, 2017, Raleigh, NC, USA

ISBN 978-1-4503-4197-4/17/05…$15.00

DOI: http://dx.doi.org/10.1145/3035918.3035941

lize the expert, claims should be presented for validation in an order that is most beneficial to the effectiveness of fusion. Assuming we can validate any data item (by asking an expert or using crowdsourcing), and know which of its claims is correct, which item should we select for validation?

The task of identifying the *best* data item for validation is challenging because we have to deal with a number of issues. First, we do not possess ground truth and, therefore, need to develop heuristics to determine the best data item. Second, we need to quantify the definition of 'best' i.e., what is the basis for deciding whether or not one data item is more suitable for validation than another? Third, data fusion typically deals with a large number of claims (hundreds of thousands), thus limiting the ability to ask questions on a very small fraction of all claims. Fourth, since the correctness of each claim may potentially influence the correctness of any other claim, the exhaustive computation of estimating the impact of validating each data item by re-running fusion, is prohibitively expensive. For example, to evaluate data item O_1 for validation, we need to assess its impact on all the $(2+2+2+1+2+2) = 11$ distinct claims of six data items. Similarly checking all data items to select the first item for validation would require $6 * 11 = 66$ computations. Scaling up this procedure to millions of claims is infeasible.

To this end, there are two major observations. First, data items have different *levels of uncertainty* because of the agreement/disagreement of sources on claims. One may expect that validating `"Minions"` would be more advantageous than validating `"Zootopia"` because S_1 and S_2 disagree on `"Minions"` while two of the three sources that vote for `"Zootopia"` agree on a common claim. This is because we expect to learn more from the validation of data items with *disagreement*. Second, although a data item may have conflict over its values, validating it may not be beneficial if it does not influence enough items. For instance, validating `"Finding Dory"` would influence source S_4 and that would have an effect only on `"Zootopia"` whereas validating `"Zootopia"` would impact all other items.

1.2 Problem Formulation

We consider a database instance $\mathcal{D}$, describe the data model of a data fusion system and formulate the problem of ordering user feedback for effective conflict resolution in data fusion. We present the terminology and notations used in this paper in Table 2.

Data Fusion. The input of data fusion is modeled as a probabilistic graphical model [19] or, more specifically, as a Bayesian network. Let $S = \{s_1, \ldots, s_n\}$ be a set of sources that provide claims about data items from set $O = \{o_1, \ldots, o_m\}$. Each data item o_i can have a number of claims, denoted by $V_i = \{v_i^1, \ldots, v_i^{|V_i|}\}$. A set of claims on all data items is denoted by $V = \{V_1, \ldots, V_{|O|}\}$. Sources provide specific claims for data items (at most one per data item), modeled as a set of observations $\Psi = \{\psi_{j,i,k}\}$, where

$$\psi_{j,i,k} = \begin{cases} 1 & \text{if } s_j \text{ votes for claim } v_i^k \text{ of } o_i \\ 0 & \text{otherwise} \end{cases}$$

EXAMPLE 1.1. *In Table 1, the set of all claims about data item* `Rio` *is* $V_6 = \{$`Jones, Saldanha`$\}$ *and the fact that source* S_1 *provides claim* `Jones` *and not* `Saldanha` *is represented by setting* $\psi_{1,6,\texttt{Jones}} = 1$ *and* $\psi_{1,6,\texttt{Saldanha}} = 0$.

Notation	Definition
O	set of data items
o_i	the i-th data item
m	number of unvalidated data items
S	set of sources
s_j	the j-th source
n	number of sources
V_i	set of claims for data item o_i
v_i^k	the k-th claim of data item o_i
$\psi_{j,i,k}$	observation of claim v_i^k by source s_j
$S(v_i^k)$	set of sources that vote for claim v_i^k
$\mathcal{F}$	data fusion system
A_j	accuracy of source s_j
p_i^k	probability that claim v_i^k of o_i is `true`
Θ	set of possible actions
θ_i	action denoting the validation of data item o_i

Table 2: Terminology used in the paper.

DEFINITION 1. *A database, D, is a tuple $\langle O, S, \Psi, V\rangle$ where O is the set of data items, S is the set of sources, $V = \{V_1, \ldots, V_{|O|}\}$ is the set of claims per data item and Ψ is the set of observations.*

Given all components defined above, we formally introduce a data fusion system with its input and output structures:

DEFINITION 2. *A data fusion system, denoted by $\mathcal{F}$, is a function that takes database D as input, and outputs a set of probability assignments (P), i.e.,*

$$\mathcal{F} : D \rightarrow \langle P \rangle$$

where for each data item $o_i \in O$, $P(v_i^k) = p_i^k \in [0, 1]$ is the correctness of claim v_i^k, i.e., the probability that claim $v_i^k \in V_i$ is `true`*. (In some cases, the output includes source accuracies (A) where for each $s_j \in S$, $A(s_j) = A_j$ is the accuracy of source s_j).*

Feedback Solicitation. To improve the effectiveness of a data fusion system, we solicit feedback in the form of validation of a data item, e.g., we ask the user to provide the `true` director of `Zootopia`.

Action. The validation of a data item $o_i \in O$ is called an *action* and is denoted by θ_i. The space of possible actions Θ, is determined by the set of data items that have not yet been validated.

Problem Statement. Given a data fusion system $\mathcal{F}$ and its output $\langle P, A\rangle$, we address the problem of determining the next action θ_i from the set of possible actions Θ to solicit feedback from a user.

1.3 Summary of Contributions

Our main contributions can be summarized as follows:

- We formalize the problem of ordering user feedback effectively to improve the performance of existing fusion techniques (Section 1.2).
- We propose strategies to generate an effective ordering in which claims should be validated (Section 4). Our item-level ranking strategies consider data items individually (Section 4.1) while our novel decision-theoretic framework, based on the concept of value of perfect information, evaluates data items holistically(Section 4.2).

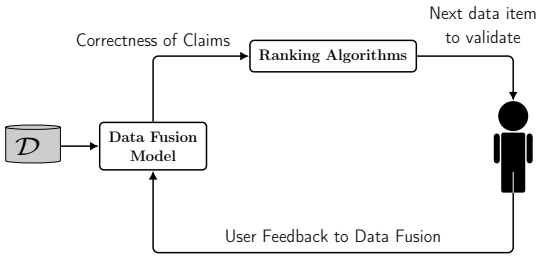


Figure 1: The proposed user feedback framework.

- To scale up the decision-theoretic framework, we derive approximation formulae that quantify the impact of a validation by analytically estimating the change it effects in the correctness of other claims. (Section 4.2.3)
- We conduct an extensive experimental evaluation on real-world datasets where we demonstrate the efficacy of the proposed methods in improving conflict resolution, and present trade-offs between user involvement and effectiveness of the methods. (Section 5)

2. SOLUTION OVERVIEW

Given conflicting data from multiple data sources and a data fusion system, we focus on the problem of determining the best data item for the user to validate (Figure 1).

To generate an ordering in which data items should be validated, we first propose two *item-level* ranking strategies that evaluate data items individually based on their local characteristics. The strategies are based on the idea of uncertainty inherent in a data item and items with higher uncertainty are preferred for validation. However, because of not considering other unvalidated data items, the item-level ranking approaches have certain limitations.

To address those limitations, we propose a novel decision-theoretic framework that assesses data items holistically and considers all other items while determining the benefit of validating a particular data item. Our decision-theoretic framework uses the concept of *value of perfect information* (VPI) [31] that is based on a *utility function* to measure the desirability of the current state of a system for its users. It then selects a claim validating which maximizes the gain in utility function. We show that this procedure leads to a prohibitively expensive computational cost because we need to fuse data each time we wish to compute the utility gain of a data item. To scale up our framework to large-scale datasets, we propose to analytically estimate the impact of a validation on other unvalidated data items, and select a claim that has the maximum utility gain over the estimates.

We incorporate the acquired user feedback in the form of initial truth labels for data fusion.

3. DATA FUSION MODEL

We start with describing the details of data fusion: consider a set of data sources S that provide conflicting claims on data items in O (see Section 1.2); the goal of data fusion is to identify the correct claim of each data item.

We assume the model proposed in [7] (`AccuNoDep`) that considers the accuracies of sources and assumes sources to be independent. Because of its ease of understanding and interpretation, this fusion model forms the basis for a number of other variants of fusion [6, 7, 24]. In this model, there are observations (the votes of sources on claims, Ψ), and hidden variables (the accuracies of sources, A, and the correctness of claims, p_i^k); the objective is to infer the hidden variables given the observations. The true claims of data items are determined by iteratively inferring the hidden variables:

ID	Probabilities of Claims
O_1	Howard (0), Spencer (1)
O_2	Stevenson (0.015), Nelson (0.985)
O_3	Docter (0.999), leFauve (0.001)
O_4	Stanton (1)
O_5	Coffin (0.921), Renaud (0.079)
O_6	Saldanha (0.985), Jones (0.015)

Table 3: Output of data fusion for the example in Table 1. Value in parenthesis shows the probability that a claim is considered correct.

1. **Correctness of a claim.** The model uses Bayesian analysis to compute the correctness of a claim from the accuracies of sources that support it. The probability of claim v_i^r of data item o_i being `true` is computed as:

$$p_i^r = p(v_i^r{=}\texttt{true} \mid \psi_{.,i,.}) = \frac{\prod_{s \in S(v_i^r)} \frac{(|V_i|-1)A(s)}{1-A(s)}}{\sum_{v_i^o \in V_i} \prod_{s \in S(v_i^o)} \frac{(|V_i|-1)A(s)}{1-A(s)}} \quad (1)$$

   where $\psi_{.,i,.}$ represents the observations for data item o_i and $S(v_i^r)$ is the set of sources that vote on claim v_i^r of o_i. In this model, only one of the claims is considered to be `true` and the rest are considered `false`.

2. **Accuracy of a source.** Source accuracies are updated using the current correctness of claims. The accuracy of source s_j is defined as the probability that its claim about a data item is `true`, and is computed as the average correctness of all its claims:

$$A(s_j) = \frac{\sum_{i=1}^{m}{}^{\psi_{j,i,k}=1} p_i^k}{N(s_j)} \quad (2)$$

   where s_j provides information about $N(s_j)$ data items.

Sources are initially assigned default accuracies. The model alternates between Steps 1 and 2 until it reaches a steady state (i.e., the accuracies of sources converge) or attains the threshold for number of iterations. Table 3 shows the output of fusion after the model has converged for the example in Table 1. A claim that has the highest probability of being true is considered correct by the model.

Note that the described fusion model is not guaranteed to converge [8]. As shown in Figure 1, we treat the data fusion model as a black box; we use the *output* of fusion to determine the next action which is, thus, independent of the convergence of the fusion model.

4. RANKING ALGORITHMS

In this section, we propose two broad ranking approaches that leverage data and the output of fusion to generate the order in which data items should be validated. The *item-level* ranking strategies presented in Section 4.1 consider

data items individually, while the decision-theoretic feedback framework of Section 4.2 evaluates data items based on their ability to impact the effectiveness of fusion on other unvalidated data items.

4.1 Item-level Ranking Strategies

This section presents two techniques that assess the local characteristics of data items to determine the next action. The techniques presented are built upon the principle of *uncertainty* inherent in a data item. Intuitively, an item with greater uncertainty offers more information to a system.

We propose using *entropy* [33] to quantify the average information content in a data item. Entropy is a way to measure the level of uncertainty in probabilistic objects. In the context of data fusion, data item o_i is a probabilistic object whose `true` claim ranges over all of its possible claims $v_i^k \in V_i$. We define the entropy of data item o_i as:

$$H_i = -\sum_{v_i^k \in V_i} p_i^k \log p_i^k \tag{3}$$

where p_i^k is the probability that claim v_i^k is `true`.

A data item that has a low entropy has a higher degree of certainty, i.e., some claim has a high probability of being `true`, compared to a data item having claims that are almost equally likely to be correct. Note that a low entropy also encapsulates the case when a false claim is considered `true` with a high probability.

Using entropy as the uncertainty measure, we determine the next action as validating the data item that has the highest entropy, i.e.,

$$a_i = \underset{\theta_i \in \Theta}{\operatorname{argmax}} \; H_i \tag{4}$$

We now present our item-level ranking algorithms that elaborate on obtaining p_i^k to use in Equation (3). In Section 4.1.1, we present an algorithm based on the disagreement of sources over claims of a data item whereas Section 4.1.2 presents an algorithm that ranks data items based on the output of data fusion.

4.1.1 Disagreement-based algorithm

This section presents Query-by-Committe (`QBC`), a technique based on the disagreement of sources over claims of a data item. `QBC` is built upon the principle of majority voting that considers the correct claim of a data item to be the one supported by a majority of the sources. The intuition is that an effective data fusion system is less likely to incorrectly identify the correct claim of an item if most of the sources agree upon it. In contrast, the `true` claim of an item disputed by many sources may be questioned. In such cases, it might be more beneficial to validate the latter data item.

`QBC` uses the votes of sources over claims to compute the correctness of a claim $v_i^k \in V_i$ as the fraction of sources (voting for o_i) that support v_i^k:

$$p_i^k = \frac{\sum_{j=1}^{n} \psi_{j,i,k}}{\sum_{r=1}^{|V_i|} \sum_{j=1}^{n} \psi_{j,i,r}} \tag{5}$$

This definition of p_k^i, termed as *vote entropy*, is used in Equation (3) to evaluate the uncertainty intrinsic to a data item. `QBC`, therefore, queries the data item most disagreed upon by sources that vote for it.

EXAMPLE 4.1. *In Table 1, the vote entropy of O_2 is computed as $H_2 = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = 0.693$, which is greater than the vote entropy of O_1 ($H_1 = -\frac{1}{3}\log\frac{1}{3} - \frac{2}{3}\log\frac{2}{3} = 0.637$). `QBC` would validate O_2 before it validates O_1.*

`QBC` has a low computational cost because it does not need to recompute entropies after a validation. However, a major drawback of `QBC` is that it does not take into account the dependencies between data items through sources.

4.1.2 Uncertainty-based algorithm

The first and foremost limitation of `QBC` is that the choice of the next action is determined solely by distribution of source votes on claims of a data item. It is agnostic to the output of fusion, i.e., it does not consider (i) accuracy of sources, and (ii) probabilities of correctness of claims. For the example in Table 1, `QBC` may select O_3 for validation even though its true claim has already been identified (Table 3).

To address this issue, we present Uncertainty Sampling (denoted by `US`), an uncertainty-based technique that selects an action the fusion system is less certain about. `US` uses the correctness of claims as output by the fusion system to compute the entropies in Equation (3). Intuitively, data items that the fusion system is least certain about are more suitable for validation, since the more confident predictions are probably correct.

EXAMPLE 4.2. *The entropy of O_5 in Table 1 is computed using the probabilities in Table 3, is $H_5 = -(0.079)\log(0.079) - (0.921)\log(0.921) = 0.276$. H_5 is greater than the entropy of all other data items and, therefore, `US` considers O_5 the most suitable for validation.*

`US` considers the output of fusion, and therefore, takes source accuracies into account; the downside is that we need to run the fusion system for each action.

The item-level ranking algorithms are easy to interpret and implement. A major drawback, however, is that these methods aim to resolve conflicts at the site of a single data item without any regard to the conflicts existing in other data items. In the following section, we present a framework that assesses data items with the objective of resolving conflicts in all unvalidated data items.

4.2 Decision-Theoretic Framework

The techniques presented in Section 4.1, although computationally inexpensive, determine actions with the view of resolving conflicts in one data item at a time. None of the methods considers possible interdependence among data items and, therefore, offers no guarantee on the improvement of fusion over other unvalidated data items.

Our objective is to globally identify the *best* action that would benefit fusion on all unvalidated data items. To this end, we design a decision-theoretic feedback solicitation framework based on the value of perfect information. We define a *utility function* to measure the usefulness of current state of fusion, and identify an action that is most likely to improve the utility of fusion for all unvalidated data items. To the best of our knowledge, none of the earlier works incorporates the value of information for the problem of data fusion.

4.2.1 Background Concepts

We introduce the basic concepts of our framework such as utility and the value of perfect information. We show

that in the absence of ground truth, we have to rely on an alternative utility function based on the idea of uncertainty reduction (referred to as the *entropy utility function*).

Utility function. We define the utility function as a function that measures the usefulness of a data fusion system. The utility of a system is higher if it is able to identify a greater number of `true` claims correctly. Let $\mathcal{T} : v_i^k \rightarrow \{\texttt{true, false}\}$ be a truth function that assigns the label `true` to a correct claim and `false` to an incorrect claim.

DEFINITION 3. *Given truth function $\mathcal{T}$, database D and fusion system $\mathcal{F} : D \rightarrow \langle P, A\rangle$, the utility function $U(D, \mathcal{F}, \mathcal{T})$ is defined as:*

$$U(D, \mathcal{F}, \mathcal{T}) = \frac{1}{|V|}\left(\sum_{V_i \in V}\sum_{v_i^k \in V_i}\frac{p_i^k \delta(\mathcal{T}(v_i^k))}{\mid V_i \mid}\right)$$

where $p_i^k \in P$ *and* $\delta(v) = \begin{cases}1, & \text{if } v \text{ is correct}\\ 0, & \text{otherwise}\end{cases}$

The utility function can be interpreted as measuring the average correctness of `true` claims based on the output of fusion system $\mathcal{F}$. The closer the utility function is to 1, the higher is the effectiveness of $\mathcal{F}$.

Value of Perfect Information. We measure the usefulness of an action θ_i with respect to our utility function by using the value of perfect information (VPI). VPI has been used widely in areas such as economics [27], healthcare [3], data cleaning [17, 40, 38, 28, 16] and classification [18].

DEFINITION 4. *The value of perfect information (VPI) of action θ_i is defined as:*

$$VPI(\theta_i) = \sum_{v_i^k \in V_i} U(D, \mathcal{F}, \mathcal{T} \mid \mathcal{T}(v_i^k){=}\texttt{true})p_i^k - U(D, \mathcal{F}, \mathcal{T})$$

The VPI of action θ_i is the expected gain in the utility function earned by validating data item o_i. To compute $U(D, \mathcal{F}, \mathcal{T} \mid \mathcal{T}(v_i^k){=}\texttt{true})$, we consider claim v_i^k to be true and input this information to the data fusion system as prior knowledge by setting $p_i^k{=}1$ and $p_i^f{=}0\ \forall\ v_i^f \in V_i \setminus \{v_i^k\}$.

A set of all possible actions, denoted by Θ, consists of an action θ_i for each unvalidated data item $o_i \in O$. Our goal is to identify the action that has the highest VPI, i.e.,

$$\theta_i = \underset{\theta_i \in \Theta}{\text{argmax}} \quad VPI(\theta_i) \tag{6}$$

4.2.2 Maximum Expected Utility

Real-world applications prevent us from using the utility function from Definition 3 because we do not possess the truth function $\mathcal{T}$, i.e., ground truth is not available. To this end, we define an *entropy utility* function to identify actions that reduce the uncertainty associated with the output of fusion. This idea, known as *uncertainty reduction*, has been extensively used in the past [37, 28, 16, 2, 42].

DEFINITION 5. *Given database D and data fusion system $\mathcal{F} : D \rightarrow \langle P, A\rangle$, the entropy utility function is defined as the sum of entropies across all data items in D, i.e.,*

$$EU(D, \mathcal{F}) = -\sum_{o_i \in O} H_i = -\sum_{o_i \in O}\sum_{v_i^k \in V_i} p_i^k \log\ p_i^k$$

where $p_i^k \in P$ is the probability that claim $v_i^k \in V_i$ is `true`.

Algorithm 1: `MEU` Algorithm

```
for each unvalidated data item o_i do
    for each claim v_i^k ∈ V_i do
        Compute EU(D, F | v_i^k = true)
    end for
    Compute ΔEU_i as in Equation (7)
end for
Select the action with the maximum ΔEU_i
```

The entropy utility function measures the average uncertainty in the correctness of claims; the closer the entropy utility is to 0, the higher is the effectiveness of fusion.

We present Maximum Expected Utility (denoted by `MEU`), a framework that integrates the entropy utility function with the concept of VPI. `MEU` uses $EU(D, \mathcal{F})$ as the utility function in Definition 4 instead of $U(D, \mathcal{F}, \mathcal{T})$ to compute the expected entropy utility gain of action θ_i as:

$$\Delta EU_i = EU(D, \mathcal{F}) - EU(D, \mathcal{F} \mid v_i^k = true)p_i^k \tag{7}$$

(Note the change in order of the terms in Equation (7) and Definition 5. This is because our goal is to reach a state of lower uncertainty than before, i.e., we ideally want the first term in Equation (7) to be greater than the second term.)

`MEU` considers the one-step lookahead state of fusion after a *potential* action and identifies one that has the highest expected entropy utility gain, i.e.,

$$\theta_i = \underset{\theta_i \in \Theta}{\text{argmax}} \quad \Delta EU_i \tag{8}$$

This kind of validation strategy is *myopic* in nature because we look only one step ahead each time we make a decision. It is possible that some action may not lead to the highest VPI at the current step but validating it can result in a higher VPI in subsequent validations. Sequential validations are challenging and often computationally expensive [17]; the present work focuses only on myopic strategies.

EXAMPLE 4.3. *For the example in Table 1, we use Table 3 to compute $EU(D, \mathcal{F}) = 0.437$. Considering O_1 for validation, Table 4 shows the output of fusion when* `Howard` *is true and Table 5 shows the output when* `Spencer` *is true. (For ease of display, we represent the columns to be claims as they appear in Table 3, e.g., for O_1, p^0 represents the correctness of claim* `Howard` *and p^1 the probability of* `Spencer`*.)*

ID	p^0	p^1
O_1	1	0
O_2	0.082	0.918
O_3	0.045	0.955
O_4	1	
O_5	0.004	0.996
O_6	0.918	0.082

Table 4: `Howard=true`.

ID	p^0	p^1
O_1	0	1
O_2	0.004	0.996
O_3	1	0
O_4	1	
O_5	0.944	0.056
O_6	0.996	0.004

Table 5: `Spencer=true`.

Using Tables 4 and 5, `MEU` computes $EU(D, \mathcal{F}|\texttt{Howard} = true) = 0.781$, and $EU(D, \mathcal{F}|\texttt{Spencer} = true) = 0.262$. The expected utility of $O_1 = 0(0.781) + 1(0.262) = 0.262$.

Table 6 shows the expected utility (EU) of all data items. `MEU` decides to validate O_5 because its utility gain ($(EU(D, \mathcal{F}) - EU_5^*) = 0.385$) is the highest among all items.*

ID	O_1	O_2	O_3	O_4	O_5	O_6
EU*	0.262	0.231	0.258	0.262	0.052	0.231

Table 6: Expected utility of data items in Table 1.

In the absence of ground truth, maximum expected utility (MEU) [31] is considered to be the best alternative to ground truth utility. The main drawback of `MEU` is its lack of efficiency. To determine the next action, `MEU` re-runs fusion $\mathcal{F}$ on database D for each claim of every data item $o \in O$. The time complexity of `MEU` is $O(m\kappa t_{\mathcal{F}})$ where m is the number of unvalidated data items in D, κ is the average number of unique claims per data item and $t_{\mathcal{F}}$ is the time needed to run $\mathcal{F}$ on one instance of data. A typical run of fusion iterates over all data items and all sources until convergence. This contributes to an $O(m\kappa\mathcal{I}(m+n))$ complexity where $\mathcal{I}$ is the average number of iterations to convergence and n is the number of sources. With data items far outnumbering sources, the result is a complexity of $O(m^2\kappa\mathcal{I})$. Concluding, `MEU` can tackle datasets a few hundred data items in size in a reasonable amount of time. Our objective is to be able to process datasets with at least a few thousands of data items.

4.2.3 Approximate-MEU

`MEU` describes a general decision-theoretic framework for the problem of ordering conflicts for user feedback in data fusion. However, the extreme computational cost of `MEU` makes it infeasible for large-scale datasets.

To this end, we present `Approx-MEU`, a method that leverages the structure of interactions between data items and sources to estimate the impact of a validation on the correctness of other unvalidated data items.

This approach is built on the intuition that an action would alter the correctness of claims of not only the validated data item but also of its neighbors (as in Figure 2). The idea is based on principles inherent in Bayesian network inference methods such as belief propagation [19], variational message passing [39] and incremental expectation-maximization [26]. These methods decompose the computation into local data item calculations and pass them to other items via messages. We consider a validation to be a local update of the correctness of claims of a data item.

Consider data items o_i and o_j. The goal of `Approx-MEU` is to estimate the correctness of claims of o_j after o_i has been validated. This computation involves the following two steps: (i) measuring the change in correctness of claims of the validated item o_i, and (ii) estimating the change in correctness of claims of o_j as a function of the change in probabilities of o_i. We estimate the correctness of claims of unvalidated data items using the method of linear approximation by differentials in the following steps.

Change in probabilities of claims of o_i

We assume an arbitrary claim $v_i^t \in V_i$ to be `true`. Upon validating o_i, the change in probability of v_i^t is: $\Delta p_i^t = (1 - p_i^t)$. This validation ensures that the remaining claims in V_i are `false`. The change in probability of $v_i^f \in V_i \setminus \{v_i^t\}$ is : $\Delta p_i^f = (0 - p_i^f) = -p_i^f$.

Propagation of changes from o_i to o_j

We consider data items o_i and o_j to be connected either through a source that votes for both of them or through a path consisting of alternating sources and items. As seen in the graph in Figure 2, O_1 and O_2 are connected (because source S_3 votes for both) whereas O_2 and O_4 are connected via the $\langle O_2, S_3, O_1, S_4, O_4\rangle$ path. We present an analysis of both the cases:

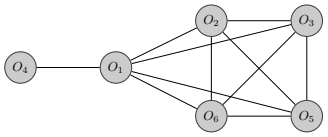


Figure 2: Graph of data items in Table 1: an edge implies there is at least one source that provides information for the connecting data items.

1. **o_i and o_j have at least one common source.** We first examine how the change in correctness of claims in V_i impacts the accuracies of sources that vote for both o_i and o_j (because change is propagated to o_j through these sources).

   Updates in source accuracies. The intuition behind the effect of changes in o_i to sources that vote on it is straightforward: we reward sources that support the correct claim $v_i^t \in V_i$ by trusting it more on information it provides on other data items. Similarly, our model penalizes sources that vote on some other claim v_i^f by discounting its information on other data items as well. From Equation (2), the change in accuracy $A(s)$ of a source s is computed as:

   $$\Delta A(s) = \begin{cases} \Delta p_i^t / N(s), & \text{if s votes for } v_i^t \\ \Delta p_i^f / N(s), & \text{if s votes for } v_i^f \in V_i \setminus \{v_i^t\} \end{cases} \tag{9}$$

   where $N(s)$ is the number of data items for which s votes.

   Propagation of updates in sources to o_j. Our next task is to measure further propagation of changes from the sources to o_j. This part of the analysis involves a short sequence of basic calculus over the formulae described in Section 3.

   We compute the change in correctness of claim $v_j^r \in V_j$ attributable to the change in probabilities of claims of o_i by the method of approximation by differentials (details of the derivation in Appendix A.1).

   $$\Delta p_j^r = -(p_j^r)^2 \sum_{v \in V_j} \left(\frac{\prod_{s \in S(v)} \frac{(|V_j|-1)A(s)}{1-A(s)}}{\prod_{s \in S(v_j^r)} \frac{(|V_j|-1)A(s)}{1-A(s)}} \right) \cdot \left(\sum_{s \in S(v)} \frac{\Delta A(s)}{A(s)(1-A(s))} - \sum_{s \in S(v_j^r)} \frac{\Delta A(s)}{A(s)(1-A(s))} \right) \tag{10}$$

   For each of the sources s that vote for o_j, the term $\Delta A(s)$ in Equation (10) takes a value as noted in Equation (9) depending on whether: (i) s supports v_i^t, (ii) s supports a claim other than v_i^t, or (iii) s does not provide any information on o_i. Clearly, if s belongs to the third category, it will not be affected by the validation of o_i.

   With Δp_j^r, the approximate change in correctness of claim v_j^r, the updated correctness of claim v_j^r is computed as:

   $$(p_j^r)' = p_j^r + \Delta p_j^r \tag{11}$$

Algorithm 2: `Approx-MEU` Algorithm

```
for each unvalidated data item o_i do
    for each claim v_i^k ∈ V_i do
        Assume v_i^k is true
        for each unvalidated data item o_j ≠ o_i do
            for each claim v ∈ V_j do
                Estimate updated correctness of v
            end for
        end for
        Compute entropy utility of updated probabilities
    end for
    Compute ΔEU_i as in Equation (13)
end for
Select next action according to Equation (8)
```

2. o_i **and** o_j **have no source in common.** We know that any change in o_i reaches data items connected to it via at least one source, i.e., through data items that are one-hop away from o_i in the graph of data items. The changes in these data items then reach data items one-hop away from them, and so on.

THEOREM 4.1. *The change in correctness, Δp_j^r, of claim $v_j^r \in V_j$ attributable to the change in correctness, Δp_i^k, of claim $v_i^k \in V_i$ is inversely proportional to the minimum number of data items a source votes for, raised to the power of d, the number of hops o_j is away from o_i.*

$$\Delta p_j^r \propto \left(\frac{1}{N^d}\right) \Delta p_i^k$$

Proof. Details of the proof are in Appendix A.2. □

Real-world datasets typically consist of few sources providing claims about a large number of data items. As a result, most of the items are connected to each other in the graph of data items. Through Theorem 4.1, we observe an exponential decay in the change in correctness of claims as we move away from the validated item.

Deciding the next action. Using Equation (11), `Approx-MEU` estimates first-order approximations of correctness of claims of data items within one hop of o_i attributable to validating claim $v_i^k \in V_i$.

We compute the entropy of data item o_j over the estimated correctness of its claims, i.e.,

$$H_j = -\sum_{v_j^k \in V_j} (p_j^k)' \log (p_j^k)' \tag{12}$$

The expected utility gain of action θ_i is then expressed as:

$$\Delta EU_i = EU(D, \mathcal{F}) - \sum_{v_i^k \in V_i} p_i^k \sum_{o_j \in O} H_j \tag{13}$$

and the next action is determined as in Equation (8).

EXAMPLE 4.4. *Consider O_3 for validation in Table 1. Table 7 shows the estimated correctness of claims obtained using Equation* (11) *when* `Docter` *is true and Table 8 shows the estimated probabilities when* `leFauve` *is correct.*
The expected utility of $O_3 = 0.999(0.401) + 0.001(0) = 0.401$.

Table 9 shows the expected utility (EU) of all data items using the approximate correctness of claims.* `Approx-MEU` *validates O_2 because it has the highest expected utility gain.*

ID	p^0	p^1
O_1	0	1
O_2	0.019	0.981
O_3	1	0
O_4	1	
O_5	0.931	0.069
O_6	0.99	0.01

Table 7: `Docter=true`.

ID	p^0	p^1
O_1	0	1
O_2	1	0
O_3	0	1
O_4	1	
O_5	1	0
O_6	1	0

Table 8: `leFauve=true`.

ID	O_1	O_2	O_3	O_4	O_5	O_6
EU*	0.437	0.184	0.401	0.437	0.235	0.313

Table 9: Expected utility of data items in Table 1.

Complexity. For each unvalidated data item, `Approx-MEU` assumes each of the claims to be `true` (one at a time) and estimates the first-order approximate correctness of claims of data items one-hop away from it. By eliminating the bottleneck iterative computation in `MEU`, `Approx-MEU` has a complexity of $O(m\kappa d)$ where m is the number of unvalidated data items, d is the average number of data items connected to a data item through a source and k is the number of claims per item. In the worst case, $d = m$, when every data item is directly connected to every other data item.

4.3 Further Optimizations

We now describe further optimizations to effectively scale up our ranking strategies. We briefly elaborate on bounding the number of data items to consider for validation and the effect of batch size on the performance of fusion.

1. **Shrinking the search space.** In datasets where all data items are connected to each other through one ore more sources, the complexity of `Approx-MEU` blows up to $O(\kappa m^2)$. To efficiently scale up the approximation formulae for such dense data, we propose a hybrid approach that takes the best insights from `QBC`, `US` and `MEU`:
   (a) Data items with high vote entropy (`QBC`) are the most disputed ones and, therefore, suitable for validation;
   (b) Data items with low uncertainty over output of fusion are less suited to validation (similar to `US`);
   (c) Among the high-entropy items, our goal (as in `MEU`) is to validate one with a greater expected utility gain.

   We denote by $\texttt{Approx-MEU}_k$, the method that ranks unvalidated data items by their vote entropies and considers the top k% data items for the impact computation step. By tuning the value of k, we improve the complexity of `Approx-MEU` to $O(\kappa k^2)$.

2. **Batch of Actions.** The present work deals with one action at a time. If we have a budget of, say, twenty actions in total, one may argue that the most effective method should identify the set of best twenty actions that would result in the maximum expected utility gain. The task of finding an optimal set of twenty actions, however, is not efficient: it is computationally expensive because the algorithm would need to consider all possible subsets of twenty actions. It is also not effective: by soliciting validation of twenty data items at once, we lose out on the opportunity to integrate earlier actions before deciding the next action. Our framework could be easily extended to solicit the top twenty actions that have the

highest expected utility gain. While slashing run-time by reducing the number of iterations, this approach is expected to converge to ground truth slower than when we validate one data item at a time. (We present the results of this approach in Appendix B.4).

4.4 Feedback Errors

So far, we have assumed access to accurate feedback from an expert. Real-world applications, however, are often faced with two major concerns: (1) Experts are expensive and often vary across domains; (2) Users (experts and otherwise) often give erroneous feedback.

To address these issues, in light of the recent advances in crowdsourcing [14], applications often turn to collecting feedback from a crowd of readily-available *workers*. Note that workers add a third dimension to the problem of data fusion previously governed by data items and sources; worker errors are independent of source (extraction) errors. Prior research that deal with non-experts [34, 9] jointly estimate user quality and true labels of data items, and query only the more trustworthy users in subsequent feedback rounds. The present work focuses on true labels of data items and does not address modeling the quality of users in a crowd setting. We assume that the crowd provides us either a single claim considered (partially) correct or probabilities representing correctness of claims of a data item.

Consider the case when a user (or, crowd) provides feedback for the data item that our ranking algorithm has determined to be the most beneficial for fusion. In the best case, all feedback is correct. To integrate erroneous input into our framework, we translate imperfect feedback to correctness of claims and leverage this prior knowledge, along with the observations, to estimate the correctness of claims for rest of the data items.

1. **Feedback confidence.** In some cases, users express confidence in their feedback, e.g., '80% certain that v_i^k is the correct claim for data item o_i'. We incorporate this knowledge into our model by assigning the confidence to correctness of the claim, i.e., $p_i^k = 0.8$ and the rest as 0.

2. **Incorrect feedback.** This case pertains to quality of the user (or, crowd) providing feedback. In case of a crowd, we assume that the crowdsourcing system processes conflicting answers from workers and provides the most accurate label.Knowing the user's (or, crowd's) error-rate ϵ, e.g., on 4 out of 6 instances, the feedback is incorrect, we compute the expected utility gain over correct and incorrect feedback. If the provided claim v_i^k is correct, we set $p_i^k = 1$ and the rest as 0. Otherwise, we set $p_i^k = 0$ and set a uniform probability distribution for rest of the claims, i.e., $p_i^r = 1/|claims|$ whenever $r \neq k$.

3. **Conflicting feedback.** We also consider the case when, instead of providing a single correct claim for a data item, the crowd simply presents the answers from different workers. For example, say for data item o_i having three claims (v_i^A, v_i^B, v_i^C), 6 workers agree on v_i^A being correct, 3 agree on v_i^B and 1 says claim v_i^C is correct. We summarize this information in the form of probabilities either by counting or some other mechanism, i.e., we conclude that $(p_i^A, p_i^B, p_i^C) = (0.6, 0.3, 0.1)$ and feed this knowledge to the data fusion model.

5. EXPERIMENTAL EVALUATION

This section presents an empirical evaluation of the proposed solutions on two real-world datasets. Our objectives are: (1) To assess the effect of acquiring feedback in improving the performance of data fusion, (2) To evaluate the proposed ranking algorithms, and (3) To analyze the trade-offs between effectiveness and efficiency offered by the various approaches. Moreover, we study the behavior of the methods on data with different characteristics and with respect to parameters such as batch size and erroneous feedback.

Datasets

To validate the proposed methods, we conducted experiments on the following real-world datasets (Table 10):

	Books	FlightsDay	Population	Flights
Items	1263	5836	40696	121567
Sources	894	38	2545	38
Claims	24303	80452	46734	1931701

Table 10: Statistics of real-world datasets.

Books: We used the books dataset from [7] that contains a listing of computer science books and their authors as provided by different bookstores registered at *abebooks.com*.

Flights: We used the flights dataset from [21] that contains status information for flights over an entire month as reported by 38 sources. A data item is an attribute (such as scheduled arrival time) of a particular flight. We permit slightly different reported values (to a maximum difference of 10 minutes) in flight times that might have arisen due to slight lag in updates, or error in estimating times.

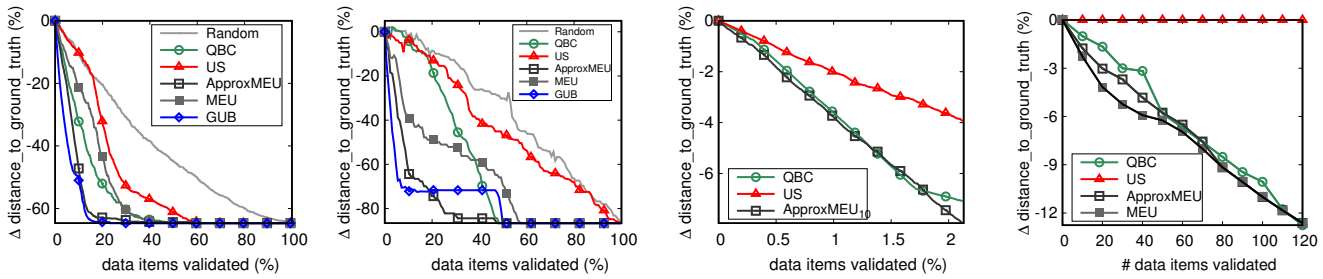
FlightsDay: We used a one-day snapshot of Flights (for the day of 12/1/2011); this dataset is representative of the Flights dataset that spans over a month's time.

Population: We used the city population dataset from [29] that contains Wikipedia edit histories of the populations of certain cities in a given year. To account for unreasonably large values and to have a source provide a single claim per data item, we adopt preprocessing steps similar to [20].

For simplification, we consider only those flight and population data items that have up to two contesting values. In Books, we consider the top two author sets per book.

Data Characteristics: We notice that our real-world datasets exhibit interesting properties: (i) Most of the data items in the flights datasets are connected to each other because the small number of sources provide information on almost all data items, (ii) Both Books and Population exhibit long-tail characteristics (Appendix B.1, Figure 8) where more than 90% sources provide information on fewer than 4% data items. Such varied characteristics of data allow us to evaluate our approaches in different scenarios.

Feedback Simulation. We simulated user feedback for data items by providing feedback as determined by the ground truth. We used the silver standard provided in [7] as the ground truth for Books. For Flights, we considered data provided by each of the carrier websites, *American Airlines, United Airlines* and *Continental*, to be the ground truth. We manually identified the true claim for data items in Populationthat have more than one claim.



(a) FlightsDay. (b) Books. (c) Flights. (d) Population.

Figure 3: Effectiveness of different ranking strategies measured as the reduction in `distance_to_ground_truth` against number of items validated. The VPI-based framework (`GUB`, `MEU`, `Approx-MEU`) is seen to demonstrate superior performance compared to the item-level ranking strategies (`QBC`, `US`) and the naïve strategy (`Random`).

Competing Methods

We compared the following ranking approaches:

1. **QBC** (Section 4.1.1): This item-level ranking method uses the distribution of claims to rank data items.
2. **US** (Section 4.1.2): An item-level ranking method that uses correctness of claims as output by the fusion system to rank data items.
3. **Greedy Upper Bound (GUB)** (Section 4.2.1): Assuming that ground truth is known, this method selects an action that results in the highest ground truth utility gain according to Definition 4.
4. **MEU** (Section 4.2.2): In the absence of ground truth, this method selects the action that has the maximum expected utility gain.
5. **Approx-MEU** (Section 4.2.3): A decision-theoretic approach that ranks data items according to their approximate impact on other unvalidated data items.
6. **Random**: This naïve method selects an action at random; all data items are considered equally beneficial.

We implemented all the algorithms in Java, and ran experiments on a Macbook Pro with 8GB RAM, 2.7 GHz Intel Core i5 processor, and OSX El Capitan 10.11.5.

Performance Metrics

Effectiveness: To evaluate the effectiveness of the proposed methods, we conducted a sequential validation of all data items having conflicting claims (in the order determined by a given method) and obtained an assignment of `true` and `false` claims using a truth function $\mathcal{T}$. We report the following metrics on the results:

1. **Distance to ground truth**: We report the improvement in output of data fusion after an action as the reduction in distance of probabilities of claims to ground truth defined as:

$$\texttt{distance\_to\_ground\_truth} = \sum_{i=1}^{|O|} \sum_{v_i^k \in V_i} \frac{\delta(\mathcal{T}(v_i^k))(1-p_i^k)}{|O|}$$

where $\delta(\mathcal{T}(v_i^k)) = 1$, if $v_i^k =$ `true`. Intuitively, the `distance_to_ground_truth` can be seen as the average error of data fusion. The smaller the `distance_to_ground_truth`, the more accurate is the output of fusion.

2. **Uncertainty**: We report the reduction in uncertainty over output of data fusion defined as the entropy across all data items:

$$\texttt{uncertainty} = -\sum_{i=1}^{|O|} \sum_{k=1}^{|V_i|} -p_i^k \log(p_i^k)$$

where p_i^k is the probability that claim $v_i^k \in V_i$ is correct. A higher value of `uncertainty` indicates less confidence in the output of data fusion.

Once a data item is validated, we retain the validation result and therefore, observe a cumulative gain of all validations. Figure 3 presents example curves for the effectiveness metrics that start at 0 (when no data item is validated) and gradually approach -100% (when all items are validated). A plot closer to the axes indicates a better method.

Efficiency: To evaluate the efficiency of an approach, we report the average time it takes to determine the next action.

5.1 Evaluation of ranking strategies

In this section, we evaluate effectiveness of the item-level ranking strategies (Section 4.1) and the decision-theoretic framework (Section 4.2) in improving the performance of data fusion. Our best-case decision-theoretic mechanism involves a utility function based on the ground truth.

Effectiveness. Assuming the availability of a ground truth utility function, we demonstrate in Figure 3, the gradual improvement in distance to ground truth for increasing number of validated data items for all the validation methods.

As illustrated in Figure 3, all the approaches improve the distance of the output of fusion to ground truth, albeit by various degrees. `Random` almost linearly decreases the distance to ground truth indicating that only the number of actions determines its effectiveness. `QBC` and `US`, through guided selection of data items, converge to ground truth faster than `Random`; `QBC` consistently performs better than `US`. Specifically, in the long-tail datasets, because the adopted data fusion model assigns either very high or low probabilities to claims, most of the data items have very low uncertainties and therefore, `US` is unable to distinguish them. On the other hand, true quality of the sources in dense datasets is aptly reflected in their accuracies and correctness of claims. The data items selected by `US` are also ones that the data fusion model has not been able to resolve, indicating that these items are probably not well-connected to other data items. Validating these data items, therefore, does not have much impact on the accuracy of other items.

We notice that `MEU` is consistently superior to `US`, indicating that we benefit from a method that aims at reducing uncertainty across all data items instead of resolving a single

time (sec)	QBC	US	MEU	ApproxMEU
Books	0.01	0.001	11.73	0.231
FlightsDay	0.045	0.002	90.00	4.401
Population	0.14	0.011	> 5 min	9.728

time (sec)	QBC	US	ApproxMEU_5	ApproxMEU_{10}
Flights	7	4	146	348

Table 11: Time taken to determine the next action.

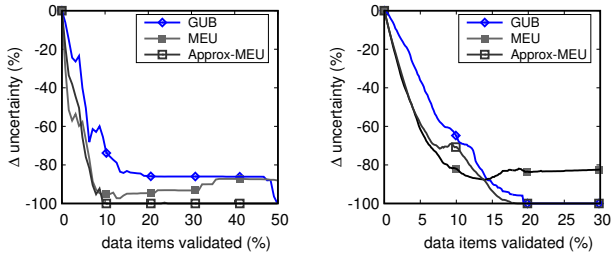
uncertain data item. We also observe that MEU and QBC have contrasting performances in long-tail datasets and in dense data. This behavior is attributable to the structure of the datasets: each source in dense data (e.g., FlightsDay) provides information on a large number of items. The change in accuracy of a source upon a validation is, therefore, not large enough to propagate to other items. It is useful in such cases to validate items with higher vote entropies first.

Not surprisingly, GUB has the steepest initial curve among all the methods. GUB takes advantage of the ground truth information and, therefore, theoretically, has the best performance in reporting the `distance_to_ground_truth`.

Interestingly, we observe that after GUB, Approx-MEU has the best performance in FlightsDay and Books – the method estimates expected correctness of claims from a validation and aims to reduce uncertainties in the estimates across all data items, thus outperforming both the item-level ranking algorithms (QBC, US). However, in Population, the room between QBC, Approx-MEU and MEU is not very large. This similarity in performance of the methods is due to sparsity of the data ($|V|/(|O| \times |S|) = 0.04\%$) which results in a very small portion of data items ($\sim 2.5\%$) having more than one claim. The idea then is to identify among these items, those that are the most beneficial to others. Both Approx-MEU and MEU, therefore, have an advantage over QBC that does not take into account the holistic impact of an action.

To scale up Approx-MEU to large dense data (Flights), we set $k = 10$ in Approx-MEU_k. With as few as a tenth of the total number of data items considered for validation, Approx-MEU_k is seen to achieve higher quality fusion results than QBC and has significantly better performance than US. Although Approx-MEU and QBC are comparable in early validations, Approx-MEU displays a notably rapid rate of convergence to ground truth as more items are validated. The results further confirms effectiveness of the decision-theoretic framework over item-level ranking methods. However, considering both effectiveness and efficiency, in such large dense data, QBC might be a better choice than Approx-MEU_k if $k << |O|$.

Efficiency. In Table 11, we report the average time taken by the methods for one validation (recall that we cannot compare GUB on real data, and we cannot scale MEU to large dense data). The item-level ranking algorithms (QBC, US) are observed to be significantly faster than the decision-theoretic framework (MEU, Approx-MEU); QBC makes a single pass over all data items and US ranks them after each validation whereas MEU and Approx-MEU fuse data with each claim of an item separately considered as prior knowledge. The high numbers for MEU motivate the need for a cheaper (but effective) alternative. Approx-MEU, while still slower than QBC and US, is faster than MEU by almost two orders of magnitude. Our goal for efficiency is to provide an interactive validation time for users of a data fusion system. We conclude that MEU cannot be used for datasets typical to data



(a) Books. (b) FlightsDay.

Figure 4: Comparing methods based on entropy utility function (MEU, Approx-MEU) against ground-truth-based method (GUB). Both MEU and Approx-MEU have comparable reductions in uncertainty (which is better than that of GUB).

fusion. From a theoretical standpoint, the time for MEU is based on time for the fusion system since it runs the system for all claims of each data item.

Practicability of Entropy Utility. The strength of GUB lies in its access to a ground truth utility function. However, real datasets provide the ground truth for a small subset of data items. In this experiment, we assess the feasibility of entropy utility function as a substitute to the ground truth utility function by comparing the performance of entropy-utility-based methods (MEU, Approx-MEU) against that of the ground-truth-based method (GUB).

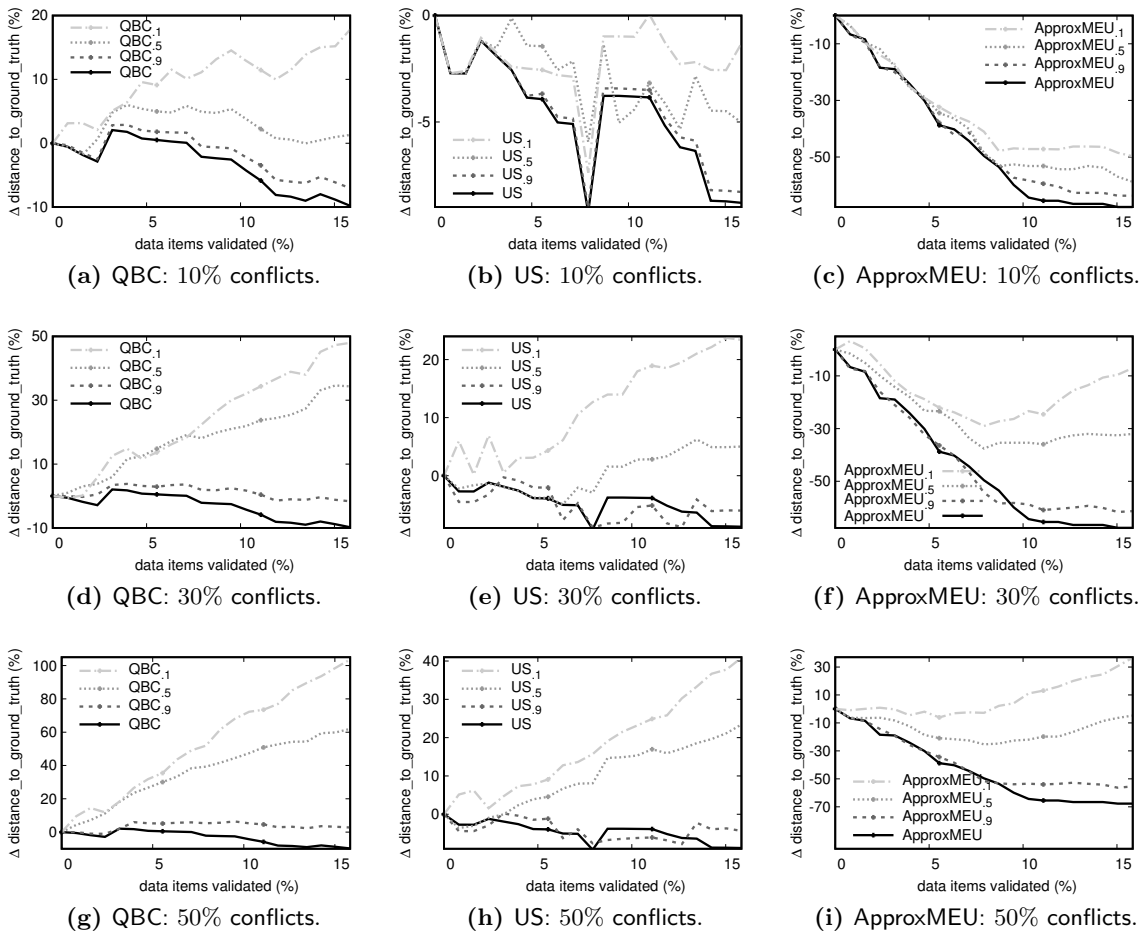
As shown in Figure 4, MEU and Approx-MEU achieve a greater reduction in uncertainty than GUB. This mechanism comes at the price of MEU converging to ground truth at a rate slower than GUB (Figures 3a and 3b). Interestingly, the rate of convergence to ground truth of Approx-MEU is better than MEU and is almost identical to GUB. Practically, however, GUB is infeasible; MEU and Approx-MEU are our best viable alternatives.

Takeaways. (1) Active feedback improves data fusion better than a passive approach (Random). (2) The decision-theoretic framework (MEU, Approx-MEU) exhibits effectiveness superior to that of the item-level ranking approaches (QBC, US); in practice, however, the latter are significantly faster methods. (3) The entropy utility function is a suitable alternative to the ground-truth utility function. (4) MEU has an extreme computation cost and cannot be used for validation on large datasets. (5) Approx-MEU is a cheaper, and also effective, substitute to MEU.

5.2 Feedback Errors

To evaluate our ranking approaches in the presence of imperfect feedback, we perform experiments that study effectiveness of the methods in different error scenarios as discussed in Section 4.4. We perform experiments on Books and FlightsDay because results were the most promising for these datasets in the previous experiments. Due to space constraints, we present only few of the experiment results.

Conflicting feedback. In this experiment, we assume access to feedback from a crowd of workers who provide correctness of all claims instead of providing a single correct label. We consolidate conflicts of the crowd by varying (1) the fraction of data items that it disagrees on (i.e., the crowd provides correctness of all claims of say, 5% data items), and (2) their consensus on the correct claim for a data item (i.e., 70% probability that the true claim is indeed correct). We



(a) QBC: 10% conflicts. (b) US: 10% conflicts. (c) ApproxMEU: 10% conflicts.

(d) QBC: 30% conflicts. (e) US: 30% conflicts. (f) ApproxMEU: 30% conflicts.

(g) QBC: 50% conflicts. (h) US: 50% conflicts. (i) ApproxMEU: 50% conflicts.

Figure 5: Conflicting feedback (Books). Each row compares methods when $x\%$ of items have conflicting feedback. Lines in a plot compare a method when correctness of the correct claim varies from 0.9 to 0.1.

vary the first parameter from $10\% - 50\%$ and the second from $10\% - 90\%$ and report the results of this experiment in Figure 5. As expected, as the crowd varies its consensus on the correct claim from 90% to 10%, the performance of all the methods consistently deteriorates. `QBC` and `US` start falling apart as the crowd's consensus degrades. The methods with 90% consensus, however, exhibit comparable performance to their no-error counterparts even when the fraction of data items with conflicting feedback increases. On the other hand, `Approx-MEU` demonstrates substantial improvement in fusion even when the consensus goes to 50% on 30% of all data items. It only starts to worsen when the crowd assigns really low probability to the correct claim for 50% of all data items.

Feedback Confidence. We simulate the confidence in feedback as a probability attached to it. This could also be likened to *worker* (or, crowd) quality, e.g., there is only 80% probability that any feedback provided by `Worker A` on a data item is correct. We assume the confidence to be varying from $80\% - 100\%$ and report the results of this experiment in Figure 6. We notice that performance of the methods consistently deteriorates as confidence decreases from 100% to 80%. While with even 90% conviction in feedback, `QBC` and `US` no longer improve fusion on Books, `Approx-MEU` is the most resilient to such feedback errors. Even at 80% confidence, `Approx-MEU` adaptively integrates erroneous input and continues to improve fusion in initial validations (although with diminished power) before tapering off and worsening after $\sim$ 8% of the data items have been validated. `Approx-MEU`$_{.9}$ almost levels out after 10% items are validated, and with `Approx-MEU`$_{.8}$, soliciting feedback after 5% validations does not boost fusion. The net improvement with `Approx-MEU`$_{.8}$ after 15% of items are validated, however, is comparable to that achieved in `QBC` and `US` without any feedback errors.

Incorrect Feedback. We assume the hypothetical case when we have an ineffective user that (either knowingly or unknowingly) provides incorrect answers. We further consider the user to be wrong on $0\% - 30\%$ of data items and report the results in Figure 7. With slight abuse of notation, the subscript with a method is used to represent the fraction of data items that the user is wrong about. We notice that as the fraction of erroneous data items increases, the methods essentially worsen fusion. However, even with 10% of data items judged incorrectly by the user, `QBC` and `approx-MEU` exhibit better performance than `US` without incorrect feedback. This demonstrates that on dense data, identifying items that have high entropy is more beneficial and more resilient to feedback errors than selecting items with `US`.

Takeaways: (1) Among all the approaches, `Approx-MEU` is

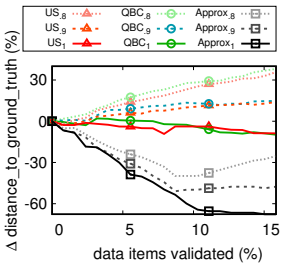


Figure 6: Feedback confidence (Books). Subscript is user confidence (or, worker quality).

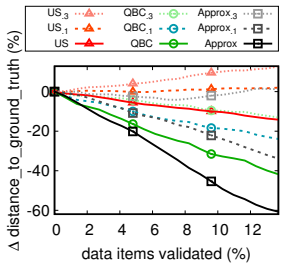


Figure 7: Incorrect Feedback (FlightsDay). Subscript is fraction of items with incorrect feedback.

most robust in the presence of feedback errors. (2) `Approx-MEU` continues to improve fusion even when the feedback is close to incorrect for a small fraction of data items. (3) On dense data, `QBC` is resilient to completely incorrect feedback on a small fraction of all data items.

6. RELATED WORK

Data fusion. The problem of conflict resolution has been extensively studied in the past and a number of techniques have been proposed (see [22] for a survey). Most of the existing fusion systems use Bayesian analysis [41, 7, 15, 36], optimization techniques [23], or probabilistic graphical models [30] to infer correct claims and source reliabilities.

The present work provides a general framework to effectively solicit feedback from a user; the item-level ranking algorithms and the general decision-theoretic algorithm (`MEU`) are applicable to any generic data fusion system. To scale up the framework, we proposed estimating change propagation across data items. While this idea of change propagation is applicable to other variants of fusion, the approximate formulae derived to scale up the framework are specific to the data fusion model; deriving the formulae for other variants would be specific to the details of the particular fusion technique, and is considered future work.

Leveraging user interaction. Concepts from decision theory and active learning have previously been used for user feedback in various data management problems such as schema matching [28, 5], dataspaces [17], entity resolution [13], classification [18], data cleaning [40, 10] and crowdsourcing [4]. Active learning [32] has a close semblance to the present work; however, as in [4], our goal is to identify the correct claims for most data items whereas in active learning, the objective is to learn a good classifier using as few labels as possible.

Solicitation of user feedback has been studied before in the context of conflict resolution [11, 10, 12] where the focus is to primarily use master data along with editing rules and integrity constraints. In contrast, we propose a user feedback framework to be integrated with a standard data fusion algorithm where we focus on minimizing user interaction. The algorithms that constitute the framework do not assume any domain-knowledge constraints and rely only on the structure of interactions between data items. Moreover, by considering <*source*, *item*, *claim*> triples, we lose information on relations among the attributes of a single item. However, by leveraging the dependencies among data items and sources through the structure of their graph, we are able to predict true values of unknown data items from known true values. Besides, master data could be considered a form of user input for our problem; the benefit from incorporating such pre-meditated user input could, however, be less than that achieved when the user is actively involved.

Prior research on estimating parameters in Bayesian networks [35] proposes using a decision-theoretic framework in the active learning setting. Our problem (data fusion) deals with observed and hidden variables, and approximate solutions from [35] cannot be directly applied to our setting.

Crowdsourcing. Ongoing research in collecting input from a crowd of workers [14, 25, 16] is related because of the characteristics of users in the feedback framework. The problem of noisy labels has been extensively studied in [34, 9] that deal with jointly estimating true labels and user quality.

The present work is orthogonal to crowdsourcing in that we do not focus on modeling user behavior to deal with imperfect feedback. In the presence of noisy feedback from a crowd of workers, any of the existing crowdsourcing approaches can be used to obtain the most accurate label for data items and plugged into our feedback framework.

Propagation of changes. The idea of propagating updates has been used in the past for collective entity resolution [1] where similarity between pairs of entities is dynamically updated as evidence from a classification result is propagated to other dependent entities. Such propagation of changes is captured in two stages in the present work: (1) The specific data fusion model studied in the paper iteratively updates correctness of claims and trustworthiness of sources to generate an assignment of correct claims, and (2) To scale up the decision-theoretic framework, `Approx-MEU` explicitly computes the changes in sources from a validation and updates claims that depend on the affected sources.

7. CONCLUSION

This paper proposed a novel pay-as-you-go approach for effectively soliciting feedback from users to resolve conflicts and improve the performance of existing data fusion techniques. To the best of our knowledge, the present work is the first to leverage user feedback in Bayesian-based data fusion models. To judiciously utilize the user, we proposed generating effective ordering of data items for validation. We presented algorithms that assess data items individually by considering their local characteristics, and also proposed a novel decision-theoretic framework that evaluates data items holistically by their ability to improve the performance of fusion. We further devised approximation formulae to scale up the decision-theoretic framework to large-scale datasets, and also explored scenarios in the presence of imperfect feedback.

Our experimental evaluation on real-world datasets confirmed that guided feedback rapidly increases the effectiveness of data fusion. The proposed methods exhibited different behavior for data with different characteristics, and also offered trade-off between effectiveness and efficiency, and the amount of feedback acquired.

8. REFERENCES

[1] I. Bhattacharya and L. Getoor. Collective entity resolution in relational data. *ACM Trans. Knowl. Discov. Data*, 1(1), Mar. 2007.

[2] R. Boim, O. Greenshpan, T. Milo, S. Novgorodov, N. Polyzotis, and W. C. Tan. Asking the right questions in crowd data sourcing. In *ICDE*, pages 1261–1264, 2012.

[3] G. B. E. Chapman and F. A. E. Sonnenberg. *Decision Making in Health Care: Theory, Psychology, and*

Applications. Cambridge University Press, 2003.
[4] X. Chen, Q. Lin, and D. Zhou. Optimistic knowledge gradient policy for optimal budget allocation in crowdsourcing. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 64–72, 2013.
[5] A. Doan, P. Domingos, and A. Y. Halevy. Reconciling schemas of disparate data sources: A machine-learning approach. In *SIGMOD*, pages 509–520, 2001.
[6] X. L. Dong, L. Berti-Equille, Y. Hu, and D. Srivastava. Global detection of complex copying relationships between sources. *PVLDB*, pages 1358–1369, 2010.
[7] X. L. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: The role of source dependence. *PVLDB*, pages 550–561, 2009.
[8] X. L. Dong, L. Berti-Equille, and D. Srivastava. Truth discovery and copying detection in a dynamic world. *VLDB*, 2(1):562–573, Aug. 2009.
[9] P. Donmez, J. G. Carbonell, and J. Schneider. Efficiently learning the accuracy of labeling sources for selective sampling. In *SIGKDD*, pages 259–268, 2009.
[10] W. Fan, F. Geerts, N. Tang, and W. Yu. Conflict resolution with data currency and consistency. *J. Data and Information Quality*, 5(1-2):6:1–6:37, Sept. 2014.
[11] W. Fan, J. Li, S. Ma, N. Tang, and W. Yu. Towards certain fixes with editing rules and master data. *Proc. VLDB Endow.*, 3(1-2):173–184, Sept. 2010.
[12] W. Fan, S. Ma, N. Tang, and W. Yu. Interaction between record matching and data repairing. *J. Data and Information Quality*, 4(4):16:1–16:38, May 2014.
[13] D. Firmani, B. Saha, and D. Srivastava. Online entity resolution using an oracle. *PVLDB*, 9(5):384–395, 2016.
[14] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. Crowddb: Answering queries with crowdsourcing. In *SIGMOD*, pages 61–72, 2011.
[15] A. Galland, S. Abiteboul, A. Marian, and P. Senellart. Corroborating information from disagreeing views. In *WSDM*, pages 131–140, 2010.
[16] N. Q. V. Hung, D. C. Thang, M. Weidlich, and K. Aberer. Minimizing efforts in validating crowd answers. In *SIGMOD*, pages 999–1014, 2015.
[17] S. R. Jeffery, M. J. Franklin, and A. Y. Halevy. Pay-as-you-go user feedback for dataspace systems. In *SIGMOD*, pages 847–860, 2008.
[18] A. Kapoor, E. Horvitz, and S. Basu. Selective supervision: Guiding supervised learning with decision-theoretic active learning. In *IJCAI*, pages 877–882, 2007.
[19] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.
[20] Q. Li, Y. Li, J. Gao, L. Su, B. Zhao, M. Demirbas, W. Fan, and J. Han. A confidence-aware approach for truth discovery on long-tail data. *PVLDB*, pages 425–436, 2014.
[21] X. Li, X. L. Dong, K. Lyons, W. Meng, and D. Srivastava. Truth finding on the deep web: Is the problem solved? *PVLDB*, pages 97–108, 2012.
[22] Y. Li, J. Gao, C. Meng, Q. Li, L. Su, B. Zhao, W. Fan, and J. Han. A survey on truth discovery. *SIGKDD Explor. Newsl.*, 17(2):1–16, Feb. 2016.
[23] Y. Li, Q. Li, J. Gao, L. Su, B. Zhao, W. Fan, and J. Han. On the discovery of evolving truth. In *SIGKDD*, pages 675–684, 2015.
[24] X. Liu, X. L. Dong, B. C. Ooi, and D. Srivastava. Online data fusion. *PVLDB*, 4(11):932–943, Aug. 2011.
[25] B. Mozafari, P. Sarkar, M. Franklin, M. Jordan, and S. Madden. Scaling up crowd-sourcing to very large datasets: A case for active learning. *PVLDB*, pages 125–136, 2014.
[26] R. M. Neal and G. E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*. Springer, 1998.
[27] J. V. Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton Univ. Press, 1944.
[28] Q. V. H. Nguyen, T. T. Nguyen, Z. Miklos, K. Aberer, A. Gal, and M. Weidlich. Pay-as-you-go reconciliation in schema matching networks. In *ICDE*, pages 220–231, 2014.
[29] J. Pasternack and D. Roth. Knowing what to believe(when you already know something). In *COLING*, 2010.
[30] J. Pasternack and D. Roth. Latent credibility analysis. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 1009–1020, New York, NY, USA, 2013. ACM.
[31] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2 edition, 2003.
[32] B. Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning.*, 2012.
[33] C. E. Shannon. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 2001.
[34] V. S. Sheng, F. Provost, and P. G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *SIGKDD*, pages 614–622, 2008.
[35] S. Tong and D. Koller. Active learning for parameter estimation in bayesian networks. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 647–653. MIT Press, 2001.
[36] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher. On truth discovery in social sensing: A maximum likelihood estimation approach. In *IPSN*, pages 233–244, 2012.
[37] S. E. Whang, P. Lofgren, and H. Garcia-Molina. Question selection for crowd entity resolution. *PVLDB*, 6(6):349–360, 2013.
[38] S. E. Whang, D. Marmaros, and H. Garcia-Molina. Pay-as-you-go entity resolution. *TKDE*, pages 1111–1124, 2013.
[39] J. M. Winn and C. M. Bishop. Variational message passing. In *Journal of Machine Learning Research*, pages 661–694, 2005.
[40] M. Yakout, A. K. Elmagarmid, J. Neville, M. Ouzzani, and I. F. Ilyas. Guided data repair. *VLDB*, pages 279–289, 2011.
[41] X. Yin, J. Han, and P. S. Yu. Truth discovery with multiple conflicting information providers on the web. *TKDE*, pages 796–808, 2008.
[42] C. J. Zhang, L. Chen, H. V. Jagadish, and C. C. Cao. Reducing uncertainty of schema matching via crowdsourcing. *PVLDB*, 6(9):757–768, 2013.

APPENDIX

A. DETAILS OF APPROX-MEU

A.1 Change in probabilities of claims of o_j from change in accuracies of sources

Given the change in accuracies of sources because of the validation of data item o_i, our objective is to estimate the changes propagated from the sources to another data item o_j. This part of the analysis involves a short sequence of basic calculus over the formulae described in Section 3 where we estimate the changes at each step by the method of approximation by differentials.

We rewrite Equation (1) as:

$$\frac{1}{p_j^r} = \sum_{v \in V_j} \frac{\prod\limits_{s \in S(v)} \frac{(|V_j|-1)A(s)}{1-A(s)}}{\prod\limits_{s \in S(v_j^r)} \frac{(|V_j|-1)A(s)}{1-A(s)}} \tag{14}$$

and represent each summation term as a function f:

$$f(v_j^r, v) = \frac{\prod_{s \in S(v)} \frac{(|V_j|-1)A(s)}{1-A(s)}}{\prod_{s \in S(v_j^r)} \frac{(|V_j|-1)A(s)}{1-A(s)}} \tag{15}$$

Equation (14), therefore, simplifies to:

$$\frac{1}{p_j^r} = \sum_{v \in V_j} f(v_j^r, v) \tag{16}$$

To compute the change in p_j^r, we estimate the approximate change in each $f(v_j^r, v)$ through a series of steps: take the logarithm of $f(v_j^r, v)$ and obtain the derivative with respect to $A(s)$, thus presenting $\Delta f(v_j^r, v)$ as:

$$\frac{\Delta f(v_j^r, v)}{f(v_j^r, v)} = \sum_{s \in S(v)} \frac{\Delta A(s)}{A(s)(1-A(s))} - \sum_{s \in S(v_j^r)} \frac{\Delta A(s)}{A(s)(1-A(s))} \tag{17}$$

For each of the sources s that vote for o_j, the term $\Delta A(s)$ in Equation (17) takes a value as noted in Equation (9) depending on whether: (i) s supports v_i^t, (ii) s supports a claim other than v_i^t, or (iii) s does not provide any information on o_i. Clearly, if s belongs to the third category, it will not be affected by the validation of o_i.

We compute the change in probability of claim $v_j^r \in V_j$ attributable to the change in probabilities of claims of o_i by taking the derivative of Equation (16):

$$\Delta p_j^r = -(p_j^r)^2 \sum_{v \in V_j} \Delta f(v_j^r, v) \tag{18}$$

The change in probability of claim $v_j^r \in V_j$ because of the validation of data item o_i can, therefore, be expressed as:

$$\Delta p_j^r = -(p_j^r)^2 \sum_{v \in V_j} \left(\frac{\prod_{s \in S(v)} \frac{(|V_j|-1)A(s)}{1-A(s)}}{\prod_{s \in S(v_j^r)} \frac{(|V_j|-1)A(s)}{1-A(s)}} \cdot \left(\sum_{s \in S(v)} \frac{\Delta A(s)}{A(s)(1-A(s))} - \sum_{s \in S(v_j^r)} \frac{\Delta A(s)}{A(s)(1-A(s))} \right) \right) \tag{19}$$

The updated probability $(p_j^r)'$ of claim $v_j^r \in V_j$ is:

$$(p_j^r)' = p_j^r + \Delta p_j^r \tag{20}$$

A.2 Proof of Theorem 4.1

Consider data items o_i and o_j that are more than one hop away from each other, i.e., they are connected via an alternating path of sources and other data items. In this section, we compute through a sequence of steps, the change in probabilities of o_j attributed to the validation of o_i.

First, the change in probabilities of o_i are propagated to sources that provide claims about it. This changes the accuracies of sources: by boosting the accuracy of those that provide a `true` claim and decreasing the accuracy of those that provide an incorrect claim. From Equation (9), if source s provides claim v_i^l about data item o_i, then the accuracy of the source changes as:

$$\Delta A(s) = \frac{\Delta p_i^l}{N(s)}$$

Change in probabilities of o_j. We represent Equation (1) for data item o_j as $p_j^r = q/t$ to obtain:

$$p_j^r t = q = \prod_{s \in S(v_j^r)} \frac{(|V_j - 1|)A(s)}{1-A(s)} \tag{21}$$

We apply the logarithm function to both sides of Equation (21) to simplify the representation for further computation as:

$$\log q = \sum_{s \in S(v_j^r)} \log \frac{(|V_j - 1|)A(s)}{1-A(s)} \tag{22}$$

Next, to compute the change in quantity q, we obtain the first derivative of the expressions in Equation (22) as:

$$\frac{dq}{q} = \sum_{s \in S(v_j^r)} d\left(\log \frac{(|V_j - 1|)A(s)}{1-A(s)} \right) = \sum_{s \in S(v_j^r)} \frac{dA(s)}{A(s)(1-A(s))}$$

and express dq in a cleaner form as:

$$dq = q \left(\sum_{s \in S(v_j^r)} \frac{dA(s)}{A(s)(1-A(s))} \right) \tag{23}$$

We express the change in probabilities of o_j by computing the first derivative of Equation (21):

$$p_j^r(dt) + (dp_j^r)t = dq \tag{24}$$

where t can be expressed as a sum of terms, t_k, similar to q for each $v_j^k \in V_j$. Using Equation (23), Equation (24) can thus be rewritten as:

$$p_j^r \left(\sum_{v_j^k \in V_j} t_k \sum_{s \in S(v_j^k)} \frac{dA(s)}{A(s)(1-A(s))} \right) + (dp_j^r)t = q \left(\sum_{s \in S(v_j^r)} \frac{dA(s)}{A(s)(1-A(s))} \right)$$

We now rearrange the terms appropriately and replace q/t by p_j^r, to express dp_j^r as:

$$dp_j^r = p_j^r \left(\sum_{s \in S(v_j^r)} \frac{dA(s)}{A(s)(1-A(s))} \right) - p_j^r \left(\sum_{v_j^k \in V_j} \frac{t_k}{t} \sum_{s \in S(v_j^k)} \frac{dA(s)}{A(s)(1-A(s))} \right) \tag{25}$$

We are interested in analyzing the upper bound on dp_j to get an estimate of the maximum change that o_i would effect upon o_j. We present a step-by-step conclusion of the same. It follows from Equation (25) that:

$$\begin{aligned}
|dp_j^r| &\leq p_j^r \left| \sum_{s \in S(v_j^r)} \frac{dA(s)}{A(s)(1-A(s))} \right| \\
&\leq p_j^r \sum_{s \in S(v_j^r)} \left| \frac{dA(s)}{A(s)(1-A(s))} \right| \\
&\leq p_j^r |S(v_j^r)| \left| \frac{dA(s)}{A(s)(1-A(s))} \right|_{max} \\
&\leq p_j^r |S(v_j^r)| \left| \frac{dp_i^t}{N(s)A(s)(1-A(s))} \right|_{max} \\
&\leq p_j^r |S(v_j^r)| \left| \frac{dp_i^t}{N'A'(1-A')} \right|_{max}
\end{aligned}$$

where $N' \leq N(s)$ is the least number of data items any source votes for and A' is the accuracy of a source that yields the minimum for function $A(s)(1 - A(s))$.
Real datasets are often faced with the situation of few sources providing information about far too many data items. As a result, N' is usually more than half the number of items in the dataset. This, coupled with p_j, dp and $A'(1 - A')$, contributes to the change in probabilities of a data item one-hop away being much less than the change in the probabilities of the validated data item.

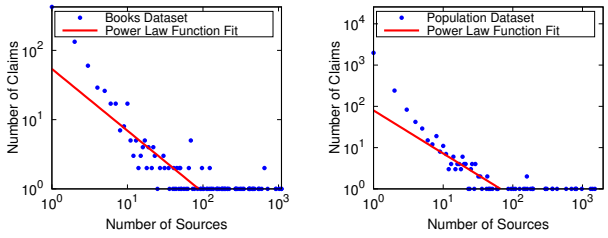
For a data item, o_k, two hops away from the validated node, following similar analysis, if o_k is reachable from o_i through o_j, we reach the conclusion that:

$$|dp_k^l| \leq \left(p_k^l |S(v_k^l)| \left| \frac{dp_j^r}{N' A'(1 - A')} \right|_{max} \right)$$
$$\leq \frac{dp_i^t}{N'^2} \left(\left| \frac{p_k^l p_j^r |S(v_k^l)||S(v_j^r)|}{(A'(1 - A'))^2} \right|_{max} \right)$$

We observe an exponential decay of the changes in probability distributions as we move away from the validated node. More specifically, the changes in probability distributions in the first hop are significantly higher than those from the second hop and so on. This is attributed to the sole reason that a typical source provides information about a large number of data items in the dataset.

B. EXPERIMENTAL EVALUATION

B.1 Dataset Characteristics



(a) Books. (b) Population.

Figure 8: Long-tail characteristics in real data.

As seen in Figure 8, both Books and Population exhibit long-tail characteristics (following the power law phenomenon) where most of the sources provide information on a small fraction of all data items and few sources provide information on a large number of items.

B.2 Relation between performance metrics

In Section 5.1, we conducted experiments to study the feasibility of entropy utility function, in terms of effectiveness and efficiency, as an alternative to the ground-truth-based utility function. We notice that the plots representing the distance to ground truth and those representing the reduction in uncertainty follow the same trend, i.e., as the distance to ground truth decreases, the uncertainty is also reduced. Moreover, the rate of reduction in these two metrics appears to be comparable for `GUB` and `MEU`. Theoretically, we can explain this behavior in one direction: as the database gets closer to ground truth, the data fusion system becomes more certain in its predictions. Therefore, the uncertainty of the database is expected to decrease. On the other hand, as uncertainty decreases, there is no guarantee that the fusion system would fare better in predicting correct claims; it simply might be more certain in wrong predictions.

To better understand the relation between the two metrics, we conducted an experimental study of the metrics for the fundamental methods, `GUB` and `MEU` (since these are our gold standards), on synthetic datasets generated using a number of parameters.

Synthetic Data Generation. Our objective in generating synthetic data is to replicate dense real-world data with $|O| >> |S|$ (typical datasets for data fusion systems, e.g., see [21]). We model most of the sources to be of good quality with few very good and few poor sources. Source accuracies, $A(s_j)$, can therefore be assumed to follow a normal distribution: $A(s_j) \sim N(a_{mean}, a_{sd})$ where a_{mean} is the average accuracy and a_{sd} is the standard deviation of the source accuracies. Density of the data, i.e. the probability that a source votes for a data item, is specified by d. The default values for the parameters, $a_{mean} = 0.8$, $a_{sd} = 0.1$ and $d = 0.4$, correspond to the characteristics of real datasets. Source S_j provides a claim for data item o_i with probability d and the claim is correct with a probability $A(S_j)$.

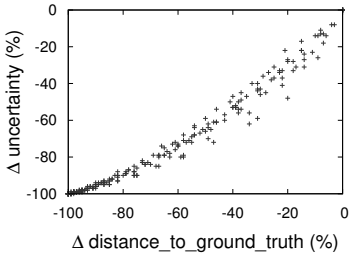


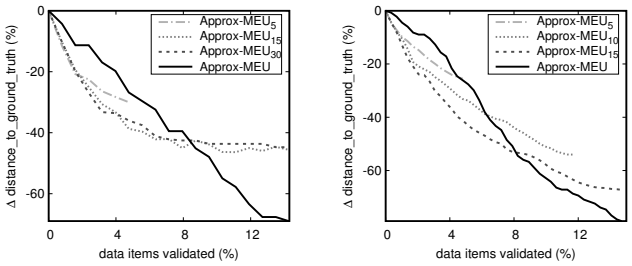
Figure 9: Scatter plot showing relation between the performance metrics.

Observation. As seen in Figure 9, we observe empirically that the distance to ground truth and uncertainty are strongly correlated. This study is further supported by the Pearson's correlation coefficient, $\rho = 0.86$. For FlightsDay, $\rho = 0.71$ and for Books, $\rho = 0.72$, indicating a moderately positive correlation. Specifically, uncertainty in the fusion predictions and their distance to ground truth go hand in hand. This additionally confirms the suitability of entropy utility as an alternative to ground truth utility function.

B.3 Exploring Approx-MEU

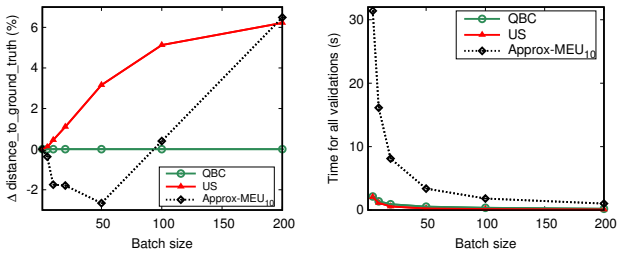
As mentioned in Section 4.2.3, in the worst case, `Approx-MEU` mandates an all-pairs computation of the impact of data items on each other — still expensive in datasets where all data items are connected to each other. In Section 4.3, we discussed optimizations to reduce the computation cost by shrinking the search space for the impact computation step; we now explore the effect of this approach i.e., the role of k in `Approx-MEU`$_k$, on the improvement in data fusion.

Effectiveness. Figure 10 demonstrates the various degrees of improvement offered by `Approx-MEU`$_k$ as k varies. Subscript k denotes the fraction of all data items considered for impact computation. When $k = 5$, we consider only the top 5% data items ranked first according to their vote entropies and then, in the order of their entropies over probabilities of claims. We compute only the impact of these 5% data items on each other; evidently, the line ends when 5% of all data items are validated. We observe that as k increases, more



(a) Books. (b) FlightsDay.

Figure 10: Hybrid approach combining QBC and Approx-MEU. Figures depict the effect of expanding the set of candidates for validation in Approx-MEU.



(a) Effectiveness. **(b)** Efficiency.

Figure 11: Effect of batch size on effectiveness of the methods and time taken to validate 200 claims from FlightsDay.

data items are considered in the impact computation step and the system converges to ground truth faster. Approx-MEU, while less effective in the beginning, gradually surpasses the improvement in fusion achieved by the Approx-MEU$_k$ methods. The plots indicate that for early validations (less than 8% of items validated), choosing as small a value as $k = 30$ (Books) or $k = 15$(FlightsDa y) results in better conflict resolution than Approx-MEU; by tuning k, we can effectively scale up the decision-theoretic framework with estimated probabilities to large datasets.

Efficiency. We report in Table 12 the time taken for one validation on the three datasets by QBC, US and Approx-MEU$_k$ with different values of k. As expected, with an increase in k, as more data items are considered for impact computation, Approx-MEU$_k$ takes longer to determine the next action. However, for the large-scale Flights data, Approx-MEU has a significantly rapid convergence to ground truth than QBC and US in slightly more than 5 minutes.

time(sec)	Books	FlightsDay	Flights
QBC	0.08	0.07	6.0
US	0.09	0.12	1.8
Approx-MEU$_5$	0.04	0.23	156
Approx-MEU$_{10}$	0.09	0.73	323
Approx-MEU$_{15}$	0.15	0.98	475

Table 12: Time taken (in seconds) by QBC, US and Approx-MEU$_k$ with different values of k.

B.4 Effect of Batch Size

Based on our intuitions about batch size (Section 4.2.2), we now study the effect of validating multiple data items simultaneously on the performance of our methods.

Effectiveness. As shown in Figure 11a, performance of QBC is not affected by batch size because by selecting data items based on their vote entropies, at the end of 200 actions, the set of validated data items remains unchanged.

With an increase in the batch size, the distance to ground truth steadily increases for US because by validating multiple data items at once, it loses the opportunity to adaptively integrate the acquired feedback.

Approx-MEU displays an interesting behavior: the method converges to ground truth faster with an initial increase in batch size, and after batchSize= 50, its performance worsens. The initial improvement is because with smaller batches, the algorithm selects data items having high entropy (e.g., entropy > 0.67); as the batch size increases, the algorithm selects data items with a mix of high and medium entropies (e.g. entropy > 0.6). By not ordering data items with medium entropies correctly, the performance of the method deteriorates with an increase in batch size.

Efficiency. We observe in Figure 11b that the time taken by QBC and US, after sorting, is effectively the time taken to fuse the data. As more data items are validated together, the fusion system reaches a steady state faster and the methods have almost flat gain in the time for all validations. Going from a batchSize of 1 to 200, the runtime of Approx-MEU, however, reduces by more than one order of magnitude. Specifically, for FlightsDay, we observe that a batchSize= 50 achieves the best improvement in fusion in about one-sixth the time taken by validating individual data items.

Takeaways: Increasing the batch size: (1) has no effect on QBC while it typically degrades performance of US and Approx-MEU (although the latter shows improvement with smaller increase in batch size), and (2) drastically reduces the time taken for validations by ApproxMEU.

Takeaways: (1) By limiting the fraction of data items for the impact computation step, Approx-MEU can be efficiently scaled up to large datasets. (2) Different values of k offer trade-offs between effectiveness and efficiency. Specifically, the smaller the value of k, the faster it takes to determine the next action although a method with a higher k rapidly converges to ground truth.