Less is More: Selecting Sources Wisely for Integration

Xin Luna Dong
AT&T Labs-Research
lunadong@research.att.com

Barna Saha
AT&T Labs-Research
barna@research.att.com

Divesh Srivastava
AT&T Labs-Research
divesh@research.att.com

ABSTRACT

We are often thrilled by the abundance of information surrounding us and wish to integrate data from as many sources as possible to improve the quality of the integrated data. However, understanding, analyzing, and using these data are often hard. *Too much* data can introduce a huge integration cost, such as expenses for purchasing data and resources for integration and cleaning. Furthermore, including low-quality data can even deteriorate the quality of integration results instead of bringing the desired quality gain. Thus, "the more the better" does not always hold for information and often "less is more".

In this paper, we study how to select a subset of sources before integration such that we can balance the quality of integrated data and integration cost. Inspired by the *Marginalism* principle in economic theory, we wish to stop integrating a new source when the marginal gain, often a function of improved integration quality, is less than the marginal cost, associated with data-purchase expense and integration resources. As a first step towards this goal, we focus on *data fusion* tasks, where the goal is to resolve conflicts from different sources. We propose a randomized solution for selecting sources for fusion and show empirically its effectiveness and scalability on both real-world data and synthetic data.

1. INTRODUCTION

1.1 Motivation

The Information Era has witnessed not only a huge volume of data, but also a huge number of sources or data feeds from websites, Twitter, blogs, online social networks, collaborative annotations, social bookmarking, data markets, and so on. The abundance of useful information surrounding us and the advantage of easy data sharing have made it possible for data warehousing and integration systems to improve the quality of the integrated data. For example, with more sources, we can increase the coverage of the integrated data; in the presence of inconsistency, we can improve correctness of the integrated data by leveraging the collective wisdom. Such quality improvement allows for more advanced data analysis and can bring a big *gain*. However, we also need to bear in mind that data collection and integration come with a *cost*. First, many data sources, such as *GeoLytics* for demographic data[1], *WDT* for weather data[2], *GeoEye* for satellite imagery[3], *American Business Database* for business listings[4], charge for their data. Second, even for sources that are free, integration requires spending resources on mapping heterogeneous data items, resolving conflicts, cleaning the data, and so on. Such costs can also be huge. Actually, the cost of integrating some sources may not be worthwhile if the gain is limited, especially in the presence of redundant data and low-quality data. We next use a real-world example to illustrate this.

EXAMPLE 1.1. *We consider a data set obtained from an online bookstore aggregator,* AbeBooks.com[5]. *We wish to collect data on CS books. There are 894 bookstores (each corresponding to a data provider), together providing 1265 CS books. They identify a book by its ISBN and provide the same attributes. We focus on* coverage *(*i.e.*, the number of provided books) and define it as the gain.*

We processed the sources in decreasing order of their coverage (note that this may not be the best ordering if we consider in addition overlaps between the sources) and reported the total number of retrieved books after probing each new source. Fig.1 plots for the first 100 sources. We observe that the largest source already provides 1096 books (86%), and the largest two sources together provide 1213 books (96%). We obtained information for 1250 books, 1260 books and all 1265 books after integrating data from 10, 35 and 537 sources respectively. In other words, after integrating the first 537 sources, the rest of the sources do not bring any new gain.

Now assume we quantify the cost of integrating each source as 1. Then, integrating the 11th to 537th sources has an extra cost of $537 - 10 = 527$ *but an additional gain of only* $1265 - 1250 = 15$. *Thus, if we are willing to tolerate a slightly lower coverage, it is even not worthwhile to integrate all of the first 537 sources.* □

This example shows that integrating new sources may bring some gain, but with a higher extra cost. Even worse, some low-quality sources may even hurt the quality of the integrated data and bring a negative gain, as we illustrate next.

EXAMPLE 1.2. *Continue with the same data set. We observed that different sources can provide quite different titles and author lists for the included books. Take author lists as an example. Even*

[1]http://www.geolytics.com/.
[2]http://www.wdtinc.com/.
[3]http://www.geoeye.com/.
[4]http://www.customlists.net/databases/american.
[5]We thank authors of [22] for providing us the data.

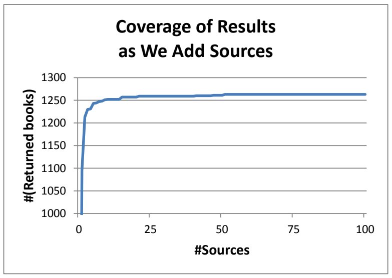


Figure 1: Coverage of results.

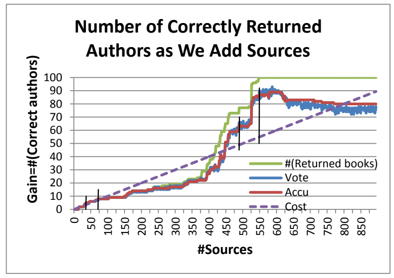


Figure 2: Returned correct results.

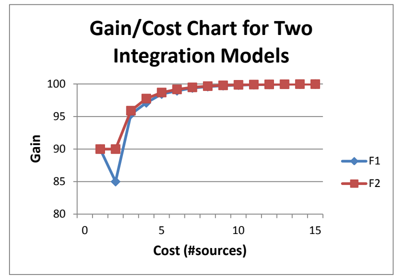


Figure 3: Different integration models.

after we normalized the author lists to a standard format and ignored middle names, each book has 1 to 23 different provided author lists and the number is 4 on average. Mistakes include missing authors, additional authors, mis-ordering of the authors, misspelling, incomplete names, etc. For evaluation purpose, we examined 100 randomly selected books and manually checked book covers for obtaining the correct author list as the gold standard.

Ideally, we would like to find the correct author list from conflicting values. We did this in two ways. First, VOTE *applies the voting strategy and chooses the author list provided by the largest number of sources. Second,* ACCU *considers in addition the accuracy of the sources: it takes the accuracy of each source as input (computed by the percentage of correctly provided values for the books inside the gold standard), assigns a higher vote to a source with a higher accuracy, and chooses the author list with the highest sum of the votes (details in Section 3).*

We considered the sources in decreasing order of their accuracy (this is just for illustration purpose and we discuss ordering of sources in Section 1.3). Fig.2 plots the gain, defined as the number of correctly returned author lists for these 100 books, as we added each new source. We observed that we obtained all 100 books after processing 548 sources (see the line for #(Returned books)). The number of correct author lists increased at the beginning for both methods; then, VOTE *hits the highest number, 93, after integrating 583 sources, and* ACCU *hits the highest number after integrating 579 sources; after that the numbers decreased for both methods and dropped to 78 and 80 respectively for* VOTE *and* ACCU. *In other words, integrating the 584th to 894th sources has a negative gain for* VOTE *and similar for* ACCU. □

This example shows that for data, "the more the better" does not necessarily hold and sometimes "less is more". As the research community for data integration has been focusing on improving various integration techniques, which is important with no doubt, we argue that it is also worthwhile to ask the question whether integrating all available data is the best thing to do. Indeed, Fig.2 shows that although in general the more advanced method, ACCU, is better than the naive method, VOTE, the result of ACCU on all sources is not as good as that of VOTE on the first 583 sources. This question is especially relevant in the big data environment: not only do we have larger volume of data, but also we have larger number of sources and more heterogeneity, so we wish to spend the computing resources in a wise way. This paper studies how we can select sources wisely before real integration or aggregation such that we can balance the gain and the cost. Source selection can be important in many scenarios, ranging from Web data providers that aggregate data from multiple sources, to enterprises that purchase data from third parties, and to individual information users who shop for data from data markets [1].

1.2 Source selection by Marginalism

Source selection in the planning phase falls in the category of resource optimization. There are two standard ways to formalize the problem: finding the subset of sources that maximizes the result quality under a given budget, or finding the subset that minimizes the cost while satisfying a minimal requirement of quality. However, neither of them may be ideal in our context, as shown next.

EXAMPLE 1.3. *Consider* ACCU *in Fig.2. Assume* only for simplicity *that the applied order is the best for exploring the sources. Suppose the budget allows integrating at most 300 sources; then we may select all of the first 300 sources and obtain 17 correct author lists. However, if we select only the first 200 sources, we can cut the cost by 1/3, while obtaining only 3 fewer correct author lists (17.6% fewer); arguably, the latter selection is better. On the other hand, suppose we require at least 65 correct author lists; then we may select the first 520 sources, obtaining 65 correct lists. However, if we instead select 526 sources, we introduce 1% more cost but can obtain 81 correct lists (improving by 25%); arguably, spending the few extra resources is worthwhile.* □

We propose a solution inspired by the *Marginalism* principle in economic theory [11]. Assuming we can measure gain and cost using the same unit (many enterprises do predict revenue and expense from integration in dollars according to some business models), we wish to stop selecting a new source when *the marginal gain is less than the marginal cost*. Here, the marginal gain is the difference between the gain after and before integrating the new source and similar for marginal cost. In our example, if the gain of finding one more correct author list is 10 times the cost of integrating one new source, the 489th source is such one marginal point.

1.3 Challenges for data integration

Source selection falls outside the scope of traditional integration tasks, such as mapping schemas, linking records that refer to the same real-world entity, and resolving conflicts. On the one hand, it is a prelude for data integration. On the other hand, how we select the sources would be closely related to the integration techniques we apply. Applying the Marginalism principle to source selection for data integration faces many challenges.

First, in economic theory *the Law of Diminishing Returns* [11] (*i.e.*, keeping adding resources will gradually yield lower per-unit returns) often holds and so we can keep adding resources until the marginal cost exceeds the marginal gain for the next unit of resource. However, the Law of Diminishing Returns does not necessarily hold in data integration, so there can be multiple marginal points. In our example (Fig.2), after we integrate 71 sources by ACCU, the gain curve flattens out and the marginal gain is much less than the marginal cost; however, starting from the 381st source, there is a sharp growth for the gain. Indeed, there are four marginal points on the curve: the 35th, 71st, 489th, and 548th sources (marked by vertical lines in Fig.2). We thus need to find all marginal points before we stop investigation.

Second, the data sources are different, providing data with different coverage and quality, so integrating the sources in different orders can lead to different quality curves (so gain curves). Each curve has its own marginal points, so we need to be able to compare all marginal points in some way and choose one as the best.

Third, applying Marginalism requires estimation of integration cost and gain. The gain is often associated with quality of the integrated data and so can be hard to estimate, as we do not know integration quality before we purchase data and apply real integration. There can be multiple quality measures (*e.g.*, coverage, accuracy, freshness, consistency, redundancy) and they can be affected by many aspects of integration, including the specific models applied for schema mapping, entity resolution, and conflict resolution, heterogeneity between the sources, and certainly also quality of the sources. We need a way to estimate integration quality, either by sampling, or by applying analysis on profiles of the sources. However, estimation on sampled data would require coordination between the sources, such as sampling on the same instances.

1.4 Our contributions in the fusion context

As a first step towards source selection, this paper focuses on the *data fusion* aspect; that is, resolving conflicts from different sources for *offline* data integration, as we illustrated in Ex.1.2. In particular, we make the following contributions.

First, we formalize several optimization goals for source selection, including the one that follows the Marginalism principle. Since each marginal point intuitively implies a locally maximum *profit* (*i.e.*, difference between gain and cost), we set the goal as to select *a subset of sources that brings the highest profit*. Since accuracy is the main measure for fusion quality, we define the gain as a function of fusion accuracy (Section 2).

Second, we identify several properties that can affect source selection, where the most important is *monotonicity*–adding a source should never decrease fusion accuracy. We revisit various fusion models [4], showing that none is monotonic, and propose a new model that satisfies monotonicity (Section 3).

Third, we show that for most fusion models, we are able to estimate resulting accuracy based *purely* on the accuracy of the input sources. We propose efficient estimation algorithms (Section 4).

Fourth, we show that in the context of data fusion source selection can be very tricky and a greedy algorithm can generate an arbitrarily bad solution. We show NP-completeness of the problems and propose a heuristic randomized approach that can efficiently approximate the optimal selection (Section 5). For our example in Section 1.2, our algorithm decided in a few minutes that the best solution is to select 26 sources that are estimated to output 97 correct author lists in the gold standard, so the profit is $97-.1*26 = 94.4$, higher than the highest profit from marginal points for the particular order in Fig.2 ($87 - 548 * .1 = 32.2$)

Finally, we conduct experiments showing that 1) when the cost is zero, we are able to find the best subset of sources that maximizes the accuracy of fusion; 2) otherwise, we can efficiently find nearly the best set of sources for fusion (Section 7).

Our results apply when data inconsistency is the major issue, such as for AbeBooks data, and can also apply by considering mistakes in schema mapping or entity resolution as wrongly provided data. In general, there are a lot of open problems, such as considering quality measures other than accuracy, resolving heterogeneity at the schema level and the instance level, and applying the techniques in various environments for warehousing and data integration. We describe one particular extension regarding coverage of the results in Section 6, and discuss the many open directions and lay out a research agenda in Section 9.

Table 1: Notations in this paper.

Notation	Meaning
$\mathcal{S}$	The set of sources in the data set.
S	A single source.
$V(S)$	Coverage of source S.
$A(S)$	Accuracy of source S.
$R(S)$	Recall of source S.
$C(S)$	Cost of source S.
$\bar{S}$	A set of sources.
F	Integration or fusion model.
$F(\bar{S})$	Results of applying F on $\bar{S}$.
$A(F(\bar{S}))$	Accuracy of results of applying F on $\bar{S}$.
$\hat{A}(F(\bar{S}))$	Estimated accuracy of results of applying F on $\bar{S}$.
$C_F(\bar{S})$	Cost of integrating sources in $\bar{S}$ by model F.
$G_F(\bar{S})$	Gain of integrating sources in $\bar{S}$ by model F.
$\mathcal{D}$	The set of data items in consideration.
$\Psi_D(\bar{S})$	Data provided by $\bar{S}$ on data item D.
p	Popularity of top-1 false value.

2. PROBLEM DEFINITION

This section first formally defines the source-selection problem and then instantiates it for the data fusion task. Notations used in this paper are summarized in Table 1.

2.1 Source selection

We consider integration from a set of data sources $\mathcal{S}$. We assume the data integration systems have provided functions that measure the *cost* and *gain* of integration. The cost is related to the expense of purchasing data from a particular source, the resources required for integration, cleaning, manual checking, etc., or any other foreseeable expense for data integration. The gain is typically decided by the quality of the integration results such as the coverage or the accuracy of the integrated data. Many enterprises apply business models to predict cost and gain in money unit (*e.g.*, US Dollars) respectively as the expense of integration and the revenue from integrated data with a certain quality; for example, one may estimate that obtaining data of 50% coverage can bring a gain (revenue) of $1M$ while obtaining data of 90% coverage attract more users and bring a gain of $100M$. The cost and gain can be different when we apply different integration models; we thus denote by $C_F(\bar{S})$ and $G_F(\bar{S})$ the cost and gain of integrating sources in $\bar{S} \subseteq \mathcal{S}$ by model F respectively. Here, F can be one or a set of integration models including schema-mapping models, entity-resolution models, data-fusion models, and so on. We assume that the cost is monotonic; that is, if $\bar{S} \subset \bar{S}'$, $C_F(\bar{S}) \leq C_F(\bar{S}')$ for any F; however, as we have shown in our motivating example, the gain is not necessarily monotonic as the resulting quality may not increase monotonically.

Ideally, we wish to maximize the gain while minimizing the cost; however, achieving both goals at the same time is typically impossible. A traditional approach is to set a requirement on one goal while optimizing the other. Accordingly, we can define the following two optimization problems.

DEFINITION 2.1. *Let $\mathcal{S}$ be a set of sources, F be an integration model, and τ_c be a budget on cost.*

- *The* MAXGLIMITC *problem finds a subset $\bar{S} \subseteq \mathcal{S}$ that maximizes $G_F(\bar{S})$ under constraint $C_F(\bar{S}) \leq \tau_c$.*
- *The* MINCLIMITG *problem finds a subset $\bar{S} \subseteq \mathcal{S}$ that minimizes $C_F(\bar{S})$ under constraint $G_F(\bar{S}) \geq \tau_g$.* □

As our analysis in Ex.1.3 shows, none of these two constrained optimization goals is ideal. Inspired by the Marginalism principle, we wish to stop integrating a new source when the marginal gain is less than the marginal cost; accordingly, we look for *a set of sources whose profit (*i.e.*, gain-cost) is the largest*, assuming the same unit is used for cost and gain. If investing infinitely is unrealistic, we can also apply a budget constraint, but unlike in MAXGLIMITC, the budget constraint is not required for balancing gain and cost. We thus define another source-selection goal as follows.

DEFINITION 2.2 (PROBLEM MARGINALISM). *Let $\mathcal{S}$ be a set of sources, F be an integration model, and τ_c be a budget on cost. The* MARGINALISM *problem finds a subset $\bar{S} \subseteq \mathcal{S}$ that maximizes $G_F(\bar{S}) - C_F(\bar{S})$ under constraint $C_F(\bar{S}) \leq \tau_c$.* □

EXAMPLE 2.3. *Consider a set $\mathcal{S}$ of 15 sources, among which one, denoted by S_0, has a high quality and the others have the same lower quality. Consider two integration models F_1 and F_2, under which each source has a unit cost. Fig.3 shows the gain of applying each model first on S_0 and then in addition on other sources.*

If we set $\tau_c = 15$, MAXGLIMITC *would select all sources on both models, with profit $99.98 - 15 = 84.98$. If we set $\tau_g = 90$,* MINCLIMITG *would select S_0 on both models, with profit $90 - 1 = 89$. Instead,* MARGINALISM *selects S_0 and 4 other sources on model F_1 and obtains a profit of $98.5 - 5 = 93.5$; it selects S_0 and 3 others on model F_2 and obtains a profit of $97.8 - 4 = 93.8$. Obviously,* MARGINALISM *can obtain higher profit than the other two approaches.* □

Solving any of these problems requires efficiently estimating the cost and gain. For cost, we assume that $C_F(\bar{S}) = \sum_{S \in \bar{S}} C(S)$ for any F; it is monotonic and typically holds in practice. The gain depends on the quality measure. In this paper we instantiate it as a function of the accuracy in data fusion, which we review next.

2.2 Data fusion and accuracy estimation

Data fusion: We consider a set of *data items* $\mathcal{D}$, each of which describes a particular aspect of a real-world entity in a domain, such as the name of a book or a director of a movie. A data item can be considered as an attribute of a record, or a cell in a relational table. We assume that each item is associated with a *single* true value that reflects the real world. On the other hand, we consider a set of data sources $\bar{S}$, each providing data for a subset of items in $\mathcal{D}$. We consider only *"good"* sources, which are more likely to provide a true value than a *particular* false value. We assume we have mapped schemas and linked records for the same real-world entity by applying existing techniques. However, different sources may still provide different values for the same data item. *Data fusion* aims at resolving such conflicts and finding the true value for each data item.

There are many fusion models. A basic one, called VOTE, takes the value provided by the largest number of sources. Advanced methods consider source trustworthiness and give higher weights to votes from more trustworthy ones [3, 8, 15, 16, 22, 23, 24]. In this paper we focus on fusion methods that select a single true value for each provided data item. We denote a particular fusion method also by F and its result on a set of sources $\bar{S}$ by $F(\bar{S})$ [6].

We measure *fusion accuracy* by the percentage of correctly returned values over all returned values and denote it by $A(F(\bar{S}))$. An important property that can affect source selection is *monotonicity*, requiring that adding a source at least will not deteriorate the quality of the fusion result. We formally defined it next.

DEFINITION 2.4 (MONOTONICITY). *A fusion model F is* monotonic *if for any $\bar{S} \subset \bar{S}' \subseteq \mathcal{S}$, we have $A(F(\bar{S})) \leq A(F(\bar{S}'))$.* □

[6]It is easy to prove that VOTE and most advanced fusion models are *order independent*; that is, the fusion result is independent of the order in which we consider the sources.

EXAMPLE 2.5. *Consider data items stating gender of people. Consider three independent sources S_1, S_2, S_3 with accuracy .9, .6, and .6, respectively. Obviously, when we integrate only S_1, the accuracy of the result is that of S_1's accuracy, .9.*

Now consider applying VOTE *on all of the three sources to decide the gender of each person. We obtain the correct gender in two cases: 1) all sources provide the correct gender (the probability is $.9*.6*.6 = .324$); 2) two of the sources provide the correct gender (the probability is $.9*.6*.4 + .9*.4*.6 + .1*.6*.6 = .468$). Thus, the accuracy of the result is $.324 + .468 = .792 < .9$, lower than that of integrating only S_1. So* VOTE *is not monotonic.* □

Gain function: We define the gain of integrating $\bar{S}$ based on the accuracy of fusing sources in $\bar{S}$; in the rest of the paper we abuse notation and denote by $G(A)$ the gain of obtaining fusion accuracy A, and by $G(A(F(\bar{S})))$ the gain of fusing $\bar{S}$ by model F. We require the gain to be monotonic with respect to fusion accuracy; that is, if $A < A'$, $G(A) \leq G(A')$. Note however that if we apply a fusion model that is not monotonic, the gain does not increase monotonically as we add more sources; both Ex.1.2 and Ex.2.5 are examples of reducing gain. When $C(S) = 0$ for each $S \in \mathcal{S}$, the MARGINALISM problem reduces to *finding the set of sources that maximizes fusion accuracy*, which is interesting in its own right.

Accuracy estimation: According to the gain function, source selection requires estimating fusion accuracy without knowing (all) real data. In fact, we can estimate it purely from source accuracy and the distribution of false values (we explain in Section 4 the information we need for the distribution); both of them can be sampled on a small subset of data according to manually decided gold standard. The advantage of such estimation over measuring fusion accuracy directly on sampled data is that the latter would require much more co-ordination between sources in sampling, as we stated in Section 1.3. We formally define the problem as follows.

DEFINITION 2.6 (ACCURACY ESTIMATION). *Let $\bar{S}$ be a set of sources, $A(S)$ denote the accuracy of $S \in \bar{S}$, $\overline{pop}$ be the distribution of false values, and F be a fusion model.* Accuracy estimation *estimates the accuracy of $F(\bar{S})$, denoted by $\hat{A}(F(\bar{S}))$.* □

This paper assumes *independence of sources* and that *the data items are not distinguishable in terms of error rate and false-value distribution*. We begin with considering only *full-coverage* sources (Section 3-5) and then extend our work by considering coverage of the sources (Section 6). Experimental results show effectiveness of our techniques in general when the assumptions do not hold (Section 7), and we leave a more extensive study in presence of source dependency for future work.

3. PROPERTIES OF FUSION MODELS

This section starts with reviewing the models presented in recent work, showing that none of them is monotonic in general. We then propose a model that considers both the accuracy of the sources and the distribution of the provided values, and show that it is monotonic for independent sources.

3.1 Existing fusion models

VOTE chooses among conflicting values the one that is provided by the most sources. As shown in Ex.2.5, it is not monotonic.

THEOREM 3.1. *The* VOTE *model is order independent but not monotonic.* □

VOTE is not monotonic because it can be biased by values provided by less accurate sources. Recent work [3, 8, 15, 16, 22, 23] considered source accuracy in fusion. We next review the model presented in [3], named ACCU; other work follows the same spirit and has similar properties.

ACCU applies Bayesian analysis. If we denote the value provided by S on data item D by $\Psi_D(S)$ and the vector of values from $\bar{S}$ by $\Psi_D(\bar{S})$, ACCU computes $Pr(v \text{ true}|\Psi_D(\bar{S}))$ for each value in the domain and chooses the one with the highest probability as true. According to the Bayes rule, it only needs to compare the inverse probability $Pr(\Psi_D(\bar{S})|v \text{ true})$ for the provided values.

ACCU assumes that (1) there are n false values for a data item in its domain and (2) these false values are equally likely to be provided by a source. Now consider the probability that source S provides $\Psi_D(S)$ on D. If $\Psi_D(S)$ is the correct value, the probability is $A(S)$; otherwise, the probability becomes $\frac{1-A(S)}{n}$. If we denote by $\bar{S}(v)$ the providers of v, under the independence assumption,

$$Pr(\Psi_D(\bar{S})|v \text{ true}) = \Pi_{S\in\bar{S}(v)} A(S) \cdot \Pi_{S\in\bar{S}\setminus\bar{S}(v)} \frac{1-A(S)}{n} \quad (1)$$

$$= \Pi_{S\in\bar{S}(v)} \frac{nA(S)}{1-A(S)} \cdot \Pi_{S\in\bar{S}} \frac{1-A(S)}{n}. \quad (2)$$

In this equation, $\Pi_{S\in\bar{S}}\frac{1-A(S)}{n}$ is the same for all values. Thus, we compute the *accuracy score* of S as $\alpha(S) = \ln\frac{nA(S)}{1-A(S)}$ and compare the *confidence* of each value, computed by

$$C(v) = \sum_{S\in\bar{S}(v)} \alpha(S). \quad (3)$$

ACCU is obviously order independent. It improves over VOTE in that it gives a less accurate source a lower vote count. However, ACCU is monotonic if and only if the above two assumptions hold.

THEOREM 3.2. ACCU *is order independent; it is monotonic if and only if there are* n *uniformly-distributed false values.* □

PROOF. (1) *If:* Let v_t be the true value and v be a false value. Consider the ratio $R = \frac{Pr(\Psi_D(\bar{S})|v_t \text{ true})}{Pr(\Psi_D(\bar{S})|v \text{ true})}$. Proving the *if* part is equivalent to showing that R does not decrease when we add a new source S. Under the assumptions, source S has probability $A(S)$ to provide the correct value v_t, and probability $\frac{1-A(S)}{n}$ to provide the false value v. According to Eq.(1), the new ratio is

$$R' = R \cdot \frac{A(S)^{A(S)} \cdot (\frac{1-A(S)}{n})^{1-A(S)}}{A(S)^{\frac{1-A(S)}{n}} (\frac{1-A(S)}{n})^{1-\frac{1-A(S)}{n}}}$$
$$= R \cdot (\frac{nA(S)}{1-A(S)})^{A(S)-\frac{1-A(S)}{n}}.$$

For "good" sources, $A(S) > \frac{1-A(S)}{n}$ and so $R' > R$.

(2) *Only if:* For any $n' \neq n$ false values and for any non-uniform distribution of n false values, we are able to construct a counter example containing 3 sources.

I. First consider the case when the n false values are not uniformly distributed. The counter example contains three sources: S_1 with accuracy $a > .5$, and S_2 and S_3 with the same accuracy $b = \frac{\sqrt{aX}}{\sqrt{1-a}+\sqrt{aX}} - \epsilon$, where $X = \sum_{i=1}^n pop(v_i)^2$, $pop(v_i)$ is the popularity of a false value v_i among all false values ($\sum_{i=1}^n pop(v_i) = 1$), and ϵ is a very small number. Note that this construction does not rely on n. We now prove that (a) S_2 and S_3 are "good" sources; (b) the sum of vote counts of S_2 and S_3 is larger than that of S_1; and (c) the accuracy of fusing $S_1 - S_3$ is lower than a, the accuracy of fusing only S_1.

To prove (a), we prove that $b > p_m(1-b)$, where p_m is the highest popularity among false values. This is equivalent to proving $\sqrt{\frac{aX}{1-a}} > p_m$, which can be derived from $\frac{aX}{1-a} > X > p_m^2$.

To prove (b), we prove that $2ln\frac{nb}{1-b} > ln\frac{na}{1-a}$; in other words, $(\frac{nb}{1-b})^2 > \frac{na}{1-a}$. This is equivalent to $nX > 1$, which can be derived from the fact that the false values are not uniformly distributed.

To prove (c), let us first enumerate the several possible worlds where we shall output the correct value. First, one of S_2 and S_3 provides the correct value and S_1 provides the correct value; the probability is $2ab(1-b)$. Second, S_1 provides the correct value and S_2 and S_3 provide different false values; the probability is $a(1-b)^2(1-X)$. Finally, both S_2 and S_3 provide the correct value; the probability is b^2. Thus, the accuracy is $b^2 + 2ab(1-b) + a(1-b)^2(1-X)$. We can prove it is less than a, which is equivalent to $(\frac{b}{1-b})^2 < \frac{aX}{1-a}$, derivable from $b < \frac{\sqrt{aX}}{\sqrt{1-a}+\sqrt{aX}}$.

II. Next consider the case when $n' \neq n$. The case of $n' < n$ is essentially the same as n values not uniformly distributed. For the case of $n' > n$, the counter example contains three sources: S_1 with accuracy $\frac{1}{1+n'} < a < \frac{n'}{n'+n^2}$, S_2 and S_3 with accuracy $b = \frac{1-a}{1-a+an^2}$. We now prove that (a) the sources are all "good" sources; (b) the sum of the vote counts of S_1 and S_2 is 0 and the vote count of S_2 and S_3 is negative; and (c) the accuracy of fusion $S_1 - S_3$ is lower than a, the accuracy of fusing only S_1.

To prove (a), we need to show for S_2 and S_3 that $b > \frac{1}{1+n'}$, which is equivalent to $\frac{an^2}{1-a} < n'$, derivable from $a < \frac{n'}{n'+n^2}$. S_1 is obvious "good".

To prove (b), we need to show that $\ln\frac{na}{1-a} + \ln\frac{nb}{1-b} = 0$, derivable from $b = \frac{1-a}{1-a+an^2}$. From it we also see that $\ln\frac{nb}{1-b} < 0$.

To prove (c), let us again enumerate the several possible worlds where we shall output the correct value. First, one of S_2 and S_3 provides the correct value and S_1 provides the correct value; the probability is $2ab(1-b)$. Second, S_1 provides the correct value and S_2 and S_3 provide false values (different or the same); the probability is $a(1-b)^2$. Note that when all sources provide the correct value, according to (b), we will compute negative vote count for the correct value and so do not output it as correct. Thus, the accuracy is $2ab(1-b) + a(1-b)^2 < a$. □

3.2 Considering value distribution in fusion

With the assumption that false values are uniformly distributed, ACCU computes a low probability for providing a *particular* false value and so can make mistakes in presence of very popular false values. We now describe POPACCU, a refinement of the ACCU model, with the following two desired features: 1) POPACCU does not assume any a-priori knowledge of the number and distribution of false values; 2) we can prove that POPACCU is monotonic.

The key idea of POPACCU is to compute the distribution of false values on a data item D from the observed data. Note however, this is hard when we do not know which value is the correct value; we thus compute the popularity of a value with respect to each other value being true. We denote by $Pop(v|v_t)$ the popularity of v among all false values conditioned on v_t being true. Then, the probability that source S provides the correct value (*i.e.*, $\Psi_D(S) = v_t$) remains $A(S)$, but the probability that S provides a particular incorrect value becomes $(1-A(S))Pop(\Psi_D(S)|v_t)$. Thus, we have

$$Pr(\Psi_D(\bar{S})|v \text{ true})$$
$$= \Pi_{S\in\bar{S}(v)} A(S) \cdot \Pi_{S\in\bar{S}\setminus\bar{S}(v)} (1-A(S)) Pop(\Psi_D(S)|v) \quad (4)$$
$$= \Pi_{S\in\bar{S}(v)} \frac{A(S)}{1-A(S)} \cdot \Pi_{S\in\bar{S}\setminus\bar{S}(v)} Pop(\Psi_D(S)|v) \cdot \Pi_{S\in\bar{S}} (1-A(S)) \quad (5)$$

Here, $\Pi_{S\in\bar{S}}(1-A(S))$ is independent of v. We next simplify the computation of $\Pi_{S\in\bar{S}\setminus\bar{S}(v)} Pop(\Psi_D(S)|v)$.

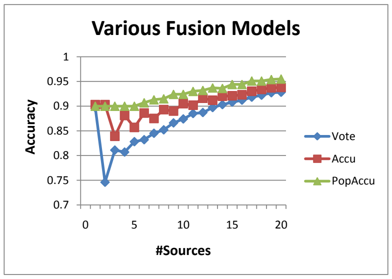


Figure 4: Model monotonicity.

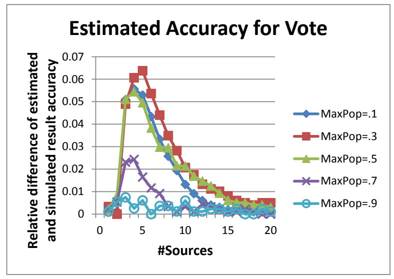


Figure 5: Estimation for VOTE.

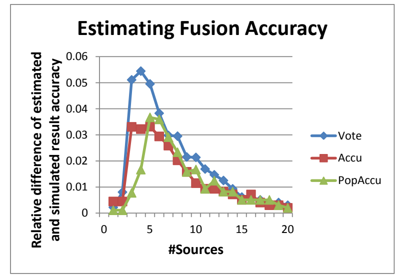


Figure 6: Accuracy estimation.

$$\Pi_{S\in\bar{S}\setminus\bar{S}(v)}Pop(\Psi_D(S)|v)=\Pi_{v_0\neq v}\left(\frac{|\bar{S}(v_0)|}{|\bar{S}|-|\bar{S}(v)|}\right)^{|\bar{S}(v_0)|}$$
$$=\frac{\Pi_{v_0}|\bar{S}(v_0)|^{|\bar{S}(v_0)|}}{|\bar{S}(v)|^{|\bar{S}(v)|}}\cdot\frac{1}{(|\bar{S}|-|\bar{S}(v)|)^{(|\bar{S}|-|\bar{S}(v)|)}}. \quad (6)$$

Since $\Pi_{v_0}|\bar{S}(v_0)|^{|\bar{S}(v_0)|}$ is independent of v, we compute the *popularity score* of a given value v as

$$\rho(v)=|\bar{S}(v)|\ln|\bar{S}(v)|+(|\bar{S}|-|\bar{S}(v)|)\ln(|\bar{S}|-|\bar{S}(v)|). \quad (7)$$

We compute the accuracy score of source S as $\alpha(S)=\ln\frac{A(S)}{1-A(S)}$ and the confidence of v as $C(v)=\sum_{S\in\bar{S}(v)}\alpha(S)-\rho(v)$. We again choose the value with the maximum confidence. Note that the accuracy score of a source can be much lower than in ACCU; thus, a value provided by low-accuracy sources can have much lower confidence. We next show several properties of POPACCU.

PROPOSITION 3.3. *When there are n false values that are uniformly distributed,* ACCU *and* POPACCU *output the same value.* □

THEOREM 3.4. *The* POPACCU *model is both order independent and monotonic.* □

PROOF. Let v_t be the true value and $v_1,\dots,v_l$ be false values. Consider the ratio $R_j=\frac{Pr(\Psi_D(\bar{S})|v_t\text{ true})}{Pr(\Psi_D(\bar{S})|v_j\text{ true})}$ for each $j\in[1,l]$. We next prove that R_j always increases for each j when we add a new source S. Source S has probability $A(S)$ to provide the correct value, and probability $(1-A(S))\cdot Pop(v_j|v_t)$ to provide the false value v_j. According to Eq.(4), the new ratio is

$$R'_j=R_j\cdot\frac{A(S)^{A(S)}}{A(S)^{(1-A(S))Pop(v_j|v_t)}\,((1-A(S))Pop(v_t|v_j))^{A(S)}}$$
$$\cdot\frac{\Pi_{i=1}^{l}\left((1-A(S))Pop(v_i|v_t)\right)^{(1-A(S))Pop(v_i|v_t)}}{\Pi_{i\neq j,i\in[1,l]}\left((1-A(S))Pop(v_i|v_j)\right)^{(1-A(S))Pop(v_i|v_t)}}$$
$$=R_j\cdot\frac{X_1X_2}{X_3};$$
$$X_1=\left(\frac{A(S)}{1-A(S)}\right)^{A(S)-(1-A(S))Pop(v_j|v_t)};$$
$$X_2=(\sharp-\sharp v_j)^{1-(1-A(S))Pop(v_j|v_t)}(\sharp v_j)^{(1-A(S))Pop(v_j|v_t)};$$
$$X_3=(\sharp-\sharp v_t)^{1-A(S)}(\sharp v_t)^{A(S)}.$$

Here, $\sharp$ denotes the number of occurrences of all values, and $\sharp v_j$ denotes the number of occurrences of v_j, $j\in[0,l]$, for D.

We can prove that X_3 obtains the maximum value when $\frac{\sharp v_t}{\sharp}=A(S)$; the maximum value is $\sharp\cdot(1-A(S))^{1-A(S)}\cdot A(S)^{A(S)}$. Similarly, X_2 obtains the minimum value when $\frac{\sharp v_t}{\sharp}=A(S)$ and further when $v_j=\frac{\sharp}{2}$; the minimum value is $\frac{\sharp}{2}$. Finally, $\frac{X_1X_2}{X_3}\geq\frac{1}{2(1-A(S))^{1-(1-A(S))Pop(v_j|v_t)}\cdot A(S)^{(1-A(S))Pop(v_j|v_t)}}$ obtains the minimum value when $A(S)=(1-A(S))Pop(v_j|v_t)$ and the minimum value is 1. Because $A(S)>(1-A(S))Pop(v_j|v_t)$ for "good" sources, $R'_j>R_j$. □

Algorithm 1: PopScore($\Psi_D(\bar{S})$)

Input : $\Psi_D(\bar{S})$: observation of values provided by $\bar{S}$ on D
Output : v: the value that is most likely to be true for D

```
// Count providers and accumulate vote counts for each value
n[v_1 : v_l] ← 0; c[v_1 : v_l] ← 0;
foreach S ∈ S̄ do
    n[Ψ_D(S)] ← n[Ψ_D(S)] + 1;
    c[Ψ_D(S)] ← c[Ψ_D(S)] + α(S);
// Compute the popularity scores
foreach i ∈ [1, l] do
    c[v_i] ← c[v_i] − n[v_i] ln n[v_i] − (|S̄| − n[v_i]) ln (|S̄| − n[v_i]);
return arg max_{i∈[1,n]} c[v_i];
```

We show in Algorithm POPACCU how we apply the POPACCU model. Obviously the algorithm takes time $O(|\bar{S}|)$.

The next example shows the advantage of POPACCU over ACCU and VOTE.

EXAMPLE 3.5. *Consider the following distribution of false values for each data item: the i-th most popular false value has popularity $(.2)^{i-1}-(.2)^i$ (so the maximum popularity is .8). Consider three sources: S_1 has accuracy .9 and provides value v_1, S_2 and S_3 have accuracy .6 and both provide value v_2. Obviously,* VOTE *would output v_2. Assuming there are 100 false values,* ACCU *computes accuracy scores for the sources as $\ln\frac{100*.9}{1-.9}=6.8,\ln\frac{100*.6}{1-.6}=5,5$, respectively; thus, v_1 has confidence 6.8 and v_2 has confidence 10, so it selects v_2.* POPACCU *computes source accuracy scores as $\ln\frac{.9}{1-.9}=2.2,\ln\frac{.6}{1-.6}=.4,.4$, respectively; the popularity scores of both values are $1\ln1+2\ln2=1.4$. Thus, v_1 has confidence $2.2-1.4=.8$ and v_2 has confidence $.8-1.4=-.6$, so* POPACCU *selects v_1.*

Note that according to our knowledge of source accuracy and distribution of false values, the probability that S_1 provides the correct value while S_2 and S_3 provide the same false value (so v_1 is true) is $.9*.4^2*(.8^2+.16^2+\dots)=.1$, and the probability that S_1 provides a false value while S_2 and S_3 provide the correct one (so v_2 is true) is $.1*.6^2=.036<.1$. Therefore, v_1 is more likely to be true and POPACCU *makes a wiser decision.*

Finally, we randomly generated synthetic data for 20 sources with accuracy $.9,.6,.6,\dots$ on 10000 data items. We started with the first source and gradually added the others; for each data set, we conducted fusion and computed the accuracy of the results (shown in Fig.4). We observed that (1) the ranking of the result accuracy is *always* POPACCU, ACCU *and* VOTE*; and (2)* POPACCU *is monotonic but* ACCU *and* VOTE *are not.* □

4. QUALITY ESTIMATION

129308A fundamental problem in source selection is gain estimation; in the fusion context this relies on estimating accuracy of fusion results. In this section we show that for most advanced fusion models, we are able to estimate fusion accuracy based purely on profiles of source quality. This is because the accuracy of sources can be considered as probability of a source providing a correct value, and so does accuracy of fusion; thus, we can apply probability analysis for accuracy estimation. Intuitively, we can enumerate all possible worlds of the provided values and sum up the probabilities of those where the model outputs the true value. Source accuracy and false-value distribution will be required in computing the probability of each possible world. Formally, we denote by $\mathbf{W}(\bar{S})$ the set of possible worlds for values provided by $\bar{S}$ on a data item and estimate the fusion accuracy of model F by

$$\hat{A}(F(S)) = \sum_{W \in \mathbf{W}(\bar{S})} Pr(W|F \text{ outputs the true value in } W). \quad (8)$$

Estimating fusion accuracy is hard because the accuracy improvement from an additional source depends not only on the accuracy of the fusion results over already probed sources, but also on the accuracy of each individual probed source. We next illustrate the hardness by an example.

EXAMPLE 4.1. *Suppose $\bar{S}_1$ contains one source with accuracy .9, $\bar{S}_2$ contains 41 sources with accuracy .6, and $\bar{S}_0$ contains 5 sources with accuracy .6. Assume there is a single false value. Fusing $\bar{S}_1$ and fusing $\bar{S}_2$ by* POPACCU *reach the same accuracy .9; however, adding $\bar{S}_0$ to $\bar{S}_2$ increases the accuracy to .915, while adding it to $\bar{S}_1$ does not increase the accuracy at all since even the total vote counts of $\bar{S}_0$ is still far lower than that of $\bar{S}_1$.* □

4.1 Hardness results

The hardness of accuracy estimation remains an open problem and our conjecture is that it is #P-hard[7] even for VOTE. Indeed, we can prove that a similar problem is #P-hard.

We now consider a subset of possible worlds, $\bar{W} \subseteq \mathbf{W}(\bar{\mathbf{S}})$ such that for each $W \in \bar{W}$, v_1 is provided more often and each of the k possible values receives at least one vote. Let $R1$ denote the predicate that "*v_1 is provided more often and each specific value receives at least one vote*" and let $R0$ denote the predicate that "*v_1 is provided more often and there exists at least one value which does not receive any vote*". Let

$$\hat{B}(\text{VOTE}(\bar{S})) = \sum_{W \in \mathbf{W} \text{ where } R1 \text{ holds}} Pr(W) = \sum_{W \in \bar{W}} Pr(W),$$

and,

$$\hat{C}(\text{VOTE}(\bar{S})) = \sum_{W \in \mathbf{W} \text{ where } R0 \text{ holds}} Pr(W)$$

Clearly,

$$\hat{A}(\text{VOTE}(\bar{S})) = \hat{B}(\text{VOTE}(\bar{S})) + \hat{C}(\text{VOTE}(\bar{S})).$$

We next show that computing $\hat{B}(\text{VOTE}(\bar{S}))$ is $\#P$-complete.

THEOREM 4.2. *Computing $\hat{B}(\text{VOTE}(\bar{S}))$ is $\#P$-complete.* □

PROOF. Clearly, the problem is in $\#P$, since given the votes of each of the m sources, we can verify in polynomial time whether the corresponding voting gives the correct answer and each value gets at least one vote. We now show it is in fact $\#P$-hard.

Consider all possible integer composition $\mathcal{C}$ of m into k parts such that the first part has strictly higher value than any other part and all the k parts have value at least 1. For example, if $m = 6$ and $k = 3$, then following are the only valid compositions satisfying the required properties: $(4, 1, 1), (3, 2, 1), (3, 1, 2)$. For each possible world $W \in \bar{W}$, if we compute the number of votes received by v_1, $v_2, \cdots, v_k$, then the corresponding count vector must belong to $\mathcal{C}$. Similarly, for each $C \in \mathcal{C}$, there exists a bunch of possible world in $\bar{W}$ with count vector as C. Let us denote by $\bar{W}_C$ all the possible worlds in $\bar{W}$ with count vector as C.

Clearly, then

$$\hat{B}(\text{VOTE}(\bar{S})) = \sum_{C \in \mathcal{C}} \sum_{W \in \bar{W}(C)} Pr(\bar{W}_C).$$

If we have $m = n + 1$ and $k = n$, then $\mathcal{C} = \{(2, \overbrace{1, 1, ..., 1}^{n})\}$.

We now reduce the following $\#P$-complete problem to computing $\hat{B}(\text{VOTE}(\bar{S}))$ with $m = n + 1$ and $k = n$:

For any non-negative real $n \times n$ matrix A with $A(i, 1) > A(i, j), \forall i \in \{1, 2, \cdot, n\}$ and $\forall j \in \{1, 2, ..., n\}$, computing permanent of A, $perm(A)$, is $\#P$-complete (Lemma 4.3).

$$perm(A) = \sum_{\sigma \in S_n} \prod_{1}^{n} A(i, \sigma(i)),$$

where the sum is over all elements of the symmetric group S_n, i.e., over all possible permutations of $1, 2, ..., n$.

Consider a given A with ith row summing to "r_i" . Consider $\bar{A}$ such that $\bar{A}(i, j) = \frac{1}{r_i} A(i, j) \forall i$ and j in $1, 2, ..., n$. Then,

$$perm(A) = \left(\prod_{i=1}^{n} r_i \right) perm(\bar{A}).$$

We thus focus on computing $perm(\bar{A})$ and reduce the computation of $perm(\bar{A})$ to computing $\hat{B}(\text{VOTE}(\bar{S}))$.

We create the following instance for VOTE. Let there be $n + 1$ sources voting for n different values. The ith source $i \in 1, 2, ..., n$ vote for the jth value with probability $\bar{A}(i, j)$. The $(n+1)$th source only votes for v_1 with probability 1. Clearly, this is a valid instance for VOTE. Since, $A(i, 1) > A(i, j), \forall i \in \{1, 2, \cdot, n\}$ and $\forall j \in \{1, 2, ..., n\}$, and therefore $\bar{A}(i, 1) > \bar{A}(i, j), \forall i \in \{1, 2, \cdot, n\}$ and $\forall j \in \{1, 2, ..., n\}$, all the sources are good.

We know for the above instance of VOTE, $\mathcal{C} = \{(2, \overbrace{1, 1, ..., 1}^{n})\}$. Now, consider any possible world $W \in \bar{W}_C$ such that sources i and j, $i \neq n + 1$, $j \neq n + 1$, vote for v_1. Then $Pr(W) = 0$, since in W, the $(n+1)$th source must vote for a value other than v_1 and the probability that the source $(n + 1)$ votes for any value other than v_1 is zero. Therefore the possible worlds in $\bar{W}_C$ that contribute non-zero value to $\hat{B}(\text{VOTE}(\bar{S}))$ must have $(n + 1)$th source voting for v_1. Since, these set of possible worlds also have count vector

[7]#P-hardness is a complexity class for hard counting problems; it is believed that #P-hard problems cannot be solved in polynomial time unless $P = NP$.

$\{(2, \overbrace{1, 1, ..., 1}^{n})\}$, as a result, the voting function that denotes the index of the value the first n sources vote for is a permutation.

Hence,

$$\hat{B}(\textsc{Vote}(\bar{S})) = Pr(\bar{W}) = Pr(W((2, \overbrace{1, 1, ..., 1}^{n})))$$

$$= Pr((n{+}1)\text{th source votes for } v_1) \sum_{\sigma \in S_n} \prod_{i=1}^{n} Pr(\text{source } i \text{ votes for } \sigma(i))$$

$$= \sum_{\sigma \in S_n} \prod_{i=1}^{n} Pr(\text{source } i \text{ votes for } \sigma(i)) = \sum_{\sigma \in S_n} \prod_{1}^{n} \bar{A}(i, \sigma(i)) = perm(\bar{A}).$$

This completes the reduction and establishes the lemma. $\square$

In Lemma 4.3, we prove that computing permanent of any square non-negative real matrix with the first coordinate of each row having a value higher than the rest of the components is $\#P$-complete.

LEMMA 4.3. *For any non-negative real $n \times n$ matrix A with $A(i, 1) > A(i, j), \forall i \in \{1, 2, \cdot, n\}$ and $\forall j \in \{1, 2, ..., n\}$, computing permanent of A, $perm(A)$, is $\#P$-complete.*

$$perm(A) = \sum_{\sigma \in S_n} \prod_{1}^{n} A(i, \sigma(i)),$$

where the sum is over all elements of the symmetric group S_n, i.e., over all possible permutations of $1, 2, ..., n$.

PROOF. It is well-known due to a result of Valiant that computing permanent of a $0-1$ square matrix is $\#P$-complete [19]. We reduce the computation of permanent of any $0-1$ square matrix to the permanent computation of the stated class of matrix in this lemma. Given any $n \times n$ $0-1$ matrix B, we create a $n+1 \times n+1$ matrix A by appending a column of length $n+1$ with all entries equaling 2 and the (n+1)th row which has 2 in the first coordinate and zero elsewhere to B. That is, the first column of A has all entries equal to 2, $A[2, 3, ..., n][2, 3, ..., n+1] = B$ and $A[n+1][1, 2, ..., n+1] = (2, 0, 0, ..., 0)$. It is easy to see that $perm(A) = 2perm(B)$. Therefore, if we can compute $perm(A)$, then $perm(B)$ can be computed as well. $\square$

We next describe a dynamic-programming algorithm that approximates fusion accuracy in PTIME for VOTE and in pseudo-PTIME for other fusion models. Our approximation relies only on source accuracy and the popularity of the *most popular* false value.

4.2 Accuracy estimation for VOTE

We start with accuracy estimation for VOTE. Consider a set of m sources $\bar{S} = \{S_1, \ldots, S_m\}$ that provide data item D. Suppose v_t is the correct value for D. VOTE outputs the true value when v_t is provided more often than any specific false value[8]; thus, what really matters in accuracy estimation is the difference between vote counts for v_t and for each other value.

In case the most popular false value, denoted by v_1, has much higher popularity than any other false value, the chance that v_1 is provided less often than another false value is small unless v_1 is not provided at all. On the other hand, the likelihood that v_1 is not provided but another false value is provided more than once is very small too. Thus, we focus on the difference between the vote counts of v_t and v_1, denoted by d, and consider three cases: (1) no false value is provided; (2) some false value but not v_1 is provided; and (3) v_1 is provided. According to our analysis, VOTE outputs v_t in case (1), and also outputs v_t with a high likelihood in case (2) when v_t is provided more than once, and in case (3) when $d > 0$.

[8]We can easily extend our model for handling ties.

Algorithm 2: VOTEACCURACY($A(S_1), \ldots, A(S_m), p$)

```
Input  : A(S_i): accuracy of source S_i, i ∈ [1, m];
         p: popularity of the most popular false value
Output : Â(Vote(S̄)): estimated accuracy of applying VOTE on S̄
Pr1[−m : m] = Pr2[−m : m] = Pr3[−m : m] = 0;
Pr1[0] ← 1;
// Compute Pr1(k, d), Pr2(k, d), Pr3(k, d)
foreach k ∈ [1, m] do
    foreach d ∈ [−i, i] do
        Compute Pr1[d], Pr2[d], Pr3[d] according to Eq.(9-11);
// Compute fusion accuracy
sum ← 0;
foreach d ∈ [1, m] do
    sum ← sum + Pr1[d] + Pr3[d];
    if d > 1 then
        sum ← sum + Pr2[d];
return sum;
```

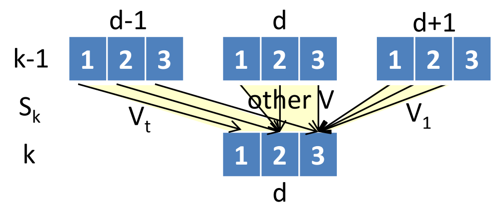


Figure 7: Transformation between cases in accuracy estimation for VOTE. Each rectangle represents the three cases for a particular set of sources and a particular d.

We define $Pr_1(k, d)$ as the probability that values provided by $S_1, \ldots, S_k, k \in [1, m]$, fall in case (1) with difference d (similar for $Pr_2(k, d)$ and $Pr_3(k, d)$). Initially, $Pr_1(0, 0) = 1$ and all other probabilities are 0. There are three possibilities for the value from S_k: if S_k provides v_t, the difference d increases by 1; if S_k provides v_1, d decreases by 1; otherwise, d stays the same. The transformation between different cases is shown in Fig.7. For example, if the first $k-1$ sources fall in case (2) with difference $d+1$ and S_k provides v_1, it transforms to case (3) with difference d. Let p be the popularity of v_1 (*i.e.*, $p = Pop(v_1|v_t)$); we can then compute the probability of each transformation and accordingly the probability of each case.

$$Pr_1(k, d) = A(S_k)Pr_1(k-1, d-1); \quad (9)$$

$$Pr_2(k, d) = A(S_k)Pr_2(k-1, d-1) + (1-p)(1-A(S_k))(Pr_1(k-1, d) + Pr_2(k-1, d)); \quad (10)$$

$$Pr_3(k, d) = A(S_k)Pr_3(k-1, d-1) + (1-p)(1-A(S_k))Pr_3(k-1, d) + p(1-A(S_k))\sum_{i=1}^{3} Pr_i(k-1, d+1); \quad (11)$$

$$\hat{A}(\text{Vote}(\bar{S})) = \sum_{d=1}^{m}(Pr_1(m, d) + Pr_3(m, d)) + \sum_{d=2}^{m} Pr_2(m, d) \quad (12)$$

Algorithm 2 (VOTEACCURACY) estimates the accuracy of VOTE according to Eq.(9-12). It has a low cost, but the approximation bound can be loose in the extreme case when the false values are uniformly distributed and each source has only a slightly higher probability to provide the true value than any particular false value (*i.e.*, $A(S) = \frac{p+\epsilon}{p+1}$, where ϵ is an arbitrarily small number).

Table 2: Results of $\langle Pr_1, Pr_2, Pr_3 \rangle$ in Ex.4.5. The probabilities for the cases where v_t is the output are in italic font.

d		S_1	S_2	S_3
-3				$< 0, 0, .002 >$
-2			$< 0, 0, .01 >$	$< 0, 0, .006 >$
-1		$< 0, 0, .05 >$	$< 0, 0, .02 >$	$< 0, 0, .054 >$
0	$< 1, 0, 0 >$	$< 0, .05, 0 >$	$< 0, .01, .21 >$	$< 0, .002, .096 >$
1		$< .9, 0, 0 >$	$< 0, .21, 0 >$	$< 0, .048, \mathit{.234} >$
2			$< .54, 0, 0 >$	$< 0, \mathit{.234}, 0 >$
3				$< \mathit{.324}, 0, 0 >$

THEOREM 4.4. VOTEACCURACY *takes time $O(|\bar{S}|^2)$. Let $\hat{A}$ be the precisely estimated accuracy and $\hat{A}_0$ be the accuracy computed by* VOTEACCURACY. *Then, $0 \le \hat{A}_0 - \hat{A} \le \frac{1-p}{1+p}$.* □

PROOF. Given the assumptions, $\hat{A}_0 > \hat{A}$. The upper bound of $\hat{A}_0 - \hat{A}$ is obtained when $A(S_i) = (1 - A(S_i))p + \epsilon, i \in [1, m]$, where ϵ is a very small number, and $p = \cdots = pop(v_{n'})$, where $n' < n$ and $A(S_i) + (1 - A(S_i))pn' \le 1$. In this case, $A(S_i) = \frac{p}{1+p}$ and $n' \le \frac{1}{p}$. Then, the distribution of values $v_0, \ldots, v_{n'}$ are almost not distinguishable. Thus, the probability of outputting each as a correct value is at most $\frac{1}{1+n'}$ and so $\lim_{\epsilon \to 0} \hat{A} \le \frac{1}{n+1}$. However, our computation assumes we will rarely output a value other than v_0 and v_1 unless v_0 is provided by only 1 source and v_1 is not provided, so $\hat{A}_0 = \frac{n'}{n'+1} - \frac{1}{n'+1} \cdot (\frac{n'-1}{n'+1})^{m-1} \le \frac{n'}{n'+1}$. Thus, $\hat{A}_0 - \hat{A} < \frac{n'-1}{n'+1} < \frac{1-p}{1+p}$. □

Empirically the difference between the estimated accuracy and the true one is typically small, as we show in the next example.

EXAMPLE 4.5. *Consider three sources where $A(S_1) = .9$, $A(S_2) = A(S_3) = .6$. Assume $p = .5$. Table 2 shows computation for $Pr_{1,2,3}$ in accuracy estimation. Take $Pr_3(3, 1)$ (the cell of column S_3 and row $d = 1$) as an example. It has contributions from $Pr_3(2, 0)$ when S_3 provides v_t (with probability .6), from $Pr_3(2, 1)$ when S_3 provides a false value other than v_1 (with probability $.4(1 - .5) = .2$), and from $Pr_1(2, 2), Pr_2(2, 2), Pr_3(2, 2)$ when S_3 provides v_1 (with probability $.4 * .5 = .2$). Thus, $Pr_3(3, 1) = .6 * .21 + .2 * 0 + .2 * .54 = .234$. The accuracy of the result is $.234 + .234 + .324 = .792$.*

Assume there are actually 10 false values with probabilities $.5, .25, .125, \cdots$. The real probability should be .7916. Instead of considering the $11^3 = 1331$ possible worlds, our algorithm computes only $(3 + 5 + 7) \times 3 = 45$ probabilities for accuracy estimation.

Fig.5 shows the difference between the estimated accuracy and the simulated accuracy on 10000 data items, when $A(S_1) = .9$, $A(S_2) = A(S_3) = \cdots = .6$ and p varies from .1 to .9. In our simulation we set the popularity of the i-th false value as $(1 - p)^{i-1} - (1 - p)^i, i \ge 1$ (so the maximum popularity is p). We observe that the peak of the difference occurs when we have less than 10 sources. When we have more than 10 sources with reasonably high accuracy, even when p is small, the difference is very small. □

4.3 Accuracy estimation for ACCU

Accuracy estimation is more complex for advanced fusion models, including ACCU and those proposed in [8, 15, 22], because each source can contribute a different vote. In particular, given a source S_i with vote count $\alpha(S_i)$, we shall use $d \pm \alpha(S_i)$ instead of $d \pm 1$ in Eq.(9-11). As a consequence, the maximum of d equals the sum of vote counts from all sources; therefore, the algorithm becomes pseudo-PTIME. We next give details for ACCU and similar techniques can be applied to other methods in [8, 15, 22].

For ACCU, we revise Eq.(9-12) as follows.

1. For each source S_i, the vote count is $\alpha(S_i)$. So instead of using $d \pm 1$, we use $d \pm \alpha(S_i)$ in Eq.(9-11).
2. When v_1 is not provided yet, outputting the true value v_t requires the vote count of v_t to be larger than the maximum vote count of the other false values, denoted by $maxC(k, d)$. We revise Eq.(12) as follows.

$$\hat{A}(\mathrm{Accu}(\bar{S})) = \sum_{d>0} (Pr_1(m, d) + Pr_3(m, d)) + \sum_{d > maxC(m,d)} Pr_2(m, d). \quad (13)$$

We maintain $maxC(k, d)$ only when $Pr_2(k, d) > 0$. Recall we assume that when v_1 is not provided yet, chance is rare for other false values to be provided more than once. So we just take the maximum vote count of the providers of less popular false values. According to Fig.7, we have

$$maxC(k, d) = \max(maxC(k - 1, d - \alpha(S_i)), maxC(k - 1, d), \alpha(S_i) * \delta), \quad (14)$$

where $\delta = 1$ when $Pr_1(k - 1, d) + Pr_2(k - 1, d) > 0$ (*i.e.*, S_k also provides this less popular false value), and $\delta = 0$ otherwise.

In addition, we make the following changes to the algorithm for efficiency reasons.

1. Accuracy scores are real values. We denote by σ the minimum difference we wish to ignore in vote-count comparison. We compute the probabilities only for $d = 0, \pm\sigma, \pm 2\sigma, \ldots, \pm \lceil \frac{\sum_{i=1}^{k} \alpha(S_i)}{\sigma} \rceil$ in the k-th round.
2. The arrays Pr_1, Pr_2, Pr_3 can be very sparse. Instead of considering each d value, in the k-th round we start from the d-values where at least one of the probabilities is not 0, and for each of them update the probabilities for difference $d - \alpha(S_k)$, d and $d + \alpha(S_k)$.

The resulting algorithm, ACCUACCURACY, takes pseudo-polynomial time and has the same approximation bound as VOTEACCURACY.

PROPOSITION 4.6. *Let s be the sum of the accuracy scores of sources in $\bar{S}$.* ACCUACCURACY *takes time $O(\frac{s|\bar{S}|}{\sigma})$. Let $\hat{A}$ be the precisely estimated accuracy and $\hat{A}_0$ be the accuracy computed by* ACCURACY. *Then, $0 \le \hat{A}_0 - \hat{A} \le \frac{1-p}{1+p}$.* □

EXAMPLE 4.7. *Continue with Ex.4.5. We use $n = 10$ in accuracy-score computation, so the scores are 4.5, 2.7, 2.7 respectively. Table 3 shows computation in accuracy estimation for* ACCU. *The accuracy is $.018 + .078 + .036 + .222 + .018 + .216 + .324 = .912$. Note that we do not count $Pr_2(3, 2.7)$, because $maxC(3, 2.7) = 4.5 > d = 2.7$. The real probability is .887. Since* ACCUACCURACY *computes only non-zero probabilities, it computes only 29 probabilities.* □

4.4 Accuracy estimation for POPACCU

POPACCU differs from ACCU in that it considers different popularities of the false values and deducts the popularity score of a value from its vote count; the difference of the vote count of v_t and v_1 becomes $d - \rho(v_t) + \rho(v_1)$. Note that $\rho(v_t)$ and $\rho(v_1)$ can vary from a possible world to another, so we cannot use the a-priori popularity value and have to compute them for each possible world. For this purpose, we (1) record the number of providers for v_t and

Table 3: Results of $\langle Pr_1, Pr_2, Pr_3, maxC\rangle$ in Ex.4.7. The probabilities for the cases where v_t is the output are in italic.

d		S_1	S_2	S_3
-9.9				$< 0, 0, .002, - >$
-7.2			$< 0, 0, .01, - >$	$< 0, 0, .004, - >$
-5.4				$< 0, 0, .002, - >$
-4.5		$< 0, 0, .05, - >$	$< 0, 0, .01, - >$	$< 0, 0, .008, - >$
-2.7			$< 0, 0, .01, - >$	$< 0, 0, .004, - >$
-1.8			$< 0, 0, .03, - >$	$< 0, 0, .006, - >$
-.9				$< 0, 0, .036, - >$
0	$< 1, 0, 0, - >$	$< 0, .05, 0, 2.7 >$	$< 0, .01, 0, 4.5 >$	$< 0, .002, .012, 4.5 >$
.9				$< 0, 0, \mathit{.018}, - >$
1.8			$< 0, 0, .18, - >$	$< 0, 0, \mathit{.078}, - >$
2.7			$< 0, .03, 0, 4.5 >$	$< 0, .012, 0, 4.5 >$
4.5		$< .9, 0, 0, - >$	$< 0, .18, 0, 2.7 >$	$< 0, \mathit{.036}, \mathit{.222}, 2.7 >$
5.4				$< 0, \mathit{.018}, 0, 4.5 >$
7.2			$< .54, 0, 0, - >$	$< 0, \mathit{.216}, 0, 2.7 >$
9.9				$< \mathit{.324}, 0, 0, - >$

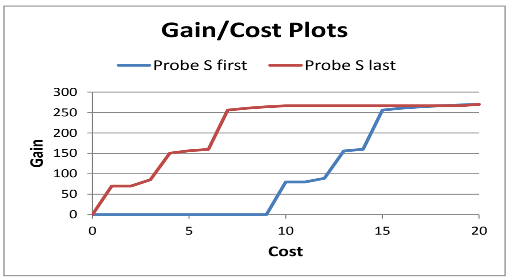


Figure 8: Gain/cost plots for Ex.5.1.

v_1, denoted by n_t and n_1, and (2) assume other false values are uniformly distributed. We compute $Pr(k, d, n_t, n_1)$ and the accuracy as follows. Initially, $Pr(0, 0, 0, 0) = 1$; then,

$$\begin{aligned} & Pr(k, d, n_t, n_1) = A(S_k)Pr(k-1, d-\alpha(S_k), n_t - 1, n_1) \\ & + p(1 - A(S_k))Pr(k-1, d+\alpha(S_k), n_t, n_1 - 1) \\ & + (1-p)(1 - A(S_k))Pr(k-1, d, n_t, n_1); \qquad (15) \\ & \hat{A}(\text{PopAccu}(\bar{S})) = \sum_{d-\rho(v_t)+\rho(v_1)>0, n_1>0} Pr(m, d, n_t, n_1) \\ & + \sum_{d-maxC(m,d,n_t,0)-\rho(v_t)+\rho(v_2)>0} Pr(m, d, n_t, 0), \qquad (16) \end{aligned}$$

where $\rho(v_2)$ denotes the popularity score for one of the less popular false values, and we compute $\rho(v_t), \rho(v_1), \rho(v_2)$ according to n_t and n_1. We can design the algorithm POPACCURACY similarly.

PROPOSITION 4.8. *Let s be the sum of the accuracy scores of sources in $\bar{S}$.* POPACCURACY *takes time $O(\frac{s|\bar{S}|^3}{\sigma})$. Let $\hat{A}$ be the precisely estimated accuracy and $\hat{A}_0$ be the accuracy computed by* POPACCURACY. *Then, $0 \leq \hat{A}_0 - \hat{A} \leq \frac{1-p}{1+p}$.* □

EXAMPLE 4.9. *Continue with Ex. 4.5. Fig.6 shows the difference between the estimated accuracy and the simulated accuracy on 10000 data items for various fusion models when $p = .5$. We observe that the peaks of the difference for* ACCU *and* POPACCU *also occur with less than 10 sources, and the differences for these two models are lower than that for* VOTE. □

5. SOURCE SELECTION

The MARGINALISM problem can be very challenging when the gain is associated with fusion accuracy, illustrated as follows.

EXAMPLE 5.1. *Consider 11 sources, where the first one S has accuracy .8 and cost 10, while the rest of the sources each has accuracy .7 and cost 1. Consider the gain function where $G(A) = 100A$ when $A < .9$, $G(A) = 150 + 200(A - .9)$ when $.9 \leq A < .95$, and $G(A) = 250 + 500(A - .95)$ when $A \geq .95$. Consider* POPACCU *and assume the most popular false value has popularity .5. Fig.8 plots gain versus cost for two orderings of the sources.*

Consider a naive strategy that greedily selects the next source that leads to the highest profit (gain−cost). According to the Law of Diminishing Returns, *we would stop when the marginal gain from the next source is less than the marginal cost. However, in our context the marginal gain does not necessarily decrease monotonically; in Fig.8 for both orderings, the second source has a lower marginal gain than some later ones (this can be true even for a continuous gain model, as shown in Fig.4). If we follow this strategy, in our example we would select only S with profit $80-10 = 70$, but selecting all sources would obtain a much higher profit $270 - 20 = 250$.*

Even if we keep trying till we exhaust all sources and select the subset with the highest profit, this greedy strategy can still fall short because the best marginal points for different sequences of sources can be different, and the one for the greedily generated sequence may not be optimal globally. In our example, the greedy algorithm would probe S first as $80 - 10 > 70 - 1$; accordingly, the selection is at best to select all sources. However, excluding S from the selection would obtain a higher profit $266.5 - 10 = 256.5 > 250$. In fact, as we show shortly, this greedy scheme can result in an arbitrarily bad solution. □

This section considers two cost models: the *constant cost model* assumes that all sources have the same cost and so the overall cost is decided by the number of sources; the *arbitrary cost model* assumes that each source has an arbitrary cost and so the overall cost is the sum of the costs. When the sources are free and we focus on data-accessing time decided mainly by the number of input sources, we can apply the former; when we need to purchase data and different sources offer different prices, we can apply the latter. Sec.5.1 shows that the various source-selection problems are in PTIME under the constant cost model but intractable under the arbitrary cost model. Sec.5.2 describes a randomized algorithm for the MARGINALISM problem.

5.1 Complexity results

Constant cost model: Assume each source has cost c; thus, the sources are indistinguishable in terms of cost. Our results are based on the following lemma.

LEMMA 5.2. *Let S be a set of full-coverage sources and $\bar{S}_0 \subseteq S$ be the $|\bar{S}_0|$ sources with the highest accuracies. Then, for any* *subset* $\bar{S} \subseteq \mathcal{S}$ *with size* $|\bar{S}_0|$ *and any fusion model* F *among* VOTE, ACCU *and* POPACCU, $\hat{A}(F(\bar{S}_0)) \geq \hat{A}(F(\bar{S}))$. □

Consider the MARGINALISM problem with a budget τ_c. We can select at most $M = \lfloor \frac{\tau_c}{c} \rfloor$ sources. We proceed in three steps: 1) sort the sources in decreasing order of their accuracy; 2) from the first M sources, iteratively add each source and compute the profit. and 3) choose the prefix subset (starting from the first source to a particular source) with the highest profit. We solve the other two problems in a similar way.

Applying a monotonic fusion model can simplify source selection. First, MAXGLIMITC can simply choose the first M sources. Second, in the special case where all sources have cost 0 so essentially the goal is to maximize fusion accuracy, we can simply choose all sources (recall that we consider only "good" sources).

THEOREM 5.3. *Under the constant cost model, the problems* MAXGLIMITC, MINCLIMITG, *and* MARGINALISM *are in PTIME for the* VOTE, ACCU, *and* POPACCU *fusion models if we use a polynomial-time oracle for fusion-accuracy estimation.* □

Arbitrary cost model: Under the arbitrary cost model, the MAXGLIMITC problem is in PTIME if we do not have a budget (*i.e.*, the budget is higher than the sum of the costs of all sources), but is NP-complete in general. We have symmetric results for MINCLIMITG. The NP-hardness of the former can be proved by a reduction from the NP-hard *0-1 Knapsack* problem and that of the latter can be proved by a reduction from the NP-hard *Partition* problem. The MARGINALISM problem can be reduced from MINCLIMITG, so it is already NP complete even if $\tau_c \geq C(\mathcal{S})$.

THEOREM 5.4. *Assume arbitrary cost model and access to a polynomial-time oracle for fusion-accuracy estimation for* VOTE, ACCU *and* POPACCU.

- *The* MAXGLIMITC *problem is in PTIME when* $\tau_c \geq C(\mathcal{S})$ *and NP-complete in general.*
- *The* MINCLIMITG *problem and the* MARGINALISM *problem are NP-complete.* □

5.2 Solving MARGINALISM

As illustrated in Example 5.1, a greedy strategy is insufficient for solving the MARGINALISM problem. Indeed, the next theorem shows that it can get arbitrarily bad results.

THEOREM 5.5. *Let* d_{opt} *be the optimal profit for a given* MARGINALISM *problem and* d *be that from the set of sources selected greedily by maximizing the profit in each step. For any* $\theta > 0$, *there exists an input set of sources and a gain model such that* $\frac{d}{d_{opt}} < \theta$. □

We next present an algorithm that applies the *Greedy Randomized Adaptive Search Procedure (GRASP)* meta-heuristic [7] to solve the MARGINALISM problems; the same idea applies to the other two problems too. GRASP solves the problems of the greedy approach in two ways. First, instead of making the greedy decision every time, in each step it randomly chooses from the top-k candidates in terms of resulting profit, and chooses the best selection from r repetitions. Second, in each repetition, after generating the initial solution, it performs local search in a hill-climbing fashion. Both components are critical in avoiding exploring the sources in a fixed order and so make it possible to reach the optimal selection.

Algorithm 3 shows the framework of the GRASP approach. It performs r iterations (Ln.2). In each iteration, the *construction phase* builds a feasible solution $\bar{S}$ (Ln.3), and then the *local-search phase* investigates the neighborhood of $\bar{S}$ in a hill-climbing fashion until reaching the local optimal solution (Ln.4). It then returns the best solution from all iterations (Ln.5-7).

Algorithm 3: GRASP($\mathcal{S}, F, r, k$)

Input : $\mathcal{S}$: sources for selection; F: fusion model; r: number of repetitions; k: finding top-k candidates
Output : $\bar{S}_{opt}$: selected sources

1 $\bar{S}_{opt} \leftarrow \emptyset$; $f_{opt} \leftarrow 0$; // f_{opt} *records the highest profit*
2 **foreach** $i \in [1, r]$ **do**
3    $< \bar{S}, g, c > \leftarrow$ CONSTRUCTION($\mathcal{S}, F, \emptyset, 0, 0, k$);
4    $< \bar{S}, g, c > \leftarrow$ LOCALSEARCH($\mathcal{S}, F, \bar{S}, g, c, k$);
5   **if** $g - c > f_{opt}$ **then**
6      $\bar{S}_{opt} \leftarrow \bar{S}$; $f_{opt} \leftarrow g - c$;
7 **return** $\bar{S}_{opt}$;

Algorithm 4: CONSTRUCTION($\mathcal{S}, F, \bar{S}, g, c, k$)

Input : $\mathcal{S}$: sources for selection; F: fusion model; $\bar{S}$: already selected sources; g: gain for $\bar{S}$; c: cost for $\bar{S}$; k: finding top-k candidates
Output : $< \bar{S}_{opt}, g, c >$: the newly selected sources and their gain and cost

1 $\bar{S}_{opt} \leftarrow \bar{S}$; $f_{opt} \leftarrow g - c$; // *Initialize the best solution as the input*
2 **foreach** $i \in [1, |\mathcal{S}| - |\bar{S}|]$ **do**
   // *Find top-k candidates*
3    $\overline{BEST} \leftarrow \emptyset$; $\bar{F} \leftarrow \emptyset$; // *Store the top-$k$ candidates*
4   **foreach** $S \in \mathcal{S} \setminus \bar{S}$ **do**
5   **if** $G(1) - c - C(S) > f_{opt}$ **then**
6        $f \leftarrow G(\hat{A}(F(\bar{S} \cup \{S\}))) - g - C(S)$;
7        $k' \leftarrow$ rank of f in $\bar{F}$;
8   **if** $k' \leq k$ **then**
9          $\overline{BEST} \leftarrow \overline{BEST} \cup \{S\}$; $\bar{F} \leftarrow \bar{F} \cup \{f\}$;
10   **if** $|\bar{F}| > k$ **then**
11   Remove the smallest value from $\bar{F}$;
12   Update $\overline{BEST}$ accordingly;
   // *Randomly select the next source from the top-k candidates*
13   **if** $\bar{F} = \emptyset$ **then**
14   **break**;
15   Randomly choose f_0 from $\bar{F}$;
16   Update $\bar{S}, g, c$ accordingly;
17   **if** $f_0 > f_{opt}$ **then**
18      $f_{opt} \leftarrow f_0$; $\bar{S}_{opt} \leftarrow \bar{S}$;
19 **return** $< \bar{S}_{opt}, f_{opt} + C(\bar{S}_{opt}), C(\bar{S}_{opt}) >$;

The construction phase (Algorithm 4) starts with the given subset of sources (empty initially) and iteratively adds a set of sources in a greedy randomized fashion. In each iteration (Ln.2-18), Ln.5 first checks for each remaining source whether beating the current best solution is possible by reaching the maximum possible gain (*i.e.*, $G(1)$), and skips the source if not. Ln.6 estimates the difference between the marginal gain and marginal cost of adding the source. Then, Ln.7-12 maintains the top-k candidates; Ln.13-16 randomly selects one of them to add next. Finally, Ln.17-19 chooses the prefix subset with the highest profit.

The local-search phase (Algorithm 5) takes the initial solution as input and iteratively explores its neighborhood for a better solution. In each iteration (Ln.2-10), it examines each of the already selected sources S (Ln.4), and compares the current solution with (1) the solution of removing S (Ln.5-6), and (2) the solution of replacing S with a subset of the remaining sources, selected by invoking CONSTRUCTION (Ln.7). It terminates when examining any

Algorithm 5: LOCALSEARCH($\mathcal{S}, F, \bar{S}, g, c, k$)

Input : $\mathcal{S}$: sources for selection; F: fusion model; $\bar{S}$: already selected sources; g: gain for $\bar{S}$; c: cost for $\bar{S}$; k: finding top-k candidates

Output : $< \bar{S}_{opt}, g, c >$: the newly selected sources and their gain and cost

1 $changed \leftarrow true$;
2 **while** $changed$ **do**
3     $changed \leftarrow false$;
4   **foreach** $S \in \bar{S}$ **do**
5         $\bar{S}_0 \leftarrow \bar{S} \setminus \{S\}$; $c_0 \leftarrow c - C(S)$;
6         $g_0 \leftarrow G(\hat{A}(F(\bar{S}_0)))$;*// Invoke estimation methods*
7         $< \bar{S}_0, g_0, c_0 > \leftarrow$CONSTRUCTION($\mathcal{S}, F, \bar{S}_0, g_0, c_0, k$);
8   **if** $g_0 - c_0 > g - c$ **then**
9             $\bar{S} \leftarrow \bar{S}_0$; $g = g_0$; $c = c_0$;
10             $changed \leftarrow true$; **break**;
11 **return** $< \bar{S}, g, c >$;

selected source cannot improve the solution (Ln.2, Ln.8-10). Since the profit cannot grow infinitely, the local search will converge.

Note that when $k = 1$, all iterations of GRASP will generate the same result and the algorithm regresses to a hill-climbing algorithm. When $k = |\mathcal{S}|$, the construction phase can generate any ordering of the sources and a high r leads to an algorithm that essentially enumerates all possible source orderings. Our experiments show that with a continuous gain model, setting $k = 5$ and $r = 20$ can obtain the optimal solution most of the time for more than 200 sources, but with a non-continuous gain model, we need to set much higher k and r.

EXAMPLE 5.6. *Consider 8 sources: the first, S, has accuracy .8 and cost 5; and each of the rest has accuracy .7 and cost 1. Consider* POPACCU *and gain function* $G(A) = 100A$. *Assume* $k = 1$*, so the algorithm regresses to a hill-climbing algorithm.*

The construction phase first selects S *as its profit is higher than the others (*$80 - 5 > 70 - 1$*). It then selects 5 other sources, reaching a profit of* $96.2 - 10 = 86.2$*. The local-search phase examines* S *and finds that (1) removing* S *obtains a profit of* $93.2 - 5 = 88.2$*; and (2) replacing* S *with the 2 remaining sources obtains a profit of* $96.2 - 7 = 89.2$*. Thus, it selects the 7 less accurate sources. It cannot further improve this solution and terminates.* □

6. EXTENSION FOR PARTIAL COVERAGE

We next extend our results for sources without full coverage. We define the coverage of source S as the percentage of its provided data items over $\mathcal{D}$, denoted by $V(S)$. First, considering coverage would affect accuracy estimation. We need to revise Eq.(9-11) by considering the possibility that the k-th source does not provide the data item at all.

$$Pr_1(k, d) = V(S_k)A(S_k)Pr_1(k-1, d-1); \tag{17}$$

$$\begin{aligned} Pr_2(k, d) &= V(S_k)A(S_k)Pr_2(k-1, d-1) \\ &+(1 - V(S_k) + V(S_k)(1-p)(1 - A(S_k))) \\ &\cdot(Pr_1(k-1, d) + Pr_2(k-1, d)); \end{aligned} \tag{18}$$

$$\begin{aligned} Pr_3(k, d) &= V(S_k)A(S_k)Pr_3(k-1, d-1) \\ &+(1 - V(S_k) + V(S_k)(1-p)(1 - A(S_k)))Pr_3(k-1, d) \\ &+V(S_k)p(1 - A(S_k))\sum_{i=1}^{3} Pr_i(k-1, d+1); \end{aligned} \tag{19}$$

Note that the revised estimation already incorporates coverage of the results and is essentially the percentage of correctly provided values over all data items (*i.e.*, the product of coverage and accuracy); we call it *recall*.

Second, the gain model can be revised to a function of recall such that it takes both coverage and accuracy into account. Lemma 5.2 does not necessarily hold any more so whether the optimization problems are in PTIME under the constant cost model remains an open problem. However, the GRASP algorithm still applies and we report experimental results in Sec.7.

7. EXPERIMENTAL RESULTS

This section reports experimental results showing that (1) our algorithms can select a subset of sources that maximizes fusion accuracy; (2) when we consider cost, we are able to efficiently find a subset of sources that together obtains nearly the highest profit; (3) POPACCU outperforms the other fusion models and we estimate fusion accuracy quite accurately; (4) our algorithms are scalable.

7.1 Experiments on real data

7.1.1 Experiment setup

Data: We experimented on two data sets. The *Book* data set contains 894 data sources that were registered at *AbeBooks.com* and provided information on computer science books in 2007 (see Ex.1.1-1.2). In total they provided 24364 listings for 1265 books on ISBN, name, and authors; each source provides .1% to 86% of the books. By default, we set the coverage and accuracy of the sources according to a gold standard containing the author lists from the book cover on 100 randomly selected books. In accuracy estimation we set the maximum popularity p as the largest popularity of false values among all data items.

The *Flight* data set contains 38 Deep Web sources among top-200 results by *Google* for keyword search "flight status". We collected data on 1200 flights for their flight number (serving as identifier), scheduled/actual departure/arrival time, and departure/arrival gate on 12/8/2011 (see [10] for details of data collection). In total they provided 27469 records; each sources provides 1.6% to 100% of the flights. We used a gold standard containing data provided by the airline websites *AA, UA*, and *Continental* on 100 randomly selected flights. We sampled source quality both for overall data and for each attribute.

Fig.9 shows distribution of recall of the sources in the two data sets. For both data sets we observe a few high-recall data sources (3 *Book* sources and 8 *Flight* sources with a recall above .5), some medium-recall sources (11 *Book* sources and 3 *Flight* sources with a recall in $[.25, .5)$), and a large number of "tail" sources with low recall. We observed very similar results on these two data sets; [5] also describes experiments on synthetic data.

Implementations: We implemented three fusion models VOTE, ACCU, and POPACCU. We handled ties by randomly choosing a value with top votes.

We considered three optimization goals: MAXGLIMITC with $\tau_c = \frac{G(1)}{2}$ ($G(1)$ corresponds to the maximum gain), MINCLIMITG with $\tau_g = G(.8)$, and MARGINALISM with $\tau_c = \infty$. We implemented GRASP for each goal; by default we set $r = 20, k = 5$. For MARGINALISM, we in addition implemented the GREEDY algorithm, which essentially invokes CONSTRUCTION with $k = 1$.

We tried different cost and gain models to study their effect on source selection. We used three gain models: LINEARGAIN assumes that the gain grows linearly with recall of fusion results, denoted by R, and sets $G(R) = 100R$; QUADGAIN assumes that the gain grows quadratically with recall and sets $G(R) = 100R^2$; STEPGAIN assumes that reaching some "milestone" of recall would significantly increase gain and so sets

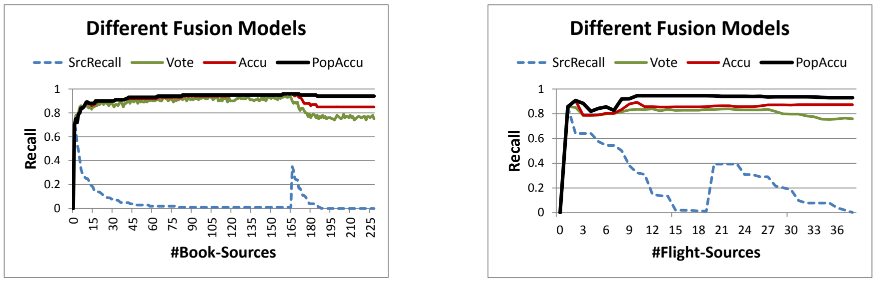


Figure 9: Fusion quality for different models.

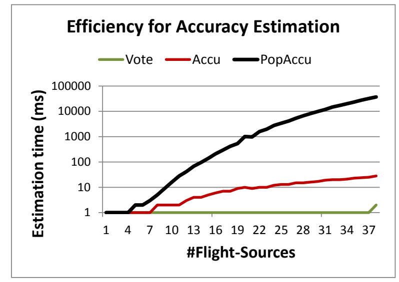


Figure 10: Estimation efficiency.

Table 4: Estimated recall vs. real fusion recall averaged on each data set.

Domain	Model	Avg real	Avg est.	Abs diff	Rel diff
Book	VOTE	.868	.939	.071	8.3%
	ACCU	.908	.971	.064	7.2%
	POPACCU	.933	.975	.043	4.7%
Flight	VOTE	.813	.877	.073	8.9%
	ACCU	.857	.956	.100	11.7%
	POPACCU	.924	.976	.052	5.7%

$$G(R) = \begin{cases} 100R & : 0 \le R < .8 \\ 100 + 100(R - .8) & : .8 \le R < .9 \\ 150 + 100(R - .9) & : .9 \le R < .95 \\ 200 + 100(R - .95) & : .95 \le R < .97 \\ 300 + 100(R - .97) & : .97 \le R \le 1 \end{cases}$$

We assigned the cost of a source in $[1, 10]$ in seven ways (we observed similar patterns for other ranges):

- CONSTCOST applies $C(S) = 1$;
- RANDOMCOST assigns a random integer cost in $[1, 10]$;
- LINEARCOVCOST assumes the cost grows linearly with the coverage of the source and applies $C(S) = 9V(S) + 1$, where $V(S)$ is the coverage of S;
- LINEARACCUCOST assumes the cost grows linearly with the accuracy of the source and applies $C(S) = 9A(S) + 1$;
- LINEARQUALCOST assumes the cost grows linearly with the recall, denoted by $R(S) = A(S)V(S)$, and applies $C(S) = 9R(S) + 1$;
- QUADQUALCOST assumes the cost grows quadratically with the recall and applies $C(S) = 9R(S)^2 + 1$;
- STEPQUALCOST assumes reaching some "milestone" of recall would significantly increase cost and so applies

$$C(S) = \begin{cases} 1 + 5R(S) & : 0 \le R(S) < .5 \\ 5 + 5(R(S) - .5) & : .5 \le R(S) < .7 \\ 7 + 5(R(S) - .7) & : .7 \le R(S) < .8 \\ 9 + 5(R(S) - .8) & : .8 \le R(S) \le 1 \end{cases}$$

We implemented in Java and experimented on a Linux server with 2.26 GHz Intel Xeon Processor X7560 and 24M Cache.

Measures: For fusion results, we compared the returned author lists with the gold standard and reported the recall. For accuracy estimation, we reported the absolute and relative difference between the estimated recall and the fusion results. For source selection we compared the selected sources by profit.

7.1.2 Maximizing fusion quality

We first considered maximizing fusion quality; this is equivalent to solving the MARGINALISM problem with zero-cost sources.

Among the 894 sources in the *Book* data set, 228 provide books in the gold standard; among them MARGINALISM selects 165 (72.4%) for POPACCU. Actually, since POPACCU is monotonic, MARGINALISM selects all "good" sources. Also, MARGINALISM selects the same sources for VOTE and ACCU. All 38 sources in the *Flight* data set provide flights in the gold standard; among them *Marginalism* selects 18 sources (50%) for POPACCU, and 15 sources (39.5%) for VOTE and ACCU.

We ordered the sources such that the selected sources are ordered before the unselected ones, and the selected (resp. unselected) sources are in decreasing order of their recall. Fig.9 shows the recall by each fusion model as we gradually added the sources in this order. We made three observations. (1) the recall of POPACCU indeed is the highest (.96 for *Book* and .95 for *Flight*) on the selected sources and gradually decreases after fusing unselected sources, showing effectiveness of the selection. (2) The recall of POPACCU increases most of the time when processing the selected sources. Even though the assumptions that the data items are indistinguishable and the sources are independent do not hold on neither data set, there are very few decreases for POPACCU at the beginning of the curve for each domain. (3) On average POPACCU improves over VOTE by 7.5% and over ACCU by 2.8% on *Book*, and by 13.7% and 7.8% respectively on *Flight*.

Table 4 compares the estimated recall with the real one. The difference is quite small and is the smallest for POPACCU. Fig.10 shows accuracy-estimation time on *Flight* (note that for each subset of sources we estimate accuracy for each attribute and then take the weighted average). POPACCU finished in 37 seconds on all sources, taking considerably longer time (3 orders of magnitude) than ACCU, which in turn took 1 order of magnitude longer time than VOTE.

7.1.3 Source selection

We next took cost into account for source selection and conducted five experiments.

I. Selection-goal comparison: Fig.11 compares different source-selection goals when we applied VOTE, LINEARGAIN, and various cost models on *Book* data (we observed the same pattern for other fusion and gain models). First, MARGINALISM has the highest profit most of the time; on average it beats MAXGLIMITC by 72% as the latter always incurs a big cost, and beats MINCLIMITG by 15% as the latter always stops with a fairly low gain (depending on the thresholds). This difference is more pronounced for the other gain models. Second, with more expensive sources, we tend to select fewer sources, so obtain a higher cost and a lower gain and thus a lower profit. In particular, under cost models CONSTCOST with $C(S) = 1$, MARGINALISM selects 7 sources and obtains an estimated gain of 90.5 (profit $90.5 - 7 = 83.5$); recall from Section 1 that with $C(S) = .1$, MARGINALISM selects 26 sources with profit $97 - 2.6 = 94.4$.

We have similar observation on *Flight* data: MARGINALISM beats MAXGLIMITC by 55% and beats MINCLIMITG by 4.9%.

II. Algorithm comparison: We applied GREEDY and GRASP with $k \in [1, 80]$ and $r \in [1, 320]$ in solving the MARGINALISM problem. We repeated the experiment 10 times on *Book*, each

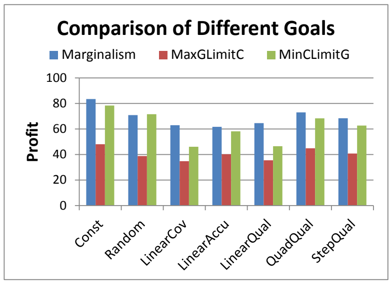


Figure 11: Source selection.

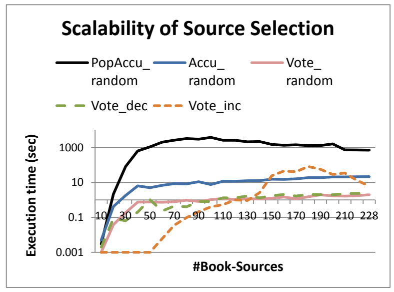


Figure 12: Scalability of selection.

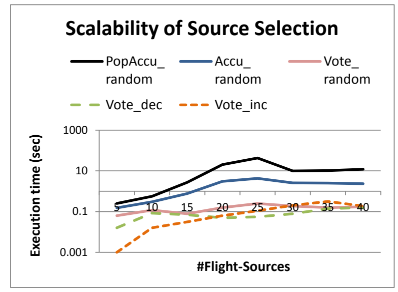


Figure 13: Scalability of selection.

Table 5: Various algorithms for MARGINALISM on the percentage of outputting the best selection and average profit difference from the best selection. Notation (k, r) denotes GRASP with top-k selections and r iterations.

Gain	Cost	Msr	Greedy	(1,1)	(5,20)	(5,320)	(10,320)
Linear	Random	Best	100%	100%	100%	100%	100%
		Diff	-	-	-	-	-
	LinearQ	Best	80%	100%	100%	100%	100%
		Diff	0.4%	-	-	-	-
Quad	Random	Best	90%	100%	100%	100%	100%
		Diff	0.4%	-	-	-	-
	LinearQ	Best	60%	100%	100%	100%	100%
		Diff	0.7%	-	-	-	-
Step	Random	Best	10%	20%	40%	50%	70%
		Diff	14.3%	13.8%	3.7%	2.8%	2.3%
	LinearQ	Best	0	20%	40%	80%	50%
		Diff	19.7%	17.8%	15.4%	2.9%	1.0%

time on randomly selected 150 sources with books in the gold standard. On each data set we compared the selections by various algorithms and chose the one with the highest profit as the best. For each method we reported the percentage of returning the best selection and for sub-optimal selections we reported the average difference on profit from the best selection. Table 5 shows the results for VOTE with RANDOMCOST or LINEARQUALCOST; we have similar observations for other cost models. We observed that (1) GREEDY has the worst performance, and the profit difference can be as high as 19.7%; (2) for LINEARGAIN and QUADGAIN, even GRASP with $k = r = 1$, which essentially is hill climbing, can usually obtain the best solution; and (3) the performance of various methods for STEPGAIN, where the gain can be noncontinuous with fusion accuracy, is much worse; GRASP with $k = 10, r = 320$ often obtains the best selection; even when the solution is not the best, the profit difference is very low.

Fig.14 shows the percentage of finding the best selection, the difference of profit, and the execution time for various combinations of r and k with VOTE, STEPGAIN, and RANDOMCOST on *Book* data. We have three observations. First, not surprisingly, repeating more times takes longer time but can often lead to better results. Second, $k = 10$ often has the highest percentage to obtain the best results and very low profit difference; setting k too low may not find the best solution, while setting k too high is close to random selection and can actually lower the performance. Third, the execution time increased when k was increased from 5 to 20, but then decreased when k went over 20, because when k is large, it is less likely to find a better solution in the random search and so there were fewer iterations in each local search. In the rest of the experiments, we set $r = 200, k = 10$ for STEPGAIN.

Source selection on *Flight* data (we randomly chose 15 sources each time) turns out to be easy. Even GREEDY obtains the optimal results for LINEARGAIN and QUADGAIN, and GRASP with $k = 5, r = 10$ obtains the optimal results for STEPGAIN.

Table 6: Profit difference for various quality measures.

Domain	Gain	Estimated accu	Overall cov	Both
Book	LINEARGAIN	0	.3%	.4%
	QUADGAIN	0	.9%	.9%
	STEPGAIN	.3%	31%	29%
Flight	LINEARGAIN	.8%	.2%	.5%
	QUADGAIN	1.5%	0	1.1%
	STEPGAIN	10.0%	.5%	3.9%

III. Fusion-model comparison: We compared various fusion models and observed quite similar selections on both data sets. For LINEARGAIN and various cost models, on *Book* the profit of VOTE is only 2.7% less than that of POPACCU on average and that of ACCU is only .3% less. In addition, we applied POPACCU on the sources selected by each fusion model, finding that the profit on selections by VOTE and ACCU is only .3% and 1% respectively less than that on selections by POPACCU. On the *Flight* data the four percentages are 1.6%, .1%, 1% and .1% resepectively. This is not surprising because no matter which fusion model we apply, our algorithm tends to select sources with high quality and low cost.

IV. Robustness: We studied the effect of using less accurate quality measures on source selection. In particular, we used the overall coverage and the accuracy computed by applying iterative fusion [3] on source selection. Table 6 shows the average profit difference over various cost models from using the precise measures. We observed that (1) for LINEARGAIN and QUADGAIN, the difference is very small, showing robustness of selection; and (2) for STEPGAIN, the difference is quite large when we use overall coverage on *Book* and estimated accuracy on *Flight*. STEPGAIN can be much more sensitive because the gain is not continuous with fusion accuracy; we did not observe a big difference for non-continuous cost models (RANDOMCOST and STEPCOST) though.

V. Scalability: Finally, we tested the scalability of our algorithm by gradually adding non-zero coverage sources. We added sources in three orders: increasing order of recall, decreasing order, and random order. Fig.12 plots the execution time for LINEARGAIN and LINEARQUALCOST and we have similar observations for other cost and gain models. First, our algorithm is fast. It took 12 minutes for POPACCU on *Book* data and less than 1 minute for any other combination of fusion model and data; this is quite acceptable since source selection is conducted offline and only once a while. Second, the execution time increases slowly after reaching a certain number of sources and may even drop: in random order when we increased the number of *Book* sources from 50 to 228 (3.56 times more), the execution time increased by 1.57 times for VOTE, by 3.32 times for ACCU, but decreased by 35% for POPACCU. This slow growth is because with presence of high-quality and low-cost sources, source selection often starts with those sources and spends very little time on other sources, whose number thus does not affect execution time much. Third, source selection reasons about only quality of data, so the execution time depend not on data size but on data accuracy: source selection took the longest

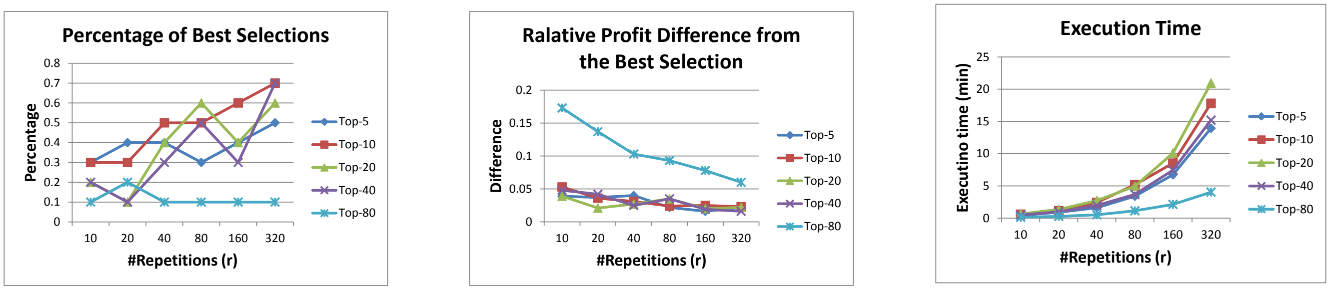


Figure 14: Effectiveness and efficiency of various parameter combinations for GRASP.

time with a large number of small-to-medium sources because of more hill-climbing steps. Fourth, source selection is the slowest for POPACCU and fastest for VOTE, consistent with our observation on accuracy-estimation time reported in Fig.10.

7.2 Experiments on synthetic data

7.2.1 *Selection quality*

We randomly generated 100 data sources with full coverage and accuracy of mean .7 and variance ranging in $[0, .3]$. We conducted (1) comparison of selection goals, (2) comparison of fusion models, (3) comparison of cost and gain models, and (4) comparison of selection algorithms, and reported the results. By default we solve the MARGINALISM problem with LINEARACCUCOST as the cost model, LINEARGAIN as the gain model, and POPACCU as the fusion model, and apply GRASP with $k = 5, r = 50$ for LINEARGAIN and $k = 10, r = 1000$ for STEPGAIN. For each experiment we repeated the experiment 10 times and reported the average.

We conducted experiments on a WindowsXP machine with 3.33 GHz 2 Duo CPU E8600 and 4GB RAM.

Selection-goal comparison: Figure 15 compares the profit and execution time of using various source-selection goals as we vary the variance of accuracy from 0 to .3. We observe that (1) MARGINALISM has the highest profit and MAXGLIMITC has the lowest profit, sometimes even negative profit; (2) MARGINALISM terminates fastest as it would stop when the cost of a source exceeds the maximal possible marginal gain, while MAXGLIMITC terminates slowest as it would try all possibilities as far as the cost budget is satisfied; (3) when the variance is large, there are more sources with very high accuracy, so the benefit is higher and the execution time is shorter; (4) the execution time is highest when the variance is around .06, so there are a lot of sources with medium but different accuracies.

Fusion-model comparison: Figure 15 compares the profit of applying different fusion models. We observe much difference in profit for various models, but the execution time for VOTE is much shorter than that for ACCU and POPACCU.

Cost-model and gain-model comparison: Figure 16 compares the profit and execution time of using various cost models and gain models as we vary the variance of accuracy from 0 to .3 (the profits for both gain models have very similar patterns and we only show that for LINEARGAIN). We observe that (1) RANDOMCOST has the highest profit as high-accuracy sources may have much lower cost, STEPCOST has the lowest cost as high-accuracy sources tend to be most expensive among different cost models; (2) when the variance is large, there are more sources with very high accuracy, so the benefit is higher; (3) under LINEARGAIN model, when we apply STEPCOST and QUADCOST, high-accuracy cost can be very expensive so GRASP took a lot of iterations to converge and was slowest; when we apply RANDOM, GRASP took longest time when accuracy is around .09 so there are a lot of medium-accuracy sources. (4) under STEPGAIN model, GRASP took longest time for various cost models when accuracy is in $[.09, .15]$ so there are a lot of medium-accuracy sources.

Algorithm comparison: Figure 17 compares various algorithms for LINEARGAIN and Figure 18 compares various algorithms for STEPGAIN. We observe that (1) for linear gain model, setting $k = 5$ and repeating 20 times already obtains very good results: although it may not always obtain the best results, the difference is very close to 0; (2) for step gain model, setting $k = 5$ and repeating 200 times can get fairly good results: most of the time it obtains the best results and the difference is close to 0; (3) for both gain models, the naive method and the hill climbing ($k = r = 1$) method are not adequate.

7.2.2 *Selection scalability*

We did two experiments to test the scalability of our algorithm. In both experiments, we randomly generated up to 1M data sources. In the first experiment, the sources have accuracy of mean .7 and variance ranging in $[.05, .3]$. In the second experiment, three sources have a high accuracy of .8, and the rest of the sources have accuracy of mean ranging in $[.1, .8]$ and variance .1. Fig.19 reports source-selection time for LINEARGAIN, LINEARQUALCOST, and VOTE model in both experiments. We have the following observations. First, source selection is fast: even for 1M sources, it took 57 minutes in the worst case. This is acceptable given that source selection is an offline process and is conducted once in a while. Second, source selection is fastest when the sources are skewed; that is, there are (a few) sources with very high accuracy and a lot of sources with very low accuracy. Indeed, in the first experiment, execution time drops from 57 minutes to 2 minutes (3%) when the variance increases from .05 to .3, and in the second experiment, execution time drops from 21 minutes to .01 minute (.1%) when the mean decreases from .4 to .1. This is because the several high-accuracy sources quickly makes the gain close to maximum gain, while the low-accuracy sources can be easily pruned since they cannot add much to the gain. Third, in both experiments, execution time increases linearly in the number of the sources. These observations show high efficiency and scalability of our algorithms.

7.3 Recommendations

We have the following recommendations according to the experimental results.

- MARGINALISM is effective for source selection as far as we can measure cost and gain in the same unit.
- For continuous gain functions, even local search performs quite well and setting $k = 5$ and $r = 20$ seems to be sufficient for GRASP; source selection is quite robust with respect to (sampled) source accuracy and coverage. source selection is much more sensitive for *StepGain*, but setting $k = 10$ and $r = 200$ typically obtains good enough results. On the other hand, different cost models do not seem to make a big difference.

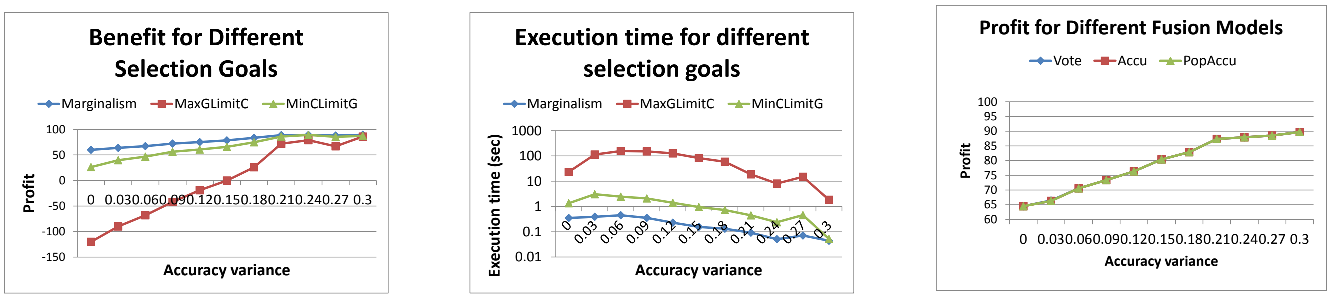


Figure 15: Comparison of various source-selection goals and fusion models.

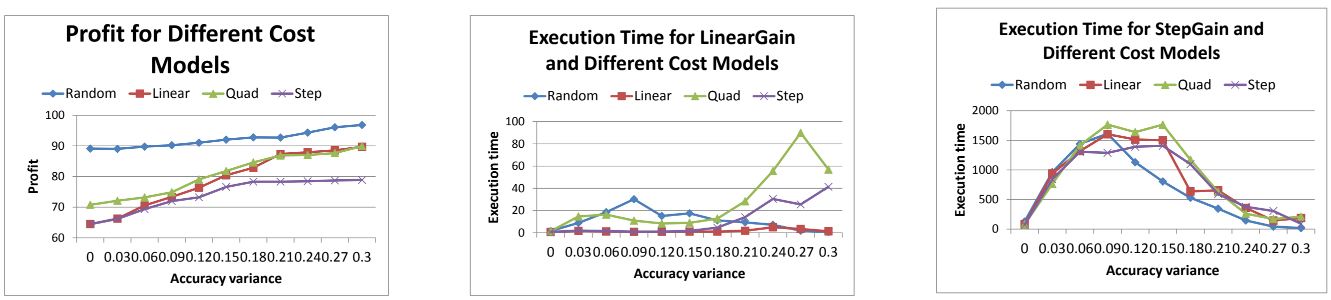


Figure 16: Comparison of various cost and gain models.

- PopAccu is preferred for real fusion, but can be expensive for accuracy estimation. Using Vote in source selection can save a lot of time and generate a set of sources nearly as good as using PopAccu.

8. RELATED WORK

To the best of our knowledge, there has been very little work towards source selection for offline data aggregation. For online data integration, there has been a lot of work on source identification for the hidden Web (see [12] for a survey), but they focus on finding sources relevant to a given query or domain and do not take quality into consideration. There has also been a fair amount of work focused on turning data-quality criteria into optimization goals for query-planning decisions in various contexts (collaborative information system [6, 13, 14, 18, 21], P2P systems [9], sensor networks [17, 20]). In particular, [13] proposed a data model for source quality and studied how to efficiently query such information; [6, 21] proposed incorporating quality requirements in queries; [18] proposed ranking returned answers according to source quality. None of them studies automatic source selection with cost in consideration and they optimize for each individual query. Naumann and Freytag [14] applied the *data envelope analysis* and measured the "efficiency" of each source by maximizing the weighted sum of quality (including *intrinsic quality, accessibility, contextual quality*) minus the weighted sum of cost (including *response time, price*). They did not discuss source selection according to the efficiency and did not consider the marginal quality gain a source can contribute regarding the rest of the sources.

Data fusion has received a lot of recent interest (see [2, 4] for surveys and [8, 15, 16, 23] for recent works). We showed that none of the existing fusion models is monotonic, and proposed a monotonic model. We are unaware of any work that estimates quality for any particular fusion model or for other integration tasks based purely on quality measures of the sources.

9. CONCLUSIONS AND RESEARCH AGENDA

This paper studies source selection with respect to data fusion. We proposed algorithms that can efficiently estimate fusion accuracy and select the set of sources that maximizes the profit. In addition, we proposed a monotonic data-fusion model and show how monotonicity can simplify source selection. Experimental results show effectiveness and scalability of our algorithms.

There are many opportunities to extend this work for full-fledged source selection for data integration. We next lay out a research agenda by describing several future research directions.

Other quality measures: We can consider other quality measures, such as freshness, consistency, redundancy of data. We can also consider relationships between the sources, such as copying relationship, correlation between provided data items, etc. Future work includes efficiently estimating quality of the integrated data and selecting sources given these new measures.

Complex cost and gain models: When we have multi-dimensional quality measures, the gain model can be much more complex. Also, the cost model can be more complex according to some sophisticated pricing strategies [1]. Future work includes providing declarative ways for cost and gain specification and studying their effect on source selection.

Using subsets of data: Different slices of data from the same source can have different quality; for example, a source may provide high-quality data for novels but low-quality data for books of other categories. Research directions include source selection with use of a subset of data from each source.

Other components of data integration: So far we incorporate mistakes in resolving schema and instance heterogeneity in source accuracy. Future work includes treating schema-mapping and entity-resolution as first-class citizens in the picture.

10. REFERENCES

[1] M. Balazinska, B. Howe, and D. Suciu. Data markets in the cloud: An opportunity for the database community. *PVLDB*, 4(12), 2011.

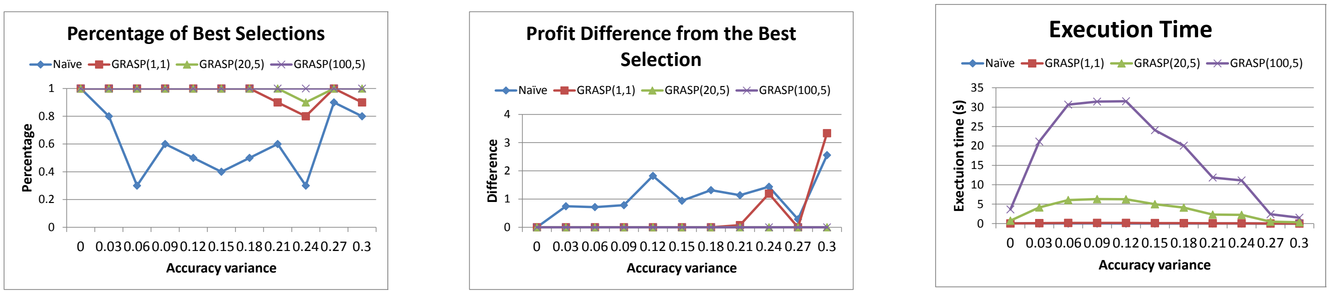


Figure 17: Comparison of various algorithms for LINEARGAIN.

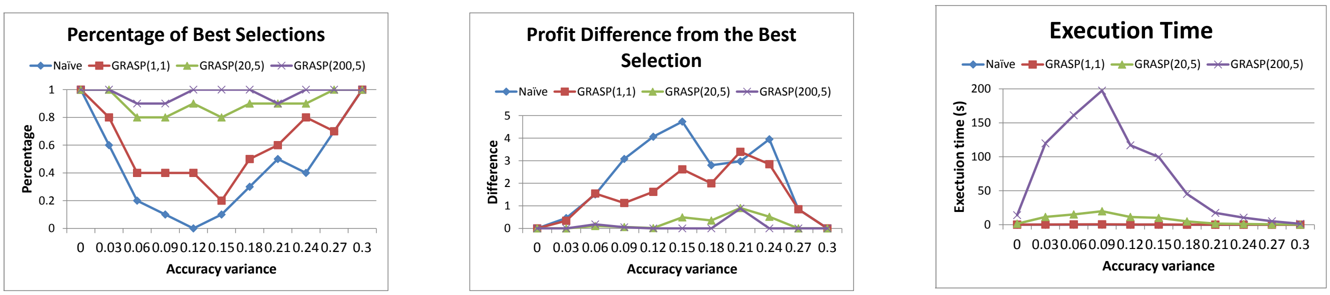


Figure 18: Comparison of various algorithms for STEPGAIN.

[2] J. Bleiholder and F. Naumann. Data fusion. *ACM Computing Surveys*, 41(1):1–41, 2008.

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

[4] X. L. Dong and F. Naumann. Data fusion–resolving data conflicts for integration. *PVLDB*, 2009.

[5] X. L. Dong, B. Saha, and D. Srivastava. Less is more: Selecting sources wisely for integration. http://www2.research.att.com/∼lunadong/publication/marginalism_report.pdf.

[6] S. M. Embury, P. Missier, S. Sampaio, R. M. Greenwood, and A. D. Preece. Incorporating domain-specific information quality constraints into database queries. *J. Data and Information Quality*, 1(2), 2009.

[7] T. Feo and M. G. Resende. Greedy randomized adaptive search procedures. *J. of Global Optimization*, 6, 1995.

[8] A. Galland, S. Abiteboul, A. Marian, and P. Senellart. Corroborating information from disagreeing views. In *WSDM*, 2010.

[9] K. Hose, A. Roth, A. Zeitz, K.-U. Sattler, and F. Naumann. A research agenda for query processing in large-scale peer data management systems. *Inf. Syst.*, 33(7-8):597–610, 2008.

[10] X. Li, X. L. Dong, K. B. Lyons, W. Meng, and D. Srivastava. Truth finding on the deep web: Is the problem solved? http://www2.research.att.com/∼lunadong/publication/webfusion_report.pdf.

[11] A. Marshall. *Pinciples of Economics*. Prometheus Books, 1890.

[12] W. Meng and C. T. Yu. *Advanced Metasearch Engine Technology*. Morgan&Claypool, 2010.

[13] G. A. Mihaila, L. Raschid, and M.-E. Vidal. Using quality of data metadata for source selection and ranking. In *WebDB*, 2000.

[14] F. Naumann, J. C. Freytag, and M. Spiliopoulou. Quality driven source selection using data envelope analysis. In *IQ*, 1998.

[15] J. Pasternack and D. Roth. Knowing what to believe (when you already know something). In *COLING*, pages 877–885, 2010.

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

[17] H. Qu, J. Xu, and A. Labrinidis. Quality is in the eye of the beholder: towards user-centric web-databases. In *SIGMOD*, 2007.

[18] M. A. Suryanto, E.-P. Lim, A. Sun, and R. Chiang. Quality-aware collaborative question answering: Methods and evaluation. In *WSDM*, 2009.

[19] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science (Elsevier)*, 8:189–201, 1979.

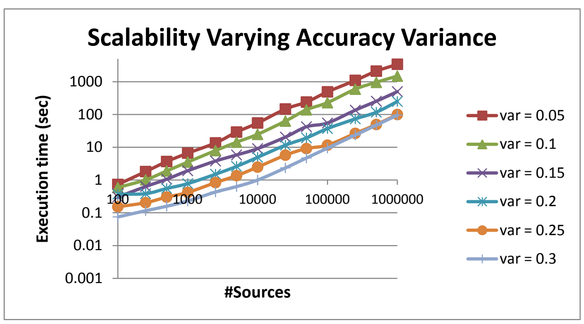
[20] H. Wu, Q. Luo, J. Li, and A. Labrinidis. Quality aware query scheduling in wireless sensor networks. In *DMSN*, 2009.

[21] N. K. Yeganeh, S. Sadiq, K. Deng, and X. Zhou. Data quality aware queries in collaborative information systems. *Lecture Notes in Computer Science*, 5446:39–50, 2009.

[22] X. Yin, J. Han, and P. S. Yu. Truth discovery with multiple conflicting information providers on the web. In *Proc. of SIGKDD*, 2007.

[23] X. Yin and W. Tan. Semi-supervised truth discovery. In *WWW*, pages 217–226, 2011.

[24] B. Zhao, B. I. P. Rubinstein, J. Gemmell, and J. Han. A bayesian approach to discovering truth from conflicting sources for data integration. *PVLDB*, 5(6):550–561, 2012.



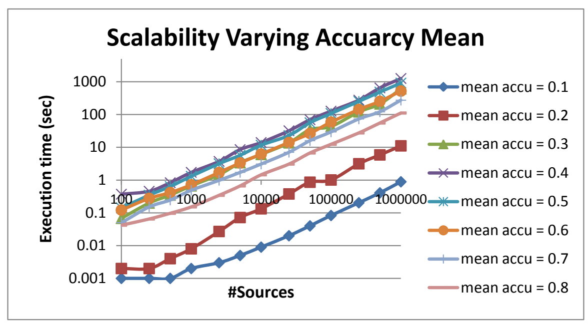


Figure 19: Scalability of source selection.