Estimating the Impact of Unknown Unknowns on Aggregate Query Results

Yeounoh Chung
Brown University
yeounoh_chung@brown.edu

Michael Lind Mortensen
Aarhus University
illio@cs.au.dk

Carsten Binnig
Brown University
carsten_binnig@brown.edu

Tim Kraska
Brown University
tim_kraska@brown.edu

ABSTRACT

It is common practice for data scientists to acquire and integrate disparate data sources to achieve higher quality results. But even with a perfectly cleaned and merged data set, two fundamental questions remain: (1) is the integrated data set complete and (2) what is the impact of any unknown (i.e., unobserved) data on query results?

In this work, we develop and analyze techniques to estimate the impact of the unknown data (a.k.a., *unknown unknowns*) on simple aggregate queries. The key idea is that the overlap between different data sources enables us to estimate the number and values of the missing data items. Our main techniques are parameter-free and do not assume prior knowledge about the distribution. Through a series of experiments, we show that estimating the impact of *unknown unknowns* is invaluable to better assess the results of aggregate queries over integrated data sources.

1. INTRODUCTION

In the past few years, the number of data sources has increased exponentially because of the ease of publishing data on the web, the proliferation of data-sharing platforms (e.g., Google Fusion Table [19] or Freebase [15]), and the adoption of open data access policies, both in science and government. The success of crowdsourcing [11, 13, 30, 29, 38, 2, 51, 16] provides another virtually unlimited source of information. This deluge of data has enabled data scientists, both in commercial enterprises and in academia, to acquire and integrate data from multiple data sources, achieving higher quality results than ever before. It is therefore not surprising that industry and academia alike have developed highly sophisticated systems and tools to assist data scientists in the process of data integration [28]. However, even with a perfectly cleaned and integrated data set, two fundamental questions remain: (1) do the data sources cover the complete data set of interest and (2) what is the impact of any unknown (i.e., unobserved) data on query results?

1.1 Unknown Data

In this work, we develop techniques to estimate the impact of the unknown data on aggregate queries of the form `SELECT AGGREGATE(attr) FROM table WHERE predicate`.

We assume a simple data integration scenario, as depicted in Figure 1. Several domain-related data sources are integrated into one database, preserving the lineage information for each data item or record. Naturally, these data sources overlap with each other, but even when put together they might not be complete. For example, all data sources in

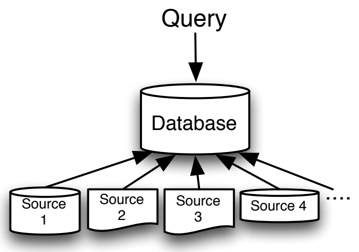


Figure 1: Simple data integration scenario where multiple data sources overlap but are not necessarily complete.

Figure 1 might list U.S. tech companies but some smaller companies might not be mentioned in any of the sources. This data integration scenario applies to a wide range of use cases ranging from crowdsourcing (where every crowd-worker can be considered a single data source [13]) to data extraction from web pages.

Estimating the impact of the unknown data (data items that are not observed in any data source) is particularly difficult as we neither know how many unique data items are missing and their values; thus, we deal with **unknown unknowns**. This characteristic distinguishes our work from what is generally known as *missing data*, or *known unknowns*, estimation in Statistics [1, 43, 40], which tries to estimate the value of unknown (missing) attributes for known records. At a first glance, it may seem impossible to estimate the impact of *unknown unknowns*; however, for a large class of data integration scenarios, the analysis of overlap of multiple data sources makes it feasible.

1.2 A Running Example

To demonstrate the impact of *unknown unknowns*, we pose a simple aggregate query to calculate the number of all employes in the U.S. tech industry, `SELECT SUM(employees) FROM us_tech_companies`, over a crowdsourced data set. We used techniques from [13] to design the crowdsourcing tasks on Amazon Mechanical Turk (AMT) to collect employee numbers from U.S. tech companies.[1] The data was manually cleaned before processing (e.g., entity resolution, removal of partial answers). Figure 2 shows the result.

The red line represents the ground truth (i.e., the total number of employees in the U.S. tech sector) for the query [39], whereas the grey line shows the result of the observed SUM query over time with the increasing number of received crowd-answers. The gap between the observed and

[1]More precisely, we only asked for companies with a presence in Silicon Valley, as we found it provides more accurate results (see also Section 6).

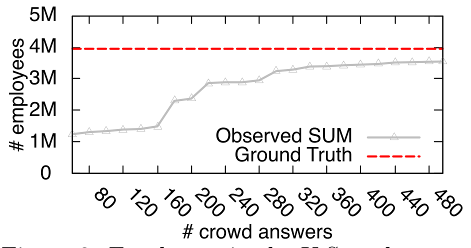


Figure 2: Employees in the U.S. tech sector

the ground truth is due to the impact of the *unknown unknowns*, which gets smaller at a diminishing rate as more crowd-answers arrive.

While the experiment was conducted in the context of crowdsourcing, the same behavior can be observed with other types of data sources, such as web pages. For instance, suppose a user searches the Internet to create a list of all solar energy companies in the U.S. The first few web pages will provide the greatest benefit (i.e., more new solar companies), while after a dozen web pages the benefit of adding another web page diminishes as the likelihood of duplicates increases. The rate of increasing overlap of data sources is indicative of the completeness of the data set.

1.3 A Naïve Solution

The same type of diminishing effect is also known as the *Species Accumulation Curve* in Ecology [47], where the rate of new species discovered decreases with increasing cumulative effort to search. Measuring species richness (i.e., counting species) is critical in many ecological studies. Plotting a *Species Accumulation Curve* provides a way to estimate the number of additional species to be discovered.

These species estimation techniques lay the foundation for estimating the impact of unknown unknowns on aggregate query results. A naïve solution for the SUM query from Section 1.2 would be to first estimate the number of *unknown data* items using species estimation techniques [46] and then use *mean substitution* to estimate their value [37]. This assumes that the missing items have on average the same attribute value as the observed (known) data items.

The naïve approach has a couple of drawbacks. First, species estimation has very strict requirements on how data is collected. Almost every data integration scenario violates these requirements, causing the estimator to significantly over/underestimate the number of missing data items.

Second, it ignores the fact that the attribute values of the missing items may be correlated to the likelihood of observing certain data items. For example, large tech companies like Google with many employees are often more well known and thus, appear more often in data sources than smaller start-ups, creating a biased data set. This is problematic as it also biases the mean and with it the estimate.

In the statistics literature, this second problem is referred to as *Missing Not At Random* (MNAR) [37, 43], where the missingness of a data item depends on its value. There are many statistical inference techniques dealing with MNAR [1, 10, 9, 1, 52, 40], but nearly all the techniques require at least partial knowledge of the record. For example, in the case of surveys, people with a high salary might be more reluctant to report their salary but have no problem stating their home address or how many children they have. Existing MNAR techniques use the reported values (e.g., the address) to infer the missing attributes. Unfortunately, this is not possible in the case of *unknown unknowns*, as we miss the entire record.

1.4 Contributions

This work is a first step towards developing techniques to estimate the impact of the *unknown unknowns* on query results. Our focus is on simple aggregate queries, especially *SUM*-aggregates, but we also touch upon other aggregations like *COUNT*, *AVG*, *MIN*, and *MAX*. We design techniques that can deal with the peculiarities of the data integration scenarios discussed before, such as uneven contributions from different sources (bias of data sources).

In this work, we use crowdsourced data sets because they are easier to collect, but the techniques are general and apply to almost all data integration scenarios that combine overlapping data sources. While we do not argue that the proposed techniques can predict black-swan-like data items (i.e., extremely rare data items), we will show that our techniques can provide useful estimates under more "normal" circumstances, which we will define more formally. For instance, in the example of Figure 2 we can get an almost perfect estimate of the impact of the *unknown unknowns* after only 350 crowd-answers. In addition, by building upon recent work on the Good-Turing estimator [31], we are able to provide an upper bound for our estimates under easy to understand conditions. In summary we make the following contributions:

- We formalize the problem of estimating the impact of *unknown unknowns* on query results and describe why existing techniques for species estimation and missing data estimation are not sufficient.
- We develop techniques to estimate the impact of the *unknown unknowns* on aggregate query results.
- We derive a first upper bound for *SUM*-aggregate queries.
- We examine the effectiveness of our techniques via experiments using both real and synthetitc data sets.

In the following, we first formalize our problem statement (Section 2), presents techniques to estimate the impact of *unknown unknowns* for sum-queries (Section 3) and propose an upper bound estimate (Section 4). Section 5 extends these techniques then to other aggregate functions and in Section 6 we evaluate our techniques, followed by related work and conclusion.

2. THE IMPACT OF UNKNOWN UNKNOWNS

In this Section, we define *unknown unknowns*, explain how data integration over multiple sources can be regarded as a sampling process and formally define our estimation goal. For convenience Appendix A contains a symbol-table.

For the purpose of this work, we treat data cleaning (e.g., entity resolution, data fusion, etc.) as an orthogonal problem. Any data cleaning techniques [1, 12, 22, 34, 35, 33] can be applied to our problem without altering the problem context. While data quality can influence the estimation quality, studying it goes beyond the scope of this paper [46]. We assume that after a proper data cleaning process we have one instance per observed entity and know exactly how many times the entity was observed across multiple data sources.

2.1 Unknown Unknowns

We assume that queries are of the form `SELECT AGGREGATE(attr) FROM table WHERE predicate`, that *table* only contains records about a single entity class (e.g., companies) and that a record in *table* corresponds to exactly one real-

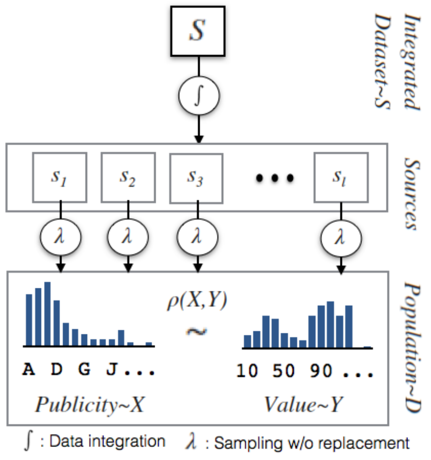


Figure 3: A sampling process for the integrated database.

world entity (e.g., IBM). Thus, in the remainder of the paper we use record, entity and data item interchangeably.

Definition 1. *(Unknown Unknowns) Let Ω be the universe of unknown size of all valid unique entities r for a given entity class and $attr_A(r)$ be the value of attribute A of r. Then the ground truth $D \subseteq \Omega$ is defined as a set of entities that satisfy the predicate, i.e., $D = \{r \in \Omega \mid predicate(r)\}$, where its size $N = |D|$ is not known. Let S be a sample with replacement from D and c be the number of unique entities in S. Unknown unknowns U refers to any unobserved entity r that exists in D but not in S: $U = D - S$ with size $N - c$.*

For our running example, Ω would be the universe of all companies in the world, D all tech companies in the US and $\sum_{r\in D} attr_{empl}(r)$ be the true number of U.S. tech sector employees. S would be a sample with duplicates and *unknown unknowns* would be every company which is not in S.

What we aim to achieve is a good estimate of the ground truth: `SELECT AGGREGATE(attr) FROM D`, when we only have S. Note, that we drop the *predicate* from the query, since every item in D already has to fulfill the *predicate*. In this work, we assume that we neither know all entities in D nor its size (i.e., *open world* assumption). This distinguishes our problem from the problem of *missing data* [37, 43, 44], which refers to incomplete data or missing attribute values.

2.2 Data Integration As Sampling Process

Data integration refers to the process of combining different data sources under a common schema [12]. For the purpose of this work, we assume that data sources are independent samples (e.g., data source are not copies from each other and instead are independently created), and we model the data integration process as a multi-stage sampling process as shown in Figure 3.

We assume l data sources $s_1...s_l$, each sampling $n_j = |s_j|$ data items from the ground truth D (e.g., the complete set of tech companies in the US with their respective number of employees), **without replacement**, as a data source typically only mentions a data item once. We assume further that every data item $d_i \in D$ has a *publicity* likelihood p_i of being sampled, following some distribution X. Likewise, the attribute values (e.g., the number of employees) have a certain likelihood to appear in the ground truth, referred to as *value likelihood*, again following some distribution Y. These two distributions are possibly correlated making the *publicity-value correlation* bigger or smaller than 0: $\rho \neq 0$.

The data sources l are then integrated into a single integrated data set S of size $n_S = \sum_{j=1}^{l} n_j$. Although each source samples without replacement from $N = |D|$ different classes (i.e., unique data item), S contains duplicates because every data source is sampling from the same underlying truth D. If l is sufficiently large, S **approximates a sample with replacement** from D, which is the reason why species estimation techniques work in the first place (we analyze the effects of smaller l in Section 3.4 and 6). The number of unique data items c in S is likely to be smaller than N. In contrast, the end-user only sees a view of S, referred to as the integrated database K (for *Known* data), which contains only one entity per unique entity in S.

This data integration model covers a large class of use cases from web integration to crowdsourcing. In the latter case, each crowd worker can be regarded as a separate data source s_j as it is known that workers also sample without replacement from D [46]. While extremely powerful, there are scenarios where this sampling model does not apply. Most importantly, data sources are not always independent [24]. Furthermore, the number of data sources l has to be large enough to have sufficient overlap between the sources (see Section 6). If any of these assumptions are violated, then only low-quality estimations are possible.

2.3 Problem Statement

We are interested in estimating the impact of *unknown unknowns* (U) to adjust aggregate query results.

Definition 2. *(The Impact of Unknown Unknowns) Given an integrated database K, the impact of unknown unknowns is defined as the difference between the current answer ϕ_K of the aggregate query over the database K and the answer over the ground-truth ϕ_D:*

$$\Delta = \phi_D - \phi_K \tag{1}$$

Our goal is to estimate the answer on the ground-truth by estimating Δ based on S:

$$\hat{\phi}_D = \phi_K + \hat{\Delta}(S) \tag{2}$$

Note that this definition works for all common aggregates including *MIN* and *MAX*, where $\hat{\Delta}$ would be the positive or negative adjustment to the observed *MIN*/*MAX* value.

3. SUM QUERY

In this section, we focus on *SUM*-aggregates to illustrate our estimation techniques. We first formalize the naïve estimator (Section 3.1), which was informally introduced in the introduction. We then develop the *frequency* estimator by making *naïve* estimator more robust to the *publicity-value correlation* (Section 3.2). Afterwards, we describe the more sophisticated *bucket* estimator (Section 3.3). Finally, we develop a *Monte-Carlo* estimator which is better suited for a smaller set of data sources (Section 3.4).

3.1 Naïve Estimator

Estimating the impact of *unknown unknowns* for *SUM* queries is equivalent to solving two sub-problems: (1) estimating how many unique data items are missing (i.e., the *unknown unknowns* count estimate), and (2) estimating the attribute values of the missing data items (i.e., the *unknown unknowns* value estimate). The *naïve* estimator uses the *Chao92* [7] species estimation technique to estimate the number of the missing data items, and *mean substitution* [37] to estimate the values of them.

Let $\phi_K = \sum_{r \in K} attr(r)$ be the current sum over the integrated database, then we can more formally define our *naïve* estimator for the impact of *unknown unknowns* as:

$$\Delta_{naive} = \underbrace{\frac{\phi_K}{c}}_{\text{Value estimate}} \cdot \underbrace{(\hat{N} - c)}_{\text{Count estimate}} \tag{3}$$

$\hat{N}$ is the estimate of the number of unique data items in the ground truth D, and c is the number of unique entities in our integrated database K (thus, $\hat{N} - c$ is our estimate of the number of the unknown data items). ϕ_K/c is the average attribute value of all unique entities in our database K.

3.1.1 Chao92 estimator

Throughout the paper, we use the popular *Chao*92 estimator. Many species estimation techniques exist [3, 6], but we choose *Chao*92 since it is more robust to a skewed publicity distribution. The *Chao92* estimator uses *sample coverage* to predict $\hat{N}$. The sample coverage C is defined as the sum of the probabilities p_i of the observed classes. Since the true distribution $p_1 ... p_N$ is unknown, we estimate C using the Good-Turing estimator [14]:

$$\hat{C} = 1 - f_1/n \tag{4}$$

The f-statistics, e.g., f_1, represent the frequencies of observed data items in the sample, where f_j is the number of data items with exactly j occurrences in the sample. f_1 is referred as *singletons*, f_2 *doubletons*, and f_0 as the *missing data* [4]. Sample coverage measures the ratio between the number of singletons (f_1) and the sample size (n). This ratio changes with the amount of duplicates in the sample. The high-level idea is the more duplicates that exist in our sample S compared to the number of singletons f_1, the more complete the sample is (i.e., higher sample coverage).

In addition, the *Chao92* estimator explicitly incorporates the skewness of the underlying distribution using *coefficient of variance* (CV) γ, a metric that is used to describe the dispersion in a probability distribution [7]. A higher CV indicates a higher variability among the p_i values, while a $CV = 0$ indicates that each item is equally likely (i.e., the items follow a uniform distribution).

Given the publicity ($p_1 \cdots p_N$) that describe the probability of the i-th class being sampled from D, with mean $\bar{p} = \sum_i p_i/N = 1/N$, CV can be expressed as follows:

$$\gamma = \left[\sum_i (p_i - \bar{p})^2 / N \right]^{1/2} / \bar{p} \tag{5}$$

However, since p_i is not available for all data items, CV has to be estimated using the f-statistic:

$$\hat{\gamma}^2 = \max \left\{ \frac{\frac{c}{\hat{C}} \sum_i i(i-1) f_i}{n(n-1)} - 1 \, , \, 0 \right\} \tag{6}$$

The final *Chao92* estimator for $\hat{N}_{Chao92}$ can then be formalated as:

$$\hat{N}_{Chao92} = \frac{c}{\hat{C}} + \frac{n(1-\hat{C})}{\hat{C}} \cdot \hat{\gamma}^2 \tag{7}$$

3.1.2 The Estimator

$\hat{N}_{Chao92}$ is our estimate for N, and comparing this to c provides us with a means of evaluating the completeness of S. By substituting $\hat{N}_{Chao92}$ for $\hat{N}$, the final *naïve* estimator can be written as:

$$\Delta_{naive} = \frac{\phi_K}{c} \cdot (\hat{N}_{Chao92} - c) = \frac{\phi_K \cdot f_1 \cdot (c + \hat{\gamma}^2 n)}{c \cdot (n - f_1)} \tag{8}$$

Note, that the *naïve* estimator does not consider any *publicity-value correlation* and thus tends to over- or under-estimate the ground truth.

3.2 Frequency Estimator

We developed a simple variation of the *naïve* estimator, which makes direct use of the frequency statistics to improve estimation quality. All coverage-based species estimation methods give special attention to the singletons f_1; the data items observed exactly once. The idea is that those items, in relation to the sample size n, give a clue about how well the complete population is covered. A ratio of f_1/n close to 1 means that almost every sample is unique, indicating that many items might still be missing. Conversely, a ratio close to 0 indicates all unique values have been observed several times, decreasing the likelihood of any unknown data. We use a similar reasoning to improve our value estimation. The key idea is that singletons are the best indicator of missing data items, and that their average value might be a better representation of the values of the missing items. Let ϕ_{f_1} be the sum of all singletons, $\sum_{r \in singletons} attr(r)$ and $\hat{N}_{Chao92}$ again be the *Chao*92 count estimate. Then the estimator can be defined as:

$$\Delta_{freq} = \frac{\phi_{f_1}}{f_1} \cdot (\hat{N}_{Chao92} - c) = \frac{\phi_{f_1}\,(c + \hat{\gamma}^2 n)}{n - f_1} \tag{9}$$

While this estimator still does not directly consider the *publicity-value correlation*, it is more robust against popular high-impact data items (i.e., data items with extreme *attribute* values). For example, in our running employee example, big companies that are highly visible like Google or IBM can significantly impact the known value estimate ϕ_K/c. However, through using the average value of the singletons, ϕ_{f_1}/f_1, it is reasonable to assume that those companies will not stay as singletons very long in any sample and thus will not impact the average value for the *unknown unknowns*. This estimator is surprisingly simple and becomes even simpler if we assume $\hat{\gamma}^2 = 0$:

$$\Delta_{freq} = \frac{\phi_{f_1} \cdot c}{n - f_1} \tag{10}$$

Note, that $\hat{\gamma}^2 = 0$ makes it a Good-Turing estimate, which also converges to the ground truth even for skewed publicity values; it might just take a bit longer [7]. While Δ_{freq} is not the best estimator (see Section 6) the simplicity makes it still useful to quickly test if an aggregate query result might be impacted by any *unknown unknowns*.

3.3 Bucket Estimator

The problem with the previous two estimators is that they do not directly consider a correlation between publicity and attribute values. We designed the *bucket* as a first estimator designed for *unknown unknowns* with *publicity-value correlation*. The idea of the estimator is to divide the attribute value range into smaller sub-ranges called buckets, and treat each bucket as a separate data set. We can then estimate the *impact of unknown unknowns* per bucket (e.g., large, medium, or small companies) and aggregate them to the overall effect:

$$\Delta_{bucket} = \sum_i \Delta(b_i) \tag{11}$$

Here Δ_{b_i} refers to the estimate per bucket and both the *frequency* or *naïve* estimator could be used. Using buckets has two effects: First, it provides a more detailed estimate on what types of companies are missing and related to that, second, the value variance per bucket decreases, making the estimate less prune to outliers (e.g., items with extreme low and high values can be "contained" in separate buckets).

The challenge with the *bucket* estimator is to determine the right size for each bucket. If the bucket size is too small, the bucket contains almost no data items. In an extreme case of having a single data item per bucket, no count or proper value estimation is possible. If the bucket size is too big, then the *publicity-value correlation* can still bias the estimate. In fact, the case with a single bucket is equivalent to using just the *naïve* or *frequency* estimator. In the following we describe two bucketing strategies.

3.3.1 Static Bucket

An easy way to define buckets is to divide the observed value range into a fixed n_b number of buckets of size w_i:

$$w_i = \frac{(a_{max} - a_{min})}{n_b} \tag{12}$$

where a_{min} (a_{max}) refers to the min (max) observed attribute value. Afterwards we apply Δ_{naive} per bucket. It is important to note that the estimate goes to infinite with buckets which only contain singletons due to division-by-zero ($n - f_1 = 0$, see equation 8), which can significantly increasing the error of the estimate for very small buckets.

Unfortunately, the optimal number of buckets varies depending on the underlying publicity distribution (see Appendix B). When the publicity distribution is more skewed and correlated to attribute values, some static buckets may contain too few data items, whereas others contain more than enough. The true publicity distribution is not known and we cannot predetermine the right number (or size) of static buckets. To this end, we found that static buckets based estimation is of little practical value.

3.3.2 Dynamic Bucket

To overcome the previously mentioned issues, we developed several alternative statistical approaches to determine the optimal bucket boundaries over time. The most notable are our uses of the error estimate/upper bounds from Section 4 and of treating f_1 as a random variable (see also Section 3.5). Surprisingly, we achieve the best performance across all our real-world use cases and simulations using a rather simple conservative approach, referred to as $\Delta_{Dynamic}$.

The core idea behind our dynamic strategy $\Delta_{Dynamic}$ is to sort the *attribute* values of S and then recursively split the range into smaller buckets only if it minimizes the estimated impact of *unknown unknowns*, i.e., the absolute Δ value. Intuitively, this is controversial since either under- or overestimation could be better for different use cases. However, there is a more fundamental reason behind this strategy.

The Foundation: Whenever we split a data set into buckets, each bucket contains less data than before the split, and the chance of an estimation error increases due to the *law of large numbers* (i.e., the less data the higher the potential variance) [31, 27]. To illustrate this, we consider the simplest case of a uniform *publicity* distribution ($\hat{\gamma} = 0$) and an even bucket split. In this case, we can show that the *Chao*92 estimate for $\hat{N}$ is bigger or equal to the *Chao*92 $\hat{N}$ before the split:

$$\begin{aligned}\hat{N}_{Chao92} = \frac{c}{1 - f_1/n} &= \overbrace{\frac{n \cdot c}{n - f_1}}^{\text{Before split}} \\ &\leq \underbrace{\frac{n_{b1} \cdot c_{b1}}{n_{b1} - f_{1_{b1}}} + \frac{n_{b2} \cdot c_{b2}}{n_{b2} - f_{1_{b2}}}}_{\text{After split}}\end{aligned} \tag{13}$$

When we split the data exactly into halves, it follows that $c_{b1} = c_{b2} = c/2$ (i.e., we split in regard to the unique values). With a uniform publicity distribution, every item is equally likely, and therefore we can assume that both buckets contain roughly the same amount of data after the split: $n_{b1} = n_{b2} \approx n/2$. However, in contrast to n and c, the number of singletons (f_1) can vary significantly between the buckets. In fact, we know that the estimators only stabilize if every item was observed several times [7] and as a consequence n has to be significantly larger than c and c significantly larger than f_1 ($n \gg c \gg f_1$). Therefore, the variance of f_1 is relatively higher than the one of n or c between the buckets and if we split, there is a higher chance that we unevenly distribute the f_1 among the buckets.

To model the uneven distribution of f_1 we introduce another parameter $\alpha \in [0, 1]$ and set $f_{1_{b1}} = \alpha \cdot f_1$ and $f_{1_{b2}} = (1-\alpha) \cdot f_1$. As a result the inequality in equation 13 becomes:

$$\underbrace{\frac{n \cdot c}{n - f_1}}_{\text{Before split}} \leq \underbrace{\frac{\frac{n}{2} \cdot \frac{c}{2}}{\frac{n}{2} - \alpha \cdot f_1} + \frac{\frac{n}{2} \cdot \frac{c}{2}}{\frac{n}{2} - (1 - \alpha) \cdot f_1}}_{\text{After split}} \tag{14}$$

Appendix C shows that the right hand side of the above inequality has its global minimum at $\alpha = 0.5$, which evaluates to $nc/(n - f_1)$ ($\hat{N}$ before split), and that the inequality always holds. Thus, it can be seen that splitting a data set into buckets not only potentially increases the error, but it does so in a monotonic way.

Yet, this does not mean that the *sum* estimate Δ always increases as well. Especially with a *publicity-value correlation*, the overall estimate of Δ over all buckets can still decrease as the average *attribute* values per bucket differ. This is in-line with our original motivation to use buckets, as we wanted to get a more detailed *unknown unknowns* estimate (e.g., how many small companies vs. large companies are missing). Bringing these two observations together, we can assume for many real-world use cases that whenever our estimate of the impact of unknown unknowns Δ increases after a split, it has a significant chance of being caused by the increasing error in $\hat{N}$, whereas when it decreases it potentially improves the estimate due to the more

Algorithm 1: Dynamic bucket generation

```
Input  : $S$
Output: List of buckets
$b_0 = (minValue(S), maxValue(S))$ ;   /* init bucket $b_0$ */
$todo = [b_0]$;                        /* list with $b_0$ */
$\delta_{min} = abs(\Delta(b_0))$ ;    /* $\Delta$ estimate over $b_0$ */
$bkts = []$ ;                          /* final bucket list */
while !$todo.empty$ do
    $b = todo.pop$ ;                   /* remove first element */
    $\delta_{tmp} = \delta_{min} - abs(\Delta(b))$;
    $tmp = (null, null)$ ;             /* Empty pair */
    for $unique\ r \in b$ do
        $(t_1, t_2) = split(b, r.value)$ ;
        if $\delta_{min} > \delta_{tmp} + abs(\Delta(t_1)) + abs(\Delta(t_2))$ then
            $\delta_{min} = \delta_{tmp} + abs(\Delta(t_1)) + abs(\Delta(t_2))$;
            $tmp = (t_1, t_2)$;
        end
    end
    if $tmp \neq (null, null)$ then
        $todo.add(t_1, t_2)$;
    else
        $bkts.add(b)$;
    end
end
return $bkts$;
```

detailed unknown estimate. While it does not always have to be the case (e.g., if the publicity-value correlation is negative) it is still an indicator for many real-world use cases (see Section 6). Based on the observations, we have devised the conservative bucket splitting strategy: only split the bucket if the overall estimate for Δ is minimized.

The Algorithm: Algorithm 1 shows the final algorithm. First we add a bucket which covers the complete value range of S to the *todo* list (line 2) and calculate the current Δ over S (line 3). Note that we take the absolute values of all estimates (Δ) to underestimate the impact of *unknown unknowns* even for the case of having negative attribute values (e.g., net losses of companies). Afterwards, we check recursively if we can split the bucket to minimize Δ until no further "underestimation" is possible (line 5-21).

We therefore remove the first bucket from the *todo* list (line 6) and calculate the Δ over S without the impact of this bucket b (line 7). Note, that during the first iteration δ_{tmp} will be 0. Afterwards, for every unique record in b, we split the current bucket b into two temporary buckets t_1 and t_2 based on the record's attribute value (line 10). If the resulting estimate using this split is bigger than any previously observed minimums (line 11), we set the new minimum to this value (line 12) and temporally store the new buckets (line 13). When the for-loop of line 9-15 finishes and if at least one new bucket was found (line 16), *tmp* will contain the new split point, which minimizes δ for the bucket, and δ_{min} the new minimum value of δ. Those buckets are then added to the *todo* list (line 17) to be checked, if splitting them again would further lower the estimate. On the other hand, if *tmp* is empty, the algorithm wasn't able to further split the bucket and the current bucket without any additional splits is added to the final bucket list (line 19). If no buckets are left in the *todo* list, the algorithm terminates and *bkts* contains the final list of buckets.

3.4 Monte-Carlo Estimator

As our experiments show, the previous estimator actually performs very well (see Section 6). However, what it does not consider is the effect of uneven contributions from data sources (i.e., one data source contains much more data than another) and the peculiarities of the sampling process itself. The *Chao*92 species estimation, like almost all other estimators, assumes sampling *with* replacement, whereas our data sources sample *without* replacement from the underlying ground truth. The reason why the *Chao*92 still works is, that with a reasonably high number of data sources the integrated data source S approximates a sample with replacement [46]. However, with either a small number of data sources or uneven contributions from sources (i.e., some sources are significantly bigger than others), S diverges significantly from a sample done with replacement, resulting in significant over- or under-estimation. In the case of crowdsourcing, the workers which provide significantly more data items than other workers, are referred to as *streakers* [46].

To address these issues, we present a *Monte Carlo*-based (MC) estimator for $\hat{N}$. The idea is that we simulate the sampling process to find the best distribution with its population size N, which best explains the observed sample including how many items s_j every data source j contributes. More formally, given $(s_1, ..., s_l)$ what we seek is a set of parameters Θ (e.g., the distribution parameters) for the MC simulation, which minimize some distance function Γ between the observed data S and the simulated data Q_Θ:

$$\underset{\Theta}{\operatorname{argmin}}\, \Gamma(S, Q_\Theta | l, [s_1, ..., s_l]) \tag{15}$$

Algorithm 2: Monte Carlo method

```
Input  : $\theta_{\hat{N}}$, $\theta_\lambda$, $S$, $[n_1, ..., n_l]$, $nbRuns$
Output: Average distance
1  $E = dist(\theta_{\hat{N}}, \theta_\lambda)$;                /* publicity of $\hat{N}$ items */
2  $\Gamma = 0.0$;                                  /* default value */
3  for $i = 1$ to $nbRuns$ do
4      $Q = []$;                                    /* simulated model */
5      for $j = 1$ to $l$ do
6          $s_i = sample(n_j, E)$;                  /* w/o repl */
7          $Q.add(s_i)$;
8      end
9      $(F_S, F_Q) = indexing(S, Q)$;
10     $F'_S = smooth(F_S, F_Q)$;
11     $\Gamma$ += $klDiv(F'_S, F_Q)$;              /* KL-divergence */
12 end
13 return $\Gamma / nbRuns$;
```

In the following we first describe the MC method for generating Q_Θ with given Θ, the distance function Γ, and finally the search strategy to find the optimal parameter Θ.

3.4.1 Monte-Carlo Method

In contrast to the other estimators, the *Monte-Carlo* estimator requires an assumption about the shape of the underlying *publicity* distribution; in this work, we use an exponential distribution for *publicity*, from which data source j samples n_j data items. Accordingly, the parameter Θ has two components: θ_N specifies the assumed number of data items, and θ_λ governs the shape (skew) of *publicity* distribution. Note, that the assumption of the exponential distributions makes the MC method a parametric model. The goal of the MC simulation is to determine how well θ_N and θ_λ help to explain the observed S.

Algorithm 2 shows our MC algorithm. First, we use an exponential distribution with skew θ_λ to sample *publicity* $(p_1 \cdots p_{\hat{N}})$ for $\theta_{\hat{N}}$ items (line 1). And then we initialize the distance to 0 (line 2). Afterwards we repeat the following procedure *nbRuns* times. For every data source (line 5) we sample n_j data items according to E, but also without replacement (line 6). The sampled items are added to Q to form a histogram (line 7) for the particular run. After simulating l sources, Q contains the simulated version of S.

To finally compare the simulated sample Q with the observed sample S, we make use of the discrete KL-divergence metric [23]. However, this requires transforming S and Q into a frequency statistic and indexing them to ensure that the right items are compared with each other (line 9).

After the indexing we have two comparable frequency statistics for S and the simulation: F_S and F_Q. However, S might contain less than $\hat{N}$ unique data items, for which the KL-divergence is not defined. We therefore adjust F_S and assign a small non-zero probability to the missing extra unique items (line 10). Finally, the two frequency statistics can be compared using the standard KL-Divergence metric and added to the total distance (line 11) and after all the simulation runs the average distance is returned (line 13).

3.4.2 Search Strategy

We can now simulate the observed sampling process leading to S, but we still need a way to find the optimal Θ, which best explains the observed sample S. The difficulty is, that even though the KL-divergence cost function is con-

Algorithm 3: Monte-Carlo based $\hat{N}$ estimation

```
   Input  : $[s_1, ...s_l]$,$c$,$N_{\hat{Chao92}}$,$nbRuns$
   Output: Estimated number of unique data items, $\hat{N}$
1  $D_{KL} = []$;                                          /* KL-divergence */
2  $n = sizes([s_1, ..., s_l])$;                                /* $[n_1, ..., n_l]$ */
3  $\Theta_{\hat{N}} = [c : \frac{(\hat{N}_{Chao92} - c)}{10} : N_{Chao92}]$;
4  $\Theta_\lambda = [-0.4 : 0.1 : 0.4]$;
5  for $\theta_{\hat{N}} \in \Theta_{\hat{N}}$ do
6      for $\theta_{\hat{N}} \in \Theta_\lambda$ do
7          $\Gamma = monteCarlo(\theta_{\hat{N}}, \theta_\lambda, n, n_r)$;          /* Alg 2 */
8          $D_{KL}.add(\Gamma)$;
9      end
10 end
11 $p = curveFit(\Theta_{\hat{N}}, \Theta_\lambda, D_{KL}, 2)$;               /* 2-D curve fit */
12 $[\hat{N}, \lambda] = \underset{\lambda \in [-0.4, 0.4], \hat{N} \in [c, N_{\hat{Chao92}}]}{\arg\min} \{p(\hat{N}, \lambda)\}$ ; /* min on the curve */
13 return $\hat{N}$;
```

vex, the integer variable $\hat{N}$ prevents us from using tractable optimization algorithms (e.g., gradient descent). Furthermore, the the distance function can be quite sensitive to small amounts of noise in D.

We therefore make the estimator more robust by first performing a grid search for Θ (line 5-10). We vary θ_N between $c \leq \hat{N} \leq \hat{N}_{Chao92}$ with a step-size $(\hat{N}_{Chao92} - c)/10$ and θ_λ between $-0.4 \leq \lambda \leq 0.4$ (i.e., almost no to heavy skew) with a step-size 0.1 (line 2 and 3). The step sizes are chosen to be small enough to efficiently model the convex curve, but large enough to be robust to any noise. Afterwards, we fit a two-dimensional curve using least-squares curve fitting (line 11) and return the $\hat{N}_{MC}$ with the minimum D_{KL} on the fitted curve as the final count estimate (line 11).

Finally, to estimate the total difference, we use our *naïve* estimation technique with $\hat{N}_{MC}$. The estimate is more robust and over-estimates less than the original *naïve* estimator as our MC method always penalizes any unmatched unique items in Q. In other words, the MC estimator favors solutions where $\hat{N}$ is closer to the number of observed unique items c.

3.5 Other Estimators

During the course of developing the above estimators, we explored various alternatives. For example, we experimented with alternative static bucket strategies (see also Appendix B). Most importantly though, we noticed that many proposed techniques can actually be combined. For instance, we can use the *frequency* estimator, instead of the *naïve* estimator, with the *bucket* (i.e., *Dynamic Bucket* approach) estimator or the *Monte-Carlo* estimator. More interestingly, we can also combine the *Monte-Carlo* estimator with the *bucket* estimator. However, as the *Monte-Carlo* estimator requires large sample sizes to be accurate, we found that it often decreases the estimation quality. Similarly, we found that the difference between the *naïve* and *frequency* estimators does not help much for the *bucket* approach (see Appendix D). For the experiments we therefore focus on the original techniques rather than the various combinations and included the other results in the appendix.

4. ESTIMATION ERROR UPPER BOUND

In this section, we derive an estimation error upper bound, specifically, the worst case estimation error of the *naïve* estimator (Equation 3). The same upper bound can easily be applied to each bucket in the *bucket* estimator, as well as the *Monte-Carlo* estimator.

To estimate the impact of *unknown unknowns* on *SUM* query results we multiply the estimate for the number of unknown data with the estimate of the values. Hence, we define the worst case estimate as the product of the worst case unknown data count and the worst case value estimate.

The $Chao92$ count estimation is based on *sample coverage* plus a correction for the skew $\hat{\gamma} > 0$. Recent work proposed a tight error bound of the Good-Turing estimator for the ground truth *unknown unknowns* distribution mass (M_0) [31]:

$$M_0 \leq \frac{f_1}{n} + (2\sqrt{2} + \sqrt{3}) \cdot \sqrt{\frac{\log 3/\epsilon}{n}} \tag{16}$$

which holds with **probability at least** $1 - \epsilon$ over the choice of the sample with $n = |S|$. The confidence parameter ϵ governs the tightness of this bound (we use $\epsilon = 0.01$ for 99% confidence). Based on equation 16, we bound $Chao92$:

$$\begin{aligned} \hat{N}_{Chao92} &= \frac{c}{\hat{C}} + \frac{n(1 - \hat{C})}{\hat{C}} \cdot \hat{\gamma}^2 \\ &\approx \frac{c}{\hat{C}} = \frac{c}{1 - M_0} \\ &\leq \frac{c}{1 - (\frac{f_1}{n} + (2\sqrt{2} + \sqrt{3}) \cdot \sqrt{\frac{\log\log 3/\delta}{n}})} \end{aligned} \tag{17}$$

Notice, that we can omit $\hat{\gamma}$ as it only makes the $Chao92$ converge faster, but does not influence the asymptotic estimate, which is based on the sample coverage.

As the distribution of the *mean substitution* ($\frac{\phi_K}{c}$) tend to a normal distribution (*Central Limit Theorem*), we define the worst case estimate of the ground truth *attribute* mean value ($\frac{\phi_D}{N}$) with the help of the sample standard deviation (σ_K):

$$\frac{\phi_D}{N} \leq \frac{\phi_K}{c} + z \cdot \sigma_K \tag{18}$$

Here z controls the confidence of the bound, and we use $z = 3$ based on the three-sigma rule of thumb [49] to have nearly all values with 99.95% confidence lie below the upper bound. The final upper bound is then the simple multiplication of the two worst case estimators (we present the results in Section 6.4):

$$\Delta_{bound} = \frac{(\frac{\phi_K}{c} + z \cdot \sigma_K) \cdot c}{1 - (\frac{f_1}{n} + (2\sqrt{2} + \sqrt{3}) \cdot \sqrt{\frac{\log\log 3/\delta}{n}})} \tag{19}$$

5. OTHER AGGREGATE QUERIES

In this section we describe how the same techniques for *SUM*-aggregates can be applied to other aggregates for estimating the impact of the unknown unknowns.

COUNT: Estimating *COUNT* is easier than *SUM* as it only requires estimating the number of unknown data items, but not their values. For instance, one could either directly use the $Chao92$ estimator or the techniques proposed in [46]. In addition, the *bucket* and *Monte-Carlo* approaches can be used simply by skipping the second step, i.e., not multiplying the estimated count with the value estimates.

AVG: The simplest way to estimate the *AVG* with *unknown unknowns* is to use the *AVG* over the observed sample S (i.e., the law of large numbers). This is reasonable because of the law of large numbers. However, S might be biased due to a *publicity-value correlation* and need to be corrected. One way to deal with the bias is to use our *bucket* approach with a simple modification on how the Δ_b per bucket are aggregated (e.g., weighted average of averages by the number of unique data items ($\hat{N}_{Chao92}$) per bucket).

MAX/MIN: At a first glance, it seems impossible to estimate *MIN* or *MAX* in the presence of *unknown unknowns*. However, we can still do better than simply returning the observed extreme values by reporting when we believe that the observed minimum or maximum value is the true extreme values. This is already very helpful in many integration scenarios and easy to do with our *bucket* estimator. The strategy divides the observed value range of S into consecutive sub-ranges (i.e., buckets); the number of *unknown unknowns* as well as their values are estimated per bucket. If the estimated *unknown unknowns* count in the highest (lowest) value range bucket is zero, then we say that we have observed the true maximum (minimum) value and only then report the highest (lowest) value.

6. EXPERIMENTS

We evaluated our algorithms on several crowdsourced and synthetic data sets to test their predictive power. Crowdsourcing allowed us to generate many real data sets and avoided the licensing issues which often comes with other data sources. We designed our experiments to answer the following questions:

- How does the estimation quality between the different estimators compare on real-world data sets?
- What is the sensitivity of our estimators in regard to data skew (*publicity-value correlation*) and streakers/imbalance of data sources?
- How useful is the upper bound?
- How early are accurate *MIN/MAX* estimates possible?

6.1 Real Crowdsourced Data

We evaluated the estimation techniques on a number of real-world data sets, each gathered independently using Amazon Mechanical Turk, following the guidelines in [13]. Here we chose four representative data sets and four aggregate queries, which show different characteristics we encountered during the evaluation.

1. **US tech revenue & employment**: For the query: *how much revenue does the US tech industry produce?*, i.e., `SELECT SUM(revenue) FROM us_tech_ companies`, we used the crowd to collect US[2] tech company names and revenues. Similarly, in an independent experiment we asked for US tech company names and number of employees, in order to answer the question: *how many people does the US tech industry employ?*, i.e., `SELECT SUM(employees) FROM us_tech_companies`. We selected the two data sets as they exhibit a steady arrival of unique answers from crowd workers.
2. **US GDP**: As a proof-of-concept experiment, we asked crowd workers to enter a US state with its GDP. This data set suffered from streakers.
3. **Proton beam**: Together with researchers from the field of Evidence Based Medicine (EBM) (group-name omitted for double blind reviewing) we created a platform for abstract screening and fact extraction and spent over \$6,000 on AMT, to screen articles about 4 different topics. Here we utilize the results on one of these, namely Proton beam: a set of articles on the benefits and harms of charged-particle radiation therapy for patients with cancer. Part of the abstract screening asked workers to supply the number of patients being studied. The question we aim to answer is *how many people, in total, participated in these type of studies*: `SELECT SUM(participants) FROM proton_beam_studies`. This data set and research question is grounded in a real world problem and unlike the other queries, this one does not have a known answer.

[2]We asked for companies in Silicon Valley to get a representative sample of US tech companies; without restrictions we received too many tiny computer shops and even non-US based companies.

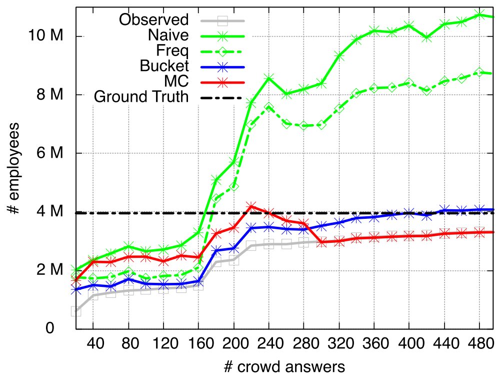


Figure 4: The best US tech-sector employment

We paid between 2 and 35 cents per task. For the Proton beam experiment we designed a qualification test and introduced hidden control tests to filter out bad workers (reference is omitted for double blind reviewing), the other experiments were done without qualification tests. For the purposes of this study, we performed data cleaning manually: if workers disagreed on the value (e.g., the number of employees of a company) we used the average.

In the following we describe the results for every data set and the following estimators: *Naïve* (naive) (Section 3.1), *frequency* (Freq) (Section 3.2), *bucket* (Bucket) (Section 3.3), and *Monte-Carlo* (MC) (Section 3.4) estimators (other estimators did not perform that well or had the same performance and are only shown in Appendix B and D).

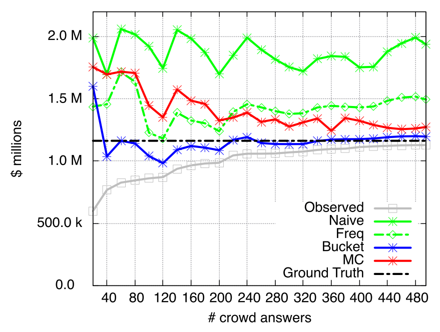
6.1.1 US Tech-Sector Employment

Figure 4 shows the *SUM* estimates from the different estimators (colored lines) for our running example `SELECT SUM(employees) FROM us_tech_companies` as well as the observed *SUM* (grey line) over time (i.e., with an increasing number of crowd-answers). As the ground-truth (dotted black line) we used the US tech sector employment report from the Pew Research Center [39].

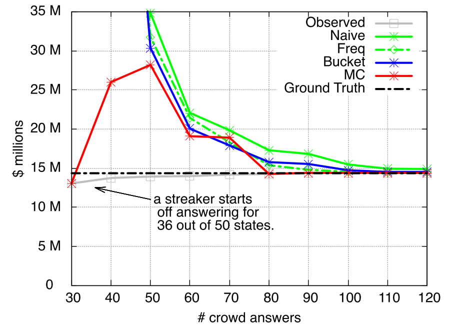
Both the *naïve* and *frequency* estimators heavily overestimate the impact of *unknown unknowns*. The *frequency* estimator does slightly better than the *naïve* estimator, which indicates that some big companies have a high publicity likelihood and were observed early on by several sources.

In contrast, the *MC* estimator does well until it falls back to the observed query result. This can be explained by a peculiarity of this experiment. After roughly 280 crowd-provided data item, all remaining companies have a rather uniform publicity likelihood. In such a case, the *MC* estimator has a tendency to favor count estimates, which are similar to the number of observed items: $\hat{N}_{MC} \sim c$. A major drawback of our *MC* estimation technique.

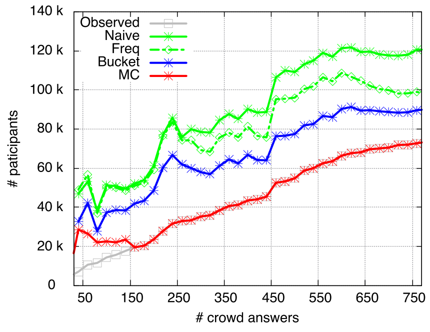
Finally, the *bucket* estimator provides the best estimate (4053160.57 at 500 crowd answers), which is only $\sim$ 2.5% above the ground truth (3951730). While it is possible that the *bucket* estimator might require more data to converge, it is also possible that the ground truth is inaccurate: the employment statistics can vary widely based on many factors



(a) US Tech Revenue



(b) US GDP



(c) Proton Beam

Figure 5: Real data experiments with aggregate SUM query

(e.g., inclusion of part-time employees, tech sector definition). We also speculate that there exist many smaller US tech start-ups that might be overlooked by survey research agencies, due to the high data collection cost. In contrast, a school of crowd workers can more easily find smaller start-ups and their number of employees on web-pages. Thus, the *bucket* estimate could be closer to the ground truth than the one by the Pew Research Center. This is an astonishing result as the cost of crowdsourcing (e.g., \$50.00 per 500 crowd-answers for US tech revenue & employment experiments) is probably only a small fraction of the cost of survey research by any major agency.

6.1.2 US Tech-Sector Revenue

Figure 5(a) shows the results for the US tech-sector revenue. In this data set, both the *naïve* and the *frequency* techniques overestimate the ground truth significantly because of the *publicity-value correlation*. While both estimators will eventually converge to the ground truth, it requires significantly more crowd-answers than what we collected.

Again, both *Monte-Carlo* and *bucket* estimators provide better estimates than *naïve* and *frequency* estimators .Yet, *Monte-Carlo* still overestimates, whereas *bucket* gives an almost perfect estimate after 240 answers. However, it can also be observed that the bucket estimator slightly overestimates at the end of the experiment. This happens because one crowd-worker suddenly reported a few unique smaller companies causing the estimator to believe that there were more. Again, we cannot say with 100% certainty that our assumed ground-truth is actually the real ground truth and the *bucket* estimate might or might not be the real value.

6.1.3 GDP per US State

Figure 5(b) shows the estimate quality for our GDP experiment. To clean the data, we substituted the crowd reported GDP values with the values from [50]. This experiment suffered from streakers, i.e., uneven contributions from crowd workers. A single crowd-worker reported almost all answers in the beginning; this kind of aggressive behavior results in unusually high f_1, which throws off the estimators.

As the figure shows, only the *Monte-Carlo* based technique can actually deal with streakers and provides a reasonable estimate even in the beginning. However, it should also be noted that all estimators converge after 60 samples (for $N = 50$). Furthermore, except for the *Monte-Carlo* estimator, there is no difference between the other estimators.

6.1.4 Proton Beam

Finally, results for Proton beam are shown in Figure 5(c). Again the *Monte-Carlo* estimator follows the observed line, which makes the estimates less interesting. Furthermore, we suspect that the *naïve* and the *frequency* estimators overestimate with constantly increasing number of unique data items (reviewed articles). By manually examining the data set, we confirm that this crowdsourcing experiment did not encounter any streakers, which may cause our estimators (e.g., *bucket*) to fail. Note that the *bucket* estimator converges to roughly $95k$, which we consider to be the best estimate of the number of participants for this particular type of cancer therapy effectiveness study.

6.1.5 Discussion

Overall, our *bucket* estimator has the highest accuracy. The only exception is when streakers are present, making the *Monte Carlo* to perform better. However, it should also be noted, that the run-time of the *Monte-Carlo* estimator is significantly higher than the other estimators. While not a serious issue for our experiments (roughly 3.5s for *Monte-Carlo* vs. 0.2s for *bucket*), it could be significant for larger data sets, as the run-time scales linearly with sample size (the inner loop in Algorithm 2 depends on the sample size). In the remainder we analyze the different estimators in more depths using simulation and make final recommendations about which estimator to use at the end of the section.

6.2 Synthetic Data Experiment

To explore the estimation quality more systematically, we used a synthetic data set with $N = 100$ unique items, each having a single attribute-value ranging from 10 to 1000 ($attr = 10, 20, 30, ..., 1000$). We further simulated the sampling process outlined in Section 2 and used an exponential distribution with parameter λ to model various publicity distributions ($\lambda = 0$: uniform; $\lambda = 4$: highly skewed). Finally, our simulation allowed us to vary the *publicity-value correlation* ($\rho = 0$: no correlation; $\rho = 1$: perfect correlation - the most frequent item also has the largest value).

Figure 6 shows the results for various synthetic data experiments, each of which is repeated 50 times and the results averaged (we omit the error bars for better readability). From left to right, we vary the number of simulated crowd-workers (i.e., sources) from $w = 100$, 10 to 5. From top to bottom, we first assume no publicity skew and no *publicity-value correlation* ($\lambda = 0, \rho = 0$), a for species estimation techniques often ideal scenario, we then show the more realistic scenario with skew and publicity-value correlation ($\lambda = 4, \rho = 1$), and finally simulate an environment where some rare items might contain high values ($\lambda = 4, \rho = 0$).

Ideal: Looking at the top-left figure with a uniform publicity distribution and a hundred workers, we can see that

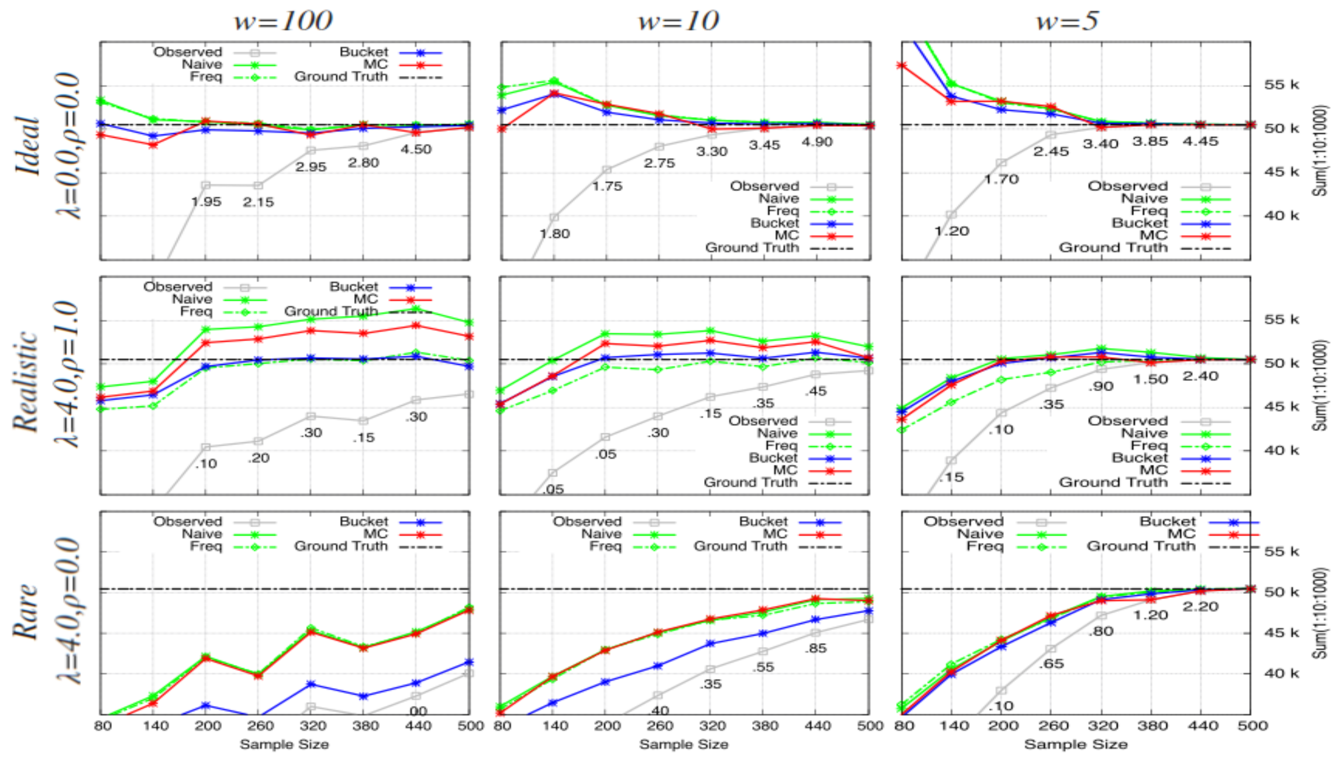


Figure 6: Synthetic data with varying number of sources (w), degrees of publicity skew (λ) & publicity-value correlation (ρ).

all estimators perform very well from the beginning. This is not surprising as all estimators work best with sampling with replacement from a uniform publicity distribution; having many workers sampling without replacement from a uniform distribution approximates sampling with replacement. With fewer numbers of workers sampling from the uniform distribution (top row), all estimators start to overestimate slightly. We conclude, that *under the ideal conditions (i.e., the original assumptions of species estimation technique) all estimators perform equally well.*

Realistic: The middle row shows the scenarios which best resemble real-world use cases as it considers a skewed *publicity* distribution with a positive *publicity-value correlation*. In this case, the *bucket* estimator always provides the best estimates. However, in contrast to the real-world experiments the *frequency* estimator also performs well. This is due to a couple of reasons: Firstly, the *publicity* is highly skewed and perfectly correlated to the values. Secondly, the item values are evenly spaced. This helps the *frequency* estimator to under-estimate as *singletons* consist of only rare low-valued items from the tail – a peculiarity of this simulation. Also interestingly, with 5 evenly contributing workers almost all estimators perform about the same. However, the *bucket* estimator has less variance (not shown). We conclude, that *under the more realistic conditions the bucket estimator performs the best and does not over-estimate the value.*

Rare events: Finally, we see in the bottom row that the *bucket* estimator is not the best choice. This is the case where we have skewed *publicity*, but no *publicity-value correlation*. In fact, all estimators perform poorly in this scenario, even with a lot of data sources (d). As the *publicity* distribution tail can take on any values (i.e., no *publicity-value correlation*, the tail (i.e., *singletons*) can contain many high-impact values or "black-swan" events. In this case, because it conservatively favors underestimation, the *bucket* estimator performs worse. In summary, *none of the estimators are able to predict black-swan events or the long tail; all the estimators underestimate the ground truth.*

6.3 Streakers

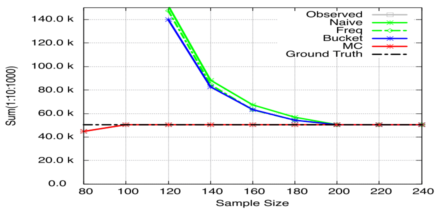
We have seen in Section 6.1.3 that the estimators can heavily overestimate in the presence of streakers. We now examine the effects of streakers using the synthetic data set with $n = 20$, $\lambda = 1.0$ and $\rho = 1.0$.

First, we consider an extreme case where each source successively provides all $N = 100$ data items; first, one data source contributes $n = 100$ items and then the second source starts to contribute its $n = 100$ items, and so on. Figure 7(a) shows that *Monte-Carlo* simply defaults to the observed sum from one source ($n = 100$), whereas all other estimators fail. This is because of the fact that all *Chao*92-based estimators assume a sample with replacement; an assumption which is strongly violated in this case. Only *Monte-Carlo* is more robust against streakers as it tries to best explain the observed S using simulation.

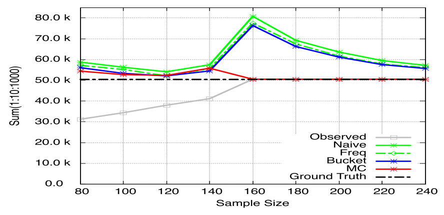
Next, we consider a more moderate case where we inject a single streaker (i.e., an overly ambitious crowd-worker). In Figure 7(b) a streaker is injected at the sample size $n = 160$, contributing all $N = 100$ unique data items directly afterwards. Similar to the previous case, all estimators, except *Monte-Carlo*, heavily overestimate in the presence of a streaker. Again, the reason is that *Monte-Carlo* uses simulation to explain the observed sample S instead of assuming that S was created using sampling *with* replacement.

6.4 Other Queries & Upper Bound

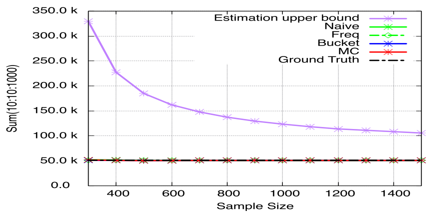
In this subsection we present results for other aggregate queries than *SUM* using the techniques from Section 5. As



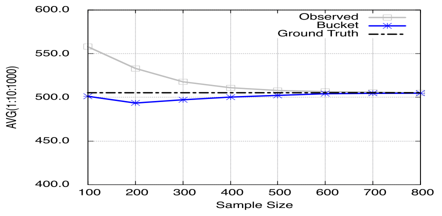
(a) streakers only



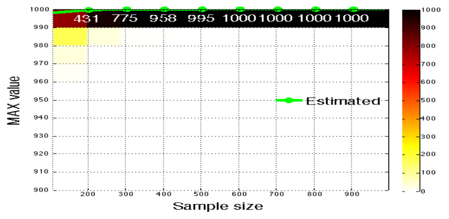
(b) a streaker injected at $n = 160$



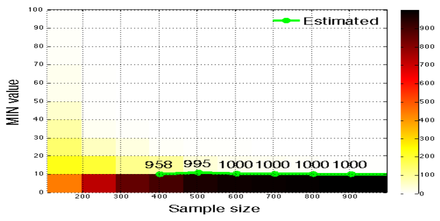
(c) upper bound



(d) AVG



(e) MAX



(f) MIN

Figure 7: Streaker effect (a-b), estimation upper bound (c), AVG query (d) and aggregate MAX/MIN queries (e)(f) experiments using a synthetic data ($\lambda = 1.0$, $\rho = 1.0$: larger values are more likely)

before we use synthetic data with 100 unique data items (e.g., with values $\{10, 20, 30, ..., 1000\}$) integrated over 20 sources with $\lambda = 1.0$ and a publicity-value correlation $\rho = 1.0$. The experiments are repeated 1000 times.

AVG: Figure 7(c) shows the observed (gray line) and estimated (blue line) for a simple average query of the form `SELECT AVG(attr) FROM table`. We only show the *bucket* estimation, as other estimates exactly overlap the observed *AVG* query results (i.e., when all *unknown unknowns* assume the same observed mean value, the *AVG* query result is the same as the observed). As with the sum-aggregates, our dynamic bucket estimator is able to correct the bias of the average because of the publicity-value correlation and provides an almost perfect estimate in this scenario.

MIN/MAX: Figure 7(d-e) compactly visualizes the observed *MIN or* MAX query results. The heat-map shows when the real MIN/MAX value was observed in the data set (the darker the color the more often the result was observed given a number of samples over the 1000 repetitions). The green line shows on average, which value was reported if the *unknown unknowns* count estimate for the highest (*MAX*) / lowest (*MIN*) bucket was zero. The text next to the green line shows how often over the 1000 repetitions the MIN/-MAX value was reported for a given sample size. As it can be seen the average is almost perfect for both *MAX* and *MIN* (note the actual minimum value is 10). That is, whenever our estimation technique for *MAX*/*MIN* reports a value the user can have more trust in it. It should be noted tough, that it is impossible to estimate rare extreme values (black swans). Thus, it is only possible to improve upon the confidence but not eliminate any doubts in the results.

Upper Bound: Finally in Figure 7(f) we show the upper-bound from Section 4 using the same synthetic data set. As it can be seen, the bound is very loose (i.e., very large compared to our estimates) and becomes more tight as we observe more data. We observed the same behavior over the real-world data sets (omitted due to space constraints). While the upper bound provides a valuable insight, it may still be too loose for many real-world scenarios and we hope to improve it in the future.

6.5 Summary

Which Estimator To Use While the *Monte Carlo* or *bucket* estimators always dominate all the others, there is no clear winner between them. The *bucket* estimator performs exceptionally well unless the data sources are imbalanced. It provides the best performance on the real-world use cases (except on the GDP experiment, which suffers from streakers); furthermore, it performs at least as good as other estimators in the simulations from Section 6.2 (except for the *rare event* case, in which all estimators fail to predict black-swan events). However, when the data sources are imbalanced the *Monte Carlo* estimator wins.

The reason is, that the *bucket* estimator is a sample coverage-based method as it uses $Chao92$ and thus, a *nonparametric model*, which does not require assumptions about the underlying distribution. However, it assumes a single sample without replacement. This assumption is not an issue as long as enough independent data sources exists (using simulations we found that 5 sources are often sufficient, see Appendix E) and every data source contributes evenly to S (i.e., there are no streakers).

In contrast, the *Monte-Carlo* estimator is a form of a *Data-Analytic Methods* and really good at adjusting to the specifically observed sampling scenario (i.e., streakers), but at a cost of being a *parametric model*. The method assumes an exponential distribution to model the *publicity* distribution, which can be good or bad depending on the true shape of the underlying distribution. Thus, our recommendation is to use the *bucket* estimator, when the analyst knows that enough data sources contribute evenly to the sample, and, otherwise, to use the more conservative *Monte Carlo* method.

While theoretically the *bucket* estimator should be fairly accurate early on, the authors of [7] found that the $Chao92$ estimator is inaccurate with very low sample coverage C (i.e., observed items are mostly *singletons*) and reported results for cases with $C \geq 0.395$ only. Based on that result, we make the general recommendation to use the estimates if the predicted sample coverage $\hat{C}$ (Equation 4) is greater than 40%.

Trust In The Results With any types of estimators the main question arises: *How can we trust the estimate?* In 1953, Good, who worked with Turing on the estimators, already pointed out that "I don't believe it is usually possible to estimate the number of species ... but only an appropriate lower bound to that number. This is because there is nearly always a good chance that there are a very large number of extremely rare species"[3]. In estimating the Impact of *unknown unknowns*, this statement is even more critical as the rare items can have extreme values.

Yet besides this obvious risks and assumptions, species estimation techniques are extensively used in biology and even helped to decipher the Enigma machine [14]. We actually believe that it comes down to a simple question: *What do you trust more? A potentially wrong answer as no missing data is considered or a potentially wrongly corrected result.* Now knowing, that with enough sources and no imbalance of sources, our *bucket* estimator rather under- than over-estimates, it can generally be said that it can only improve the answer (see the simulations and real-world experiments). With imbalance of and/or only a few data sources, the answer is less clear, as the estimators also more often over-estimate, even the conservative *Monte Carlo* technique (e.g., see Figure 5(b)). Thus, the true answer lies probably somewhere in between. With the help of our upper bound, we can give the user at least a value range and an idea where the true value might be. It should be noted though, that the upper bound requires also two new assumptions: an item probability of at least $1-\epsilon$ and that the value mean follows a normal distribution, which in some rare cases might be violated. Still we believe, knowing something is wrong and a best guess, where the true value might be, is better than staying on the blind-side. In this work, we made a first step in the direction, while a lot remains to be done from developing more tighter bounds, better ways to deal with the imbalance of sources, and easier ways to convey the meaning (and assumptions) of the estimates to the user.

7. RELATED WORK

Traditional query processing assumes the database to be complete (i.e., closed world assumption). Furthermore, nearly all sampling-based query processing techniques assume knowledge of the population size [18]; hence, none of these are suitable for our problem with *unknown unknowns*. To the best of our knowledge, this is the first work on estimating the impact of the *unknown unknowns* on query results (i.e., aggregate query processing in the *open world*).

Species estimation: Most related to this work are the various species estimation techniques, like $Chao92$ [7, 5, 3]. Recent work [48] in this area even tries to estimate the shape of the population (e.g., support size, N). We could use these techniques in place of $Chao92$ to estimate the number of *unknown unknowns*, but not to directly estimate the *impact of unknown*, as the shape does not concern the values of *unknown unknowns*.

Species estimation techniques have also been used to estimate the size of search engine indexes and the deep web [25]. The problem is similar to our *unknown unknowns* count estimation, and the most common technique (i.e., capture-recapture) is also based on the species estimation techniques [26]. However, they again do not consider the *unknown unknowns* value.

Species estimation techniques have also been used in the context of distinct value estimation for a database table [18, 8]. However those techniques leverage the knowledge of the table size to avoid over-estimation.

Survey Methodology & Missing Data: There is a vast body of literature on sampling-based statistical inferences to estimate population statistics [45, 32, 20] or techniques to deal with missingness of values [43, 1, 10, 9, 52].

However, *unknown unknowns* are different from the missing data; missingness refers to the case when the record is known, but one (or more) of the values/attributes is missing. In addition, most of the techniques assume to know the population size to categorize something as missing (e.g., a registered subject participates and leaves before the study completes, a subject deliberately returns an empty questionnaire, only this many subjects out of that many people responded, etc.) and, to some extent, knowing the cause of missingness (e.g., missing completely at random, missing at random, missing not at random) to select appropriate techniques. Moreover, the statistical inference techniques, e.g., multiple imputation based EM/maximum likelihood estimation [1, 10], propensity score estimation [9], or *Markov Chain Monte Carlo* simulation [1, 52]) used to fill the missing variables, require the known non-missing attributes of the record with missing values to be able to use an inference model. In the case of *unknown unknowns*, these assumptions are violated as the entire record (i.e., all attributes) are missing.

Missing data is also well studied in databases [40, 37, 21]; however, as traditional RDBMS query processing function under the *closed world* assumption, they do not consider *unknown unknowns* as part of the query processing and largely consider it a data cleaning aspect.

Recent works [21, 41] defined database completeness in a partly *open world* semantic (i.e., database can be incomplete, which causes incorrect query results) and use the completeness information to denote the completeness of query results. Similar in spirit to our work, they investigate the impact on query results of entire database records that may be missing [41]; however, they also assume the knowledge of population size (e.g., there are 7 days in a week, there are this many cities in France) to define the completely missing records and measure the completeness.

Sampling-Based Query Processing: To cope with aggregates over large data sets, sampling based estimation techniques have been proposed as part of query processing [36, 17, 42]. One limiting aspect of any sampling based estimation techniques, though, is that they assume a complete database (i.e., *closed world*).

8. CONCLUSION

Integrating various data sources into a unified data set is one of the most fundamental tools to achieve high quality answers. However, even with the best data integration techniques, some relevant data might be missing from the integrated data set. In this work, we have developed techniques to quantify the impact of any such missing data on simple aggregate query results. The challenge lies in the fact that the existence and the value of the missing data is unknown. To our knowledge, this is the first work on estimating the impact of *unknown unknowns* on query results.

By nature, our techniques cannot predict black swan events (i.e., extremely rare data items) due to a heavily skewed publicity distribution. However, based on our evaluation results, we believe that the proposed techniques can provide valuable insights for users; rather than blindly believing the closed-world query result, the user gets an idea of what the impact of *unknown unknowns* might be.

There are several interesting future directions. Currently, none of our estimators provides the best performance under all circumstances. The *Monte-Carlo* estimator is very robust against streakers, whereas the *bucket* estimator provides the most accurate results, if no streakers are present. How to develop a robust estimator in all scenarios remains an important area for future work. Similarly, developing a tighter upper-bound for aggregate queries would be of great value. Finally, extending the proposed techniques for more complex aggregate queries (e.g., with joins) also remains open for future work.

This work is an important step towards providing higher quality query results. After all, we live in a big data world where even an integrated data set over multiple sources is possibly incomplete.

9. REFERENCES

[1] P. D. Allison. Handling missing data by maximum likelihood. In *SAS global forum*, pages 1–21, 2012.

[2] S. Amer-Yahia, A. Doan, J. Kleinberg, N. Koudas, and M. Franklin. Crowds, clouds, and algorithms: Exploring the human side of "big data" applications. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, 2010.

[3] J. Bunge and M. Fitzpatrick. Estimating the Number of Species: A Review. *Journal of the American Statistical Association*, 88(421), 1993.

[4] K. P. Burnham and W. S. Overton. Estimation of the Size of a Closed Population when Capture Probabilities vary Among Animals. *Biometrika*, 65(3), 1978.

[5] A. Chao. Nonparametric Estimation of the Number of Classes in a Population. *SJS*, 11(4), 1984.

[6] A. Chao. Species estimation and applications. In *Encyclopedia of Statistical Sciences, 2nd Edition*, pages 7907–7916. Wiley, New York, 2005.

[7] A. Chao and S. Lee. Estimating the Number of Classes via Sample Coverage. *Journal of the American Statistical Association*, 87(417):210–217, 1992.

[8] M. Charikar, S. Chaudhuri, R. Motwani, and V. Narasayya. Towards estimation error guarantees for distinct values. In *Proceedings of the Nineteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '00, pages 268–279. ACM, 2000.

[9] R. B. D'Agostino Jr and D. B. Rubin. Estimating and using propensity scores with partially missing data. *Journal of the American Statistical Association*, 95(451):749–759, 2000.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[11] A. Doan, R. Ramakrishnan, and A. Y. Halevy. Crowdsourcing systems on the world-wide web. *Commun. ACM*, 54(4):86–96, Apr. 2011.

[12] D. Florescu, D. Koller, and A. Y. Levy. Using probabilistic information in data integration. In *Proceedings of the 23rd International Conference on Very Large Data Bases*, VLDB '97, 1997.

[13] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. Crowddb: Answering queries with crowdsourcing. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, SIGMOD '11, 2011.

[14] I. J. Good. The Population Frequencies of Species and the Estimation of Population Parameters. *Biometrika*, 40(3/4), 1953.

[15] Google. Freebase. `https://www.freebase.com`, 2015. Accessed: 2015-07-08.

[16] D. Haas, M. Greenstein, K. Kamalov, A. Marcus, M. Olszewski, and M. Piette. Reducing error in context-sensitive crowdsourced tasks. In *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

[17] P. J. Haas. *Hoeffding Inequalities for Join Selectivity Estimation and Online Aggregation*. IBM, 1996.

[18] P. J. Haas et al. Sampling-based estimation of the number of distinct values of an attribute. In *Proc. of VLDB*, 1995.

[19] A. Y. Halevy. Data publishing and sharing using fusion tables. In *CIDR*, 2013.

[20] L. Kish. *Survey sampling*. John Wiley and Sons, 1965.

[21] W. Lang, R. V. Nehme, E. Robinson, and J. F. Naughton. Partial results in database systems. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 1275–1286. ACM, 2014.

[22] U. Leser and F. Naumann. Query planning with information quality bounds. In H. Larsen, T. Andreasen, H. Christiansen, J. Kacprzyk, and S. Zadrożny, editors, *Flexible Query Answering Systems*, volume 7 of *Advances in Soft Computing*, pages 85–94. Physica-Verlag HD, 2001.

[23] M. Lexa. Useful facts about the kullback-leibler discrimination distance. *Houston, Texas*, 2004.

[24] X. Li, X. L. Dong, K. Lyons, W. Meng, and D. Srivastava. Truth finding on the deep web: Is the problem solved? *PVLDB*, 6(2):97–108.

[25] J. Liang. Estimation Methods for the Size of Deep Web Textural Data Source: A Survey. `cs.uwindsor.ca/richard/cs510/survey_jie_liang.pdf`, 2008.

[26] J. Lu and D. Li. Estimating deep web data source size by capture—recapture method. *Inf. Retr.*, 13(1):70–95, Feb. 2010.

[27] R. Lynch and B. Kim. Sample size, the margin of error and the coefficient of variation. *InterStat*, 2010.

[28] M. Magnani and D. Montesi. A survey on uncertainty management in data integration. *J. Data and Information Quality*, 2(1):5:1–5:33, July 2010.

[29] A. Marcus, E. Wu, D. R. Karger, S. Madden, and R. C. Miller. Demonstration of qurk: a query processor for humanoperators. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2011, Athens, Greece, June 12-16, 2011*, pages 1315–1318, 2011.

[30] A. Marcus, E. Wu, S. Madden, and R. C. Miller. Crowdsourced databases: Query processing with people. In *CIDR 2011, Fifth Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 9-12, 2011, Online Proceedings*, pages 211–214, 2011.

[31] D. A. McAllester and R. E. Schapire. On the convergence rate of good-turing estimators. In *COLT*, pages 1–6. Citeseer, 2000.

[32] J. McClave and T. Sincich. *Statistics*. Pearson, 2013.

[33] W. Meng, K.-L. Liu, C. Yu, W. Wu, and N. Rishe. Estimating the usefulness of search engines. In *Data Engineering, 1999. Proceedings., 15th International Conference on*, pages 146–153, Mar 1999.

[34] F. Naumann, J.-C. Freytag, and U. Leser. Completeness of integrated information sources. *Inf. Syst.*, 29(7):583–615, Sept. 2004.

[35] M. T. Neiling and H.-J. Lenz. Data integration by means of object identification in information systems. In *In Proceedings of European Conference on Information Systems*, 2000.

[36] F. Olken and D. Rotem. Simple random sampling from relational databases. In *VLDB*, volume 86, pages 25–28, 1986.

[37] J. W. Osborne. *Best practices in data cleaning: A complete guide to everything you need to do before and after collecting your data*. Sage, 2012.

[38] A. Parameswaran and N. Polyzotis. Answering Queries using Humans, Algorithms and Databases. In *Proc. of CIDR*, 2011.

[39] Pew Research Center. How u.s. tech-sector jobs have grown, changed in 15 years. `http://pewrsr.ch/PtqZDA`, 2014. Accessed: 2015-07-08.

[40] E. Rahm and H. H. Do. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, 23(4):3–13, 2000.

[41] S. Razniewski, F. Korn, W. Nutt, and D. Srivastava. Identifying the extent of completeness of query answers over partially complete databases.

[42] J. Rice. *Mathematical statistics and data analysis*. Cengage Learning, 2006.

[43] D. B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

[44] B. Saha and D. Srivastava. Data quality: The other face of big data. In *IEEE 30th International Conference on Data Engineering, Chicago, ICDE 2014, IL, USA, March 31 - April 4, 2014*, pages 1294–1297, 2014.

[45] R. Sapsford. *Survey Research*. SAGE Publications, 1999.

[46] B. Trushkowsky, T. Kraska, M. J. Franklin, and P. Sarkar. Crowdsourced enumeration queries. In *ICDE*, pages 673–684, 2013.

[47] K. I. Ugland, J. S. Gray, and K. E. Ellingsen. The species–accumulation curve and estimation of species richness. *Journal of Animal Ecology*, 72(5):888–897, 2003.

[48] G. Valiant and P. Valiant. Estimating the unseen: an n/log (n)-sample estimator for entropy and support size, shown

optimal via new clts. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 685–694. ACM, 2011.

[49] Wikipedia. 68–95–99.7 rule. https://en.wikipedia.org/wiki/68-95-99.7_rule, 2015. Accessed: 2015-07-08.

[50] Wikipedia. List of u.s. states by gdp. https://en.wikipedia.org/wiki/List_of_U.S._states_by_GDP, 2015. Accessed: 2015-07-08.

[51] T. Yan, V. Kumar, and D. Ganesan. Crowdsearch: Exploiting crowds for accurate real-time image search on mobile phones. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 77–90, New York, NY, USA, 2010. ACM.

[52] Y. C. Yuan. Multiple imputation for missing data: Concepts and new development (version 9.0). *SAS Institute Inc, Rockville, MD*, 2010.

APPENDIX

A. SYMBOL TABLE

Ω	Universe of all valid entities (unknown size)
r	A valid unique entity or data item
D	Ground truth or the underlying population
S	Observed sample of size $n = \lvert S\rvert$, with duplicates
K	Integrated database with only unique entities from S
U	*Unknown unknowns* that exist in D, but not in S or K
M_0	*Unknown unknowns* distribution mass in D
c	The number of unique data items in S; $c = \lvert K\rvert$
s_j	Source j with $n_j = \lvert s_j\rvert$ data items
N	The size of the ground truth; $N = \lvert D\rvert$
ϕ	The aggregated query result: e.g., ϕ_D (over D)
Δ	*The impact of unknown unknowns*: $\Delta = \phi_D - \phi_K$
f_j	A frequency statistic, i.e., the number of data items with exactly j occurrences in S.
F	The set of frequency statistics, $\{f_1, f_2, ..., f_n\}$
ρ	The correlation between publicity and value distributions, i.e., *publicity-value correlation*
γ	Coefficient of variance (data skew measure)
C	Sample coverage, also $C = 1 - M_0$

Table 1: Symbols

B. STATIC BUCKET BASED ESTIMATOR

In Section 3.3.1, we state that the optimal number of buckets depends on the underlying *publicity* distribution. Here, we elaborate on this with the two examples.

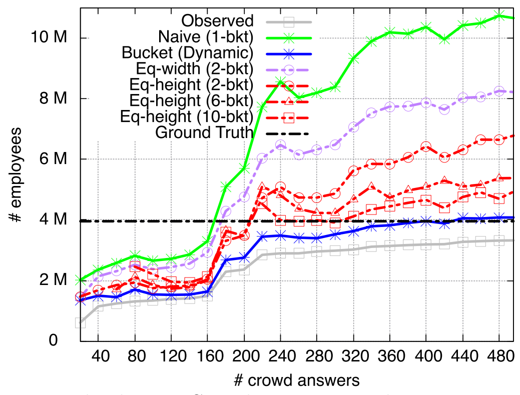


Figure 8: The best US tech-sector employment estimation with static buckets. Splitting into more buckets improves estimation. Eq-width (6-bkt, 10-bkt) are missing due to some of the buckets are empty.

Figure 8 shows the US tech-sector employment estimates by various estimators: *Naive* (1-bucket), *Bucket* (a.k.a., *Dynamic Bucket*), and *Static Bucket* (Eq-width and Eq-height). In this particular example, splitting into more buckets improves estimation, as the underlying *publicity* distribution is skewed and correlated to the values (i.e., larger companies are more well known).

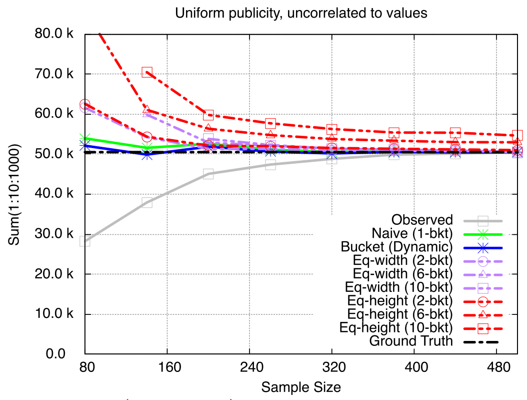


Figure 9: Sum(1:10:1000) estimation with static buckets. Splitting less (e.g., *Naive*) improves estimation. Data points are missing when some buckets contain *singletons* only (i.e., infinite estimation).

In contrast, in the simulated case in Figure 9, splitting into less (e.g., *Naive*) improves estimation as the underlying *publicity* is uniform. Notice, that in both examples above, the *bucket* estimator yields the best estimates, dynamically resizing buckets on its own.

Also notice, that we consider two variants of static buckets: the one described in the paper, **equi-width**, which divides the observed value range into a fixed number of buckets, and another obvious variant, **equi-height**, which divides the observed sample, sorted by value, evenly into a fixed number of buckets. Both static bucket types are simple to use, but they require parameter tuning for the optimal number of buckets, which is hard to predict without knowing the true *publicity* distribution.

C. THE INCREASE IN COUNT ESTIMATE AFTER BUCKET SPLIT

In equation 14, we claimed that the count estimation ($\hat{N}_{Chao92} = nc/(n - f_1)$) of a bucket increases after splitting the bucket, if data items are evenly distributed over the *attribute* value range, and there is no *publicity-value correlation*:

$$\hat{N}_{Chao92} = \frac{c}{1 - f_1/n} = \frac{\overbrace{n \cdot c}^{\text{Before split}}}{n - f_1}$$
$$\leq \underbrace{\frac{\frac{n}{2} \cdot \frac{c}{2}}{\frac{n}{2} - \alpha \cdot f_1} + \frac{\frac{n}{2} \cdot \frac{c}{2}}{\frac{n}{2} - (1 - \alpha) \cdot f_1}}_{\text{After split}}$$

The α parameter governs the split of the original singleton count (f_1) into a pair of smaller buckets. We assume n and c are evenly distributed between the split buckets, as items are evenly distributed over the value range, and all values are equally likely (no *value-publicity correlation*). We now show that the above inequality holds by showing that the right hand side (after split) is minimized at $nc/(n - f_1)$. Note that $nc/(n - f_1)$ is a positive number as $n \geq f_1 \geq 0$

and $c \geq 0$.

To find the minimum, we take the first derivative of the right hand side (denoted by $\mathcal{R}$) with respect to α:

$$\mathcal{R}' = \frac{-c \cdot f_1 \cdot n}{4(-(1-\alpha) \cdot f_1 + \frac{n}{2})^2} + \frac{-c \cdot f_1 \cdot n}{4(-\alpha \cdot f_1 + \frac{n}{2})^2}$$

Solving $\mathcal{R}' = 0$, we get $\alpha = 0.5$; we have $\mathcal{R}(0.5) = nc/(n - f_1)$ as shown below:

$$\mathcal{R}(0.5) = \frac{\frac{n}{2} \cdot \frac{c}{2}}{\frac{n}{2} - 0.5 \cdot f_1} + \frac{\frac{n}{2} \cdot \frac{c}{2}}{\frac{n}{2} - (1-0.5) \cdot f_1}$$
$$= \frac{\frac{n}{2} \cdot \frac{c}{2} + \frac{n}{2} \cdot \frac{c}{2}}{\frac{n}{2} - 0.5 \cdot f_1} = \frac{n \cdot c}{n - f_1}$$

Finally, we show $\mathcal{R}(0.5) = nc/(n - f_1)$ is the minimum by ensuring $\mathcal{R}''(0.5) > 0$:

$$\mathcal{R}'' = \frac{c \cdot f_1^2 \cdot n}{2(-(1-\alpha) \cdot f_1 + \frac{n}{2})^3} + \frac{c \cdot f_1^2 \cdot n}{2(-\alpha \cdot f_1 + \frac{n}{2})^3}$$

$$\mathcal{R}''(0.5) = \frac{c \cdot f_1^2 \cdot n}{2(-(1-0.5) \cdot f_1 + \frac{n}{2})^3} + \frac{c \cdot f_1^2 \cdot n}{2(-0.5 \cdot f_1 + \frac{n}{2})^3}$$
$$= \frac{c \cdot f_1^2 \cdot n}{(-0.5 \cdot f_1 + \frac{n}{2})^3} = \frac{8c \cdot f_1^2 \cdot n}{(-f_1 + n)^3}$$

Note that $n \geq f_1$, and this makes $\mathcal{R}'' > 0$; $\mathcal{R}$ is minimized at $nc/(n - f_1)$ and the inequality holds true:

$$\overbrace{\frac{n \cdot c}{n - f_1}}^{\text{Before split}} \leq \underbrace{\frac{\frac{n}{2} \cdot \frac{c}{2}}{\frac{n}{2} - \alpha \cdot f_1} + \frac{\frac{n}{2} \cdot \frac{c}{2}}{\frac{n}{2} - (1-\alpha) \cdot f_1}}_{\text{After split}}$$

D. OTHER ESTIMATORS

Many proposed techniques can be combined: we can use the *frequency* estimator, instead of the *naïve* estimator, with the *bucket* (i.e., *Dynamic Bucket* approach) estimator or the *Monte-Carlo* estimator. We can also combine the *Monte-Carlo* estimator with the *bucket* estimator.

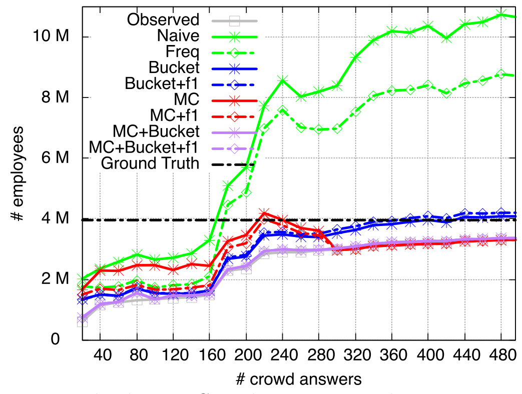


Figure 10: The best US tech-sector employment estimation with other estimators

However, as the *Monte-Carlo* estimator requires large sample sizes to be accurate, combining it with *bucket* estimator often results in lower estimation quality (i.e., each bucket contains a smaller sample). Furthermore, each bucket (a smaller value range) entails a part of the underlying *publicity* distribution; hence, the *publicity* distribution per bucket appears more uniform. As a major drawback, the *Monte-Carlo* estimator exhibits a tendency to favor its count estimate $\hat{N}_{MC} \sim c$ (see Section 6.1.1). Such tendency gets more imminent in *Monte-Carlo* with *Bucket* estimator as seen in Figure 10. Similarly, we found that the difference between the *naïve* and *frequency* estimators is not significant for the *bucket* estimator (i.e., uniform *publicity*).

E. NUMBER OF SOURCES

Bucket estimator is non-parametric and works well with with both uniform and skewed distributions; however, it assumes a sample S sampled with replacement. This assumption is appropriate as long as enough independent data sources contribute evenly to S.

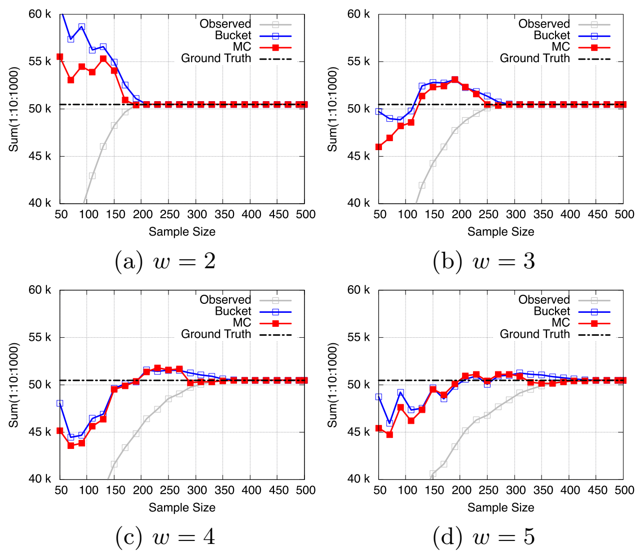


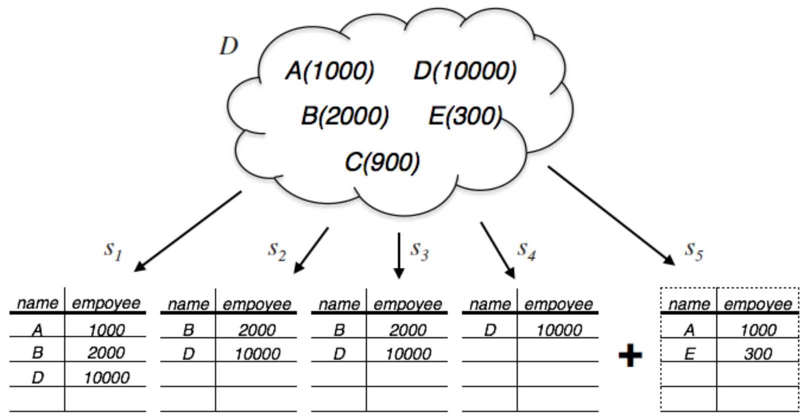
Figure 11: Synthetic data ($\lambda = 4.0$,$\rho = 1.0$) with varying number of sources (w). *Bucket* estimator performs better with more independent sources (i.e., more overlaps).

In Figure 11, we illustrate this with a synthetic data (skewed *publicity* correlated to item *attribute* values). In this particular example, more than 5 sources result in enough overlaps for *bucket* to estimate accurately; however, the minimum number of sources would vary with the date set. In addition, *Monte-Carlo* estimator converges faster as it does not assume a sample sampled with replacement.

F. A TOY EXAMPLE

In this section, we walk through the different estimators step by step using a simple toy example. Again, we use the same query, `SELECT SUM(employee) FROM K`, from the introduction but over a very simplistic data set, shown in Figure 12. It should be noted, that this toy example can not convey any statistical properties because of its small size, but we can explain the general reasoning behind the techniques using the example.

Figure 12 shows the data integration scenario of our example. We assumes that the ground truth D consists of 5 companies $\{A, B, C, D, E\}$ (the bubble on the top), with different numbers of employees (e.g., company A has 1000, whereas company B has 2000). In the beginning we have four data sources $\{s_1, s_2, s_3, s_4\}$ each mentioning some of these companies, thus they sample without replacement from



name	empoyee
A	1000
B	2000
D	10000

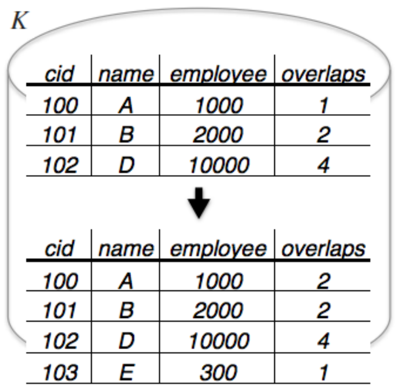
name	empoyee
B	2000
D	10000

name	empoyee
B	2000
D	10000

name	empoyee
D	10000

name	empoyee
A	1000
E	300

(a) Multiple sources s_i sampled without replacement from the unknown population D. s_5 is added later to the original integrated database.



cid	name	employee	overlaps
100	A	1000	1
101	B	2000	2
102	D	10000	4

cid	name	employee	overlaps
100	A	1000	2
101	B	2000	2
102	D	10000	4
103	E	300	1

(b) Integrated Database K, before (top) and after (bottom) adding s_5

Figure 12: A toy example for `SELECT SUM(employee) FROM K`

	before adding s_5 ($n = 7,\ c = 3,\ f_1 = 1,\ \hat{\gamma}^2 = 0.1667$)	after adding s_5 ($n = 10,\ c = 4,\ f_1 = 1,\ \hat{\gamma}^2 = 0$)
Ground Truth	$\phi_D = 1000 + 2000 + 900 + 10000 + 300 = 14200$	
Observed	$\phi_K = 1000 + 2000 + 10000 = 13000$	$1000 + 2000 + 10000 + 300 = 13300$
Naive	$\phi_K + \Delta_{naive} = \phi_K + \frac{\phi_K \cdot f_1 \cdot (c + \hat{\gamma}^2 n)}{c \cdot (n - f_1)}$ $= 13000 + \frac{13000 \cdot 1 \cdot (3 + 0.1667 \cdot 7)}{3 \cdot (7 - 1)}$ ≈ 16009	$= 13300 + \frac{13300 \cdot 1 \cdot (4 + 0 \cdot 9)}{4 \cdot (9 - 1)}$ ≈ 14962
Freq	$\phi_K + \Delta_{freq} = \phi_K + \frac{\phi_{f_1}\,(c + \hat{\gamma}^2 n)}{n - f_1}$ $= 13000 + \frac{1000\,(3 + 0.1667 \cdot 7)}{7 - 1}$ ≈ 13694	$= 13300 + \frac{300\,(4 + 0 \cdot 9)}{9 - 1}$ $= 13450$
Bucket	$\phi_K + \Delta_{bucket} = \phi_K + \Delta_{b_1:\{A,B\}} + \Delta_{b_2:\{D\}}$ $= \phi_K + \{\Delta_{naive}\}_{b_1} + \{\Delta_{naive}\}_{b_2}$ $= 13000 + \frac{3000 \cdot 1 \cdot (2 + 0 \cdot 3)}{2 \cdot (3 - 1)}$ $+ \frac{10000 \cdot 0 \cdot (1 + 0 \cdot 4)}{1 \cdot (4 - 0)}$ $= 14500$	$= \phi_K + \Delta_{b_1:\{A,E\}} + \Delta_{b_2:\{B\}} + \Delta_{b_3:\{D\}}$ $= \phi_K + \{\Delta_{naive}\}_{b_1} + \{\Delta_{naive}\}_{b_2} + \{\Delta_{naive}\}_{b_3}$ $= 13300 + \frac{1300 \cdot 1 \cdot (2 + 0 \cdot 3)}{2 \cdot (3 - 1)}$ $+ \frac{2000 \cdot 0 \cdot (1 + 0 \cdot 2)}{1 \cdot (2 - 0)} + \frac{10000 \cdot 0 \cdot (1 + 0 \cdot 4)}{1 \cdot (4 - 0)}$ $= 13950$

Table 2: `SELECT SUM(employee) FROM K` results with different *unknown unknowns* estimators: *bucket* estimator gives the most accurate estimation of ϕ_D,

D. For instance data source s_1 lists companies A, B, and D. In the example we also assume a *publicity-value correlation*; that is, the biggest company D appears in all data sources ($\{s_1, s_2, s_3, s_4\}$), while smaller companies appear in fewer sources. To show how the estimates improve, we assume that the data source s_5 is added later on (visualized through the plus). The tables in Figure 12(b) show the integrated database before (top) and after (bottom) adding the fifth data source. For convenience, the last column shows, how many times each company was observed across the multiple data sources.

Table 2 shows the estimates by different estimators before and after adding the fifth data source. We exclude *Monte-Carlo* estimator due to its simulation based nature. The top row contains the relevant statistics of K. For instance, with 4 data sources, the number of observed items / sample size is $n = 7$, the number of observed unique items is $c = 3$ (i.e., companies A, B, and D from the top table in Figure 12(b)), the number of singletons $f_1 = 1$ (i.e., company D as it is the only company, which was observed exactly ones across the data sources). and the calculated *coefficient of variance* (CV) $\gamma = 0.1667$ calculated over the sample.

Before adding the fifth data source, the observed total sum is $\phi_K = 1000 + 2000 + 10000 = 13000$, after adding the fifth data source $\phi_K = 1000 + 2000 + 10000 = 13300$. In this example, the observed total sum does not converge to the ground truth of 14200

Table 2 shows the values with calculations for the different estimators. As it can be seen, the naïve estimator performs the worse; the estimator is quite far off, especially with 4 data sources. The reason is the value estimator (*mean substitution*) used. The average number of employees is $\phi_K/3 \approx 4333$. Thus all missing companies (i.e., *unknown unknowns*) are also assumed to be that big. Now knowing that bigger companies are more likely to be sampled, now the naïve estimator heavily over-estimates.

In contrast, the *frequency* estimator performs much better than the naïve estimator because it assumes that the missing companies have the average value over *singletons*, which includes A, but not the extremely big company D; the missing companies are assume to have a value of $\phi_{f_1}/1 = 1000$. Because less popular companies are more likely to be smaller (i.e., the *publicity-value correlation*), this yields to a much better estimate.

Finally, the *bucket* estimator performs the best. Before adding the fifth source, the algorithm creates two buckets: $b_1 : \{A, B\}$ and $b_2 : \{D\}$. The estimate quality of *bucket* persists even after we add s_5 (i.e., *Bucket* is the best). In this case, the *bucket* estimator generates $b_1 : \{A, E\}$, $b_2 : \{B\}$ and $b_3 : \{D\}$. The *bucket* estimator automatically groups the small companies (A and E) together and uses their average number of employees for the missing companies (all other buckets have unknown count estimation of 0); in this example, the *bucket* estimator has a smoothed value in between 300 and 1000. This is particularly more desirable compared to the case of the *frequency* estimator: E is the new one and only *singleton* and ϕ_{f_1} is now 300.