Towards Quantifying Uncertainty in Data Analysis & Exploration

Yeounoh Chung[1], Sacha Servan-Schreiber[1], Emanuel Zgraggen[2], Tim Kraska[2]
[1]Brown University, Providence, RI, USA
[2]MIT CSAIL, Cambridge, MA, USA

Abstract

In the age big data, uncertainty in data constantly grows with its volume, variety, and velocity. Data is noisy, biased and error-prone; blindly applying even the most advanced data analysis techniques can easily mislead users to incorrect analytical conclusions. Also, compounding the problem of uncertain data is the uncertainty in data analysis and exploration. A typical end-to-end data analysis pipeline involves cleaning and processing the data, summarizing different characteristics and running more complex machine learning algorithms to extract interesting insights. The problem is that each step in the pipeline is error-prone and imperfect. From the input to the output, the uncertainty propagates and compounds. This paper discusses the challenges in dealing with various forms of uncertainty in data analysis and provides an overview of our work on Quantifying the Uncertainty in Data Exploration (QUDE), a toolset for safe and reliable data analysis.

1 Introduction

Tableau, scikit-learn, RapidMiner, Trifacta, Tamr, or Vizdom/IDEA are just a small collection of tools [2, 1, 18], which aim to make Data Science more accessible for everyone. However, democratizing Data Science also comes with a risk. For example, many of the "new" users these tools try to reach are not trained in statistics and thus, do not understand the nuances of the algorithms they use. Visual tools, like Tableau, make it possible to quickly test hundreds of hypotheses, and thus, it significantly increases the risk of finding false insights. Data integration tools, like Trifacta and Tamr, make it easier to integrate data, but might also hide data errors. Putting all these together in an end-to-end data analysis and exploration pipeline compounds and further increases the risks of uncertainty.

In this paper, we discuss several potential uncertainty in data analytics, and provide an overview of our work on Quantifying the Uncertainty in Data Exploration (QUDE): a new tool set to automatically quantify the different types of uncertainty/errors within data exploration pipelines. In one hand, there are the obvious types of uncertainty, such as outliers, which can be easily detected via simple visualizations (e.g., error bars, scatter plots). On the other hand, there are various types of uncertainty that are critical for reliable analysis results, but often overlooked. The goal of the QUDE project is to provide techniques for automatically detecting and quantifying such missed types of the uncertainty. This can help users without deep statistical or machine learning backgrounds to derive more safe and reliable data analytical conclusions.

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

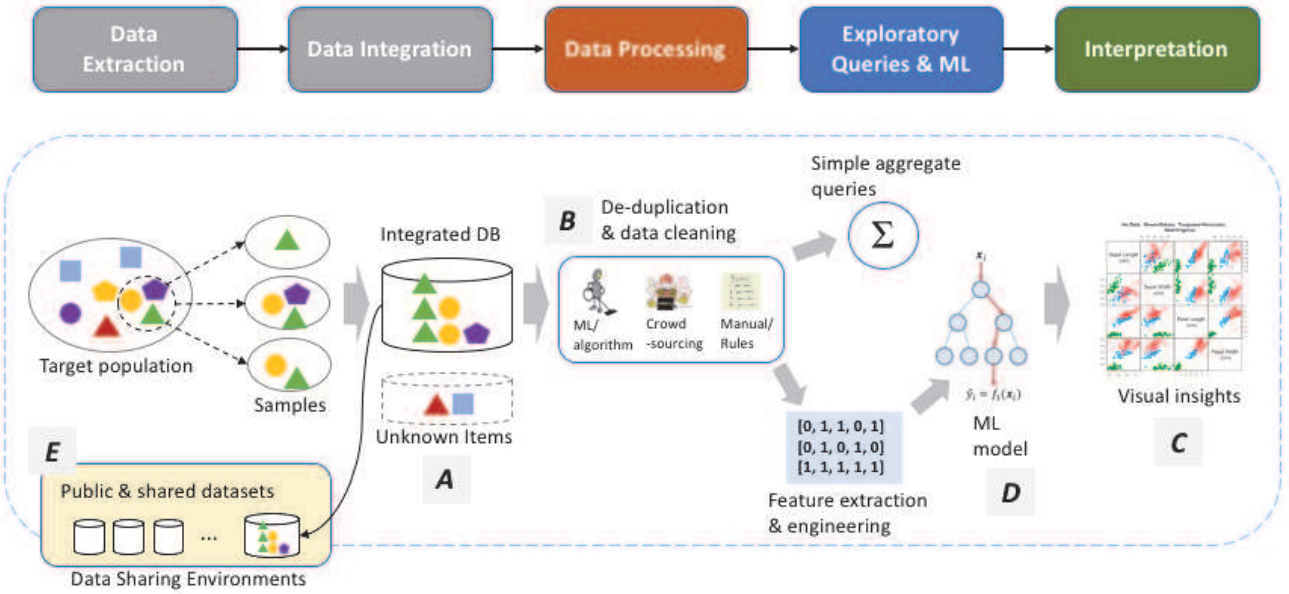

Figure 1: Data analysis & exploration workflow (top) and an example pipeline (bottom).

1.1 Types of Uncertainty in Data Analysis

A typical end-to-end data analysis pipeline involves collecting and cleaning data, summarizing different characteristics and running complex machine learning algorithms. At a high level, a data analysis and exploration workflow involves multiple stages illustrated in Figure 1. Namely, we need to extract high quality data from multiple data sources (or samples), clean the data to remove the inconsistency and merge the same entity in heterogeneous formats (i.e., de-duplication), apply data analysis techniques (e.g., simple aggregate queries or more complex machine learning algorithms) to extract useful information and extract interesting insights. Finally, we need to present the results in the right context for the interpretation.

While similar data analysis best practices can produce actionable insights and discoveries, *one should not take anything at face value*. The size and complexity, noise and incompleteness with big data not only impede the progress of the pipeline, but also make each step in the pipeline more error-prone. The uncertainty around the quality of the intermediate results propagates and compounds, making it even more difficult to validate the output results. Worse yet, the nature of data analysis and exploration tasks requires testing multiple hypotheses, applying different testing procedures with different combinations of data analysis techniques over the input data. The problem is that this "repeated attempts to find interesting facts, increases the chance to observe seemingly significant correlations by chance" [46], which is more formally known as Multiple Comparisons Problem (MCP) [28]. Obviously, many quality-related concerns exist in the entire pipeline from entity resolution problems up to Simpson-Paradox problems for aggregated results. In the following, we therefore highlight five types of uncertainty, which have been less explored in the literature and/or no automatic techniques exist to control the uncertainty type.

A. Missing Unknown Data Items [39, 15, 16]: Incompleteness of the data is one of the most common sources of uncertainty in practice. For instance, if unknown data items are missing (i.e., *we can't tell if the database is complete or not*) from the unknown target population, even a simple aggregate query result, like SUM, can be questionable. It is challenging to make sure that we have collected all the important data items to derive correct data analysis. On the contrary, the traditional survey and sampling methodologies work under a *closed-world* assumption, where there exist no unknown items.

B. Undetected Data Errors [14]: Complicating the challenges of incomplete data is also the quality of the collected data items. Real-world data is noisy and almost always comes with a variety of errors. Such

Table 1: More commonly-studied data analysis errors/uncertainty [12, 36, 26, 31, 35, 19, 20]. QUDE focuses on other types of uncertainty that are crucial for safe and reliable data analysis, but often overlooked in practice.

Data Extraction	Data Integration	Data processing	Exploratory Queries & ML	Interpretation
Sample selection bias	Entity resolution	Combination of error types	Model selection	Human involvement
Data source validity	Un-/semi-structured data	Data cleaning & coverage	Hyper-parameter tuning	Visualization selection
Disparate data sources	Data enrichment	Human involvement	Model bias & variance	Deceptive visualizations
		Missing information	Data under-/over-fitting	
			Feature engineering	
			Concept/distribution shift	

"dirty" data must be removed or corrected because errors can and will bias the results. There are many techniques to identify and repair the errors, but no single technique can guarantee a perfect error coverage. The challenge is that, one should use a number of orthogonal cleaning techniques or hire a lot of crowd-workers without knowing when to stop. Thus, we want to estimate *how many errors are still remaining in the data set*, without knowing the ground truth (a complete/perfect set of constraint rules or the true number of errors in the data set).

C. False Discovery [46]: Extracting insights from data requires repeated analysis and adjustment of hypotheses. With every new hypothesis and insight, the probability of encountering an interesting discovery by chance increases (also known as the MCP). Unfortunately, the problem is often overlooked in data analysis. In fact, many reported results, including published scientific findings, are false discoveries [27]. Thus, it is very important to control the MCP to ensure reliable data analysis and exploration.

D. Model Quality [13]: ML is one of the most popular tools for learning and making predictions on data. For its use, ensuring good ML model quality leads to more accurate and reliable data analysis results. The most common practice for model quality control is to consider various test performance metrics on separate validation data sets (e.g., cross-validation); however, the problem is that the overall performance metrics can fail to reflect the performance on smaller subsets of the data. At the same time, evaluating the model on all possible subsets of the data is prohibitively expensive, which is one of the key challenges in solving this uncertainty problem. Furthermore, missing unknown data items or sampling bias, in general, can also degrade the quality of the model. The challenge is that most ML/inference models perform badly on unseen instances, if similar examples are not learned during training.

E. Data Sharing Environments: When data is shared, a host of new problems increase uncertainty in data analysis. Namely, controlling false discoveries becomes much harder across several institutions or research groups given that the many hypotheses are posed against the shared data. A naïve solution would be to regulate the data sharing all together via a third-party service (*or don't share at all*). But this hinders scientific progress and is too costly to implement.

1.2 Our Goal and Contributions

As part of QUDE, we set out to quantify the uncertainty around the data analysis pipeline, which, in turn, should provide various measures to correct and validate the output results and discoveries. In this work, we provide an overview of our ongoing research, elaborating the uncertainty and its impact on the data analysis and exploration results, as well as the challenges associated with each case of uncertainty. Our initial prototype of QUDE focuses on the above five uncertainty types for two reasons: One, they are important for safe and reliable data analysis; Two, they are overlooked by the common data analysis and data wrangling systems [2, 1, 38, 3, 18]. Table 1 lists other types of errors or uncertainty that are more commonly considered in practice.

In the remainder of this paper, we discuss the above uncertainty cases in more detail and propose techniques to quantify and control the uncertainty (Sections 2, 3, 4, 5, 6); we conclude in Section 7 with some ideas on promising research directions as future work.

2 Uncertainty as Missing Unknown Data Items

First, we look at how uncertainty in a form of missing unknown data items (a.k.a., *unknown unknowns* [15, 16]) affects aggregate query results (e.g., AVG, COUNT, MIN/MAX), which are common in exploratory data analysis. It is challenging to make sure that we have collected all the important data items to derive correct data analysis, especially when we deal with real-world big data; there is always a chance that some items of unknown impacts are missing from the collected data set. To this end, we propose sound techniques to derive aggregate queries with the *open-world assumption* (the data set may or may not be complete).

2.1 An Illustrative Example

To demonstrate the impact of *unknown unknowns*, we pose a simple aggregate query to calculate the number of all employees in the U.S. tech industry, `SELECT SUM(employees) FROM us_tech_companies`, over a crowdsourced data set. We used techniques from [24] 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, whereas the grey line shows the result of the observed SUM query over time with the increasing number of received crowd-answers. As the ground-truth, we used the US tech sector employment report from the Pew Research Center [37]. The gap between the observed and the ground truth is due to the impact of the *unknown unknowns*, which gets smaller at a diminishing rate as more crowd-answers arrive.

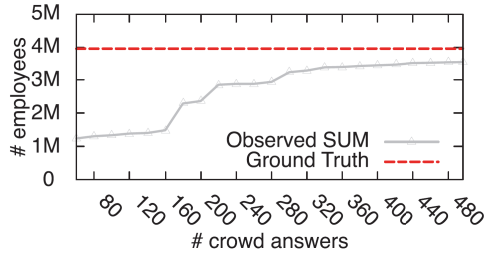

Figure 2: Employees in the U.S. tech sector

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.

2.2 Estimating The Impact of Unknown Unknowns

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 $Chao92$ [9] species estimation technique to estimate the number of the missing data items, and *mean substitution* [36] 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:

$$\Delta_{naive} = \underbrace{\frac{\phi_K}{c}}_{\text{Value estimate}} \cdot \underbrace{(\hat{N} - c)}_{\text{Count estimate}} \quad (1)$$

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

$\hat{N}$ 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.

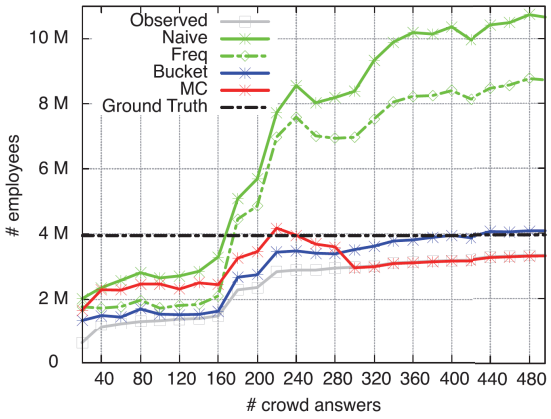

Figure 3: Employees in the U.S. tech sector estimation. While, *Naïve* approach heavily overestimates, *Bucket* estimator achieves the best results.

Figure 3 shows the results of the U.S. tech sector employment estimation. The *Naïve* approach heavily overestimates, since most observed companies are large (i.e., larger companies are more popular and likely to be sampled) and the value estimation is much higher than the true average number of employees. To account for this *publicity-value correlation*, we have proposed several estimators [16], and the *Bucket* estimator yielded the best results across different real-world data sets.

The idea of the *Bucket* 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:

$$\Delta_{bucket} = \sum_i \Delta(b_i) \tag{2}$$

Here $\Delta(b_i)$ refers to the estimate per bucket and both the *frequency* or *naïve* estimator could be used. 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 too few data items for any meaningful estimation. If the bucket size is too big, then the *publicity-value correlation* can still bias the estimate. Our *Bucket* estimator automatically splits the attribute value range to define buckets, which gives the most safe, conservative overall estimates.

The same techniques for *SUM*-aggregates can be applied to other aggregates for estimating the impact of *unknown unknowns*. For more details, as well as other proposed estimation techniques, we refer interested readers to our previous work [16].

3 Uncertainty as Undetected Data Errors

It is almost guaranteed that any real-world data sets contain some types of error (e.g., missing values, inconsistent records, duplicate entities). This is an important source of uncertainty in data analysis because those errors would almost surely corrupt the analysis results. Unfortunately, there has not been much work to measure the data quality or estimate the number of undetected/remaining data errors in a data set; the best practices in data cleaning basically employ a number of orthogonal data cleaning algorithms or crowd-source the task in a hope that the increased cleaning efforts would result in a perfect data set. As part of QUDE, we developed a new Data Quality Metric [14], which can guide the cleaning efforts.

3.1 An Illustrative Example

While this is a seemingly simple task, it is actually extremely challenging to define data quality without knowing the ground truth; previous works define data quality through counting the losses to gold standard data or violations of the constraint rules set forth by domain-specific heuristics and experts [44, 8, 11, 17, 32]. In practice, however, such ground truth data or rules are not readily available and are incomplete (i.e., there exists a "long tail" of errors). For instance, take a simple data cleaning task where we want to identify (and manually fix) malformed US home addresses in the database, shown in Figure 4. As in Guided Data Repair (GDR) [44], we

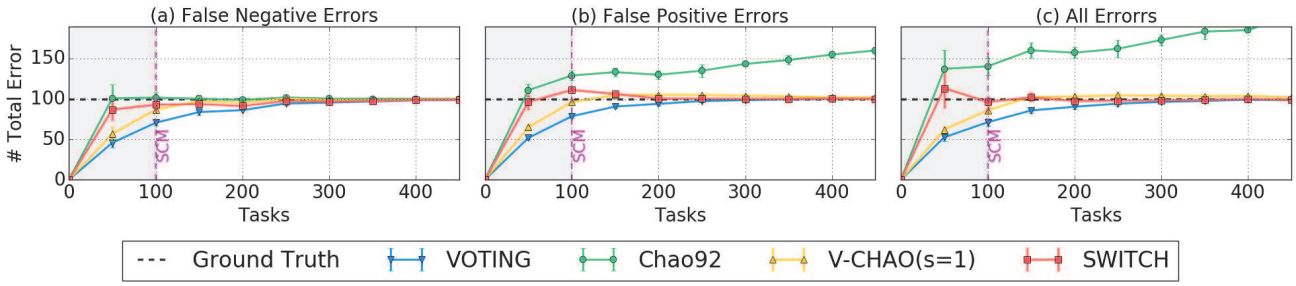

Figure 5: Total error estimates using the simulated datasets. The vanilla species estimation (*Chao92*) heavily overestimates in the presence of false positive errors; *SWITCH* is the most robust estimator against all error types.

might have a set of repair rules for missing values ($r1$, $r2$) and functional dependency violations ($r1$, $r3$, $r6$). However, the repair rules may not cover US state/city name misspellings ($r3$, $r4$) or wrong home addresses ($r5$, $r6$). Once errors are identified, a human can verify the proposed errors and automatic repairs. Similarly, as in CrowdER [42], we can run inexpensive heuristics to identify errors and ask crowd-workers to confirm. In both of these cases, the fallibility of the system in the form of false negative (e.g., "long tail" or missed errors) and false positive (e.g., even humans can make mistakes) errors is a big concern.

3.2 Data Quality Metric (DQM)

We want to design a statistical estimator to address both of the issues. That is, we need to estimate the number of remaining errors without knowing the ground truth in the presence of false negative and false positive errors. A simple approach is to extrapolate the number of errors from a small *"perfectly clean"* sample [43]: (1) we take a small sample, (2) perfectly clean it manually or with the crowd, and (3) extrapolate our findings to the entire data set. For example, if we found 10 new errors in a sample of 1000 records out of 1M records, we would assume that the total data set contains 10000 additional errors. However, this naïve approach presents a *chicken-and-egg* paradox. If we clean a very small sample of data, it may not be representative and thus will give an inaccurate extrapolation or estimates based off it. For larger samples, how can the analyst know that the sample itself is perfectly clean without a quality metric? In our work [14], we propose a *more robust and efficient way to estimate the number of all (eventually) detectable errors*.

	address	city	state	zip
r1:	15440 Southwest Mallard Drive Appartment # 101	Portland		97007
r2:	15440 SW Mallard Drive Apt # 102	Portland	OR	
r3:	12855 Southwest Dipper Lane Apartment # 101	Patland	OR	97007
r4:	289 Angell steet, unit 1H	Providence	RA	2912
r5:	Boston House, 239 S Indian River Dr	Fort Pierce	FL	34950
r6:	12345 ABCD street # EFGH	New York	NY	97007

Figure 4: Erroneous US home addresses: $r1$ and $r2$ contain missing values; $r3$ and $r4$ contain invalid city names and zip codes; $r1$, $r3$, and $r6$ violate a functional dependency ($zip \rightarrow city,\ state$); $r5$ is not a home address, and $r6$ is a fake home address in a valid format.

Interestingly, this problem is related to estimating the completeness of query results using species estimation techniques as first proposed in [40]. For the ease of exposition, let us assume that, as with most practical data cleaning tools, we rely on humans to verify the errors via crowd-sourcing (e.g., CrowdER [42]). Also, to overcome the human errors, we hire multiple workers to review each item. In this setting, we can think of our data quality problem as estimating the size of the set containing all distinct errors that we would discover upon adding more workers. The idea is to estimate the number of distinct records that will be marked erroneous if an infinite number of workers/resources are added, using species estimation techniques. Unfortunately, it turns out that false positives have a profound impact on the estimation quality of how many errors the data set contains.

Figure 5 shows the estimation results using different proposed techniques proposed. Note that the vanilla species estimation (*Chao92*) heavily overestimates in the presence of false positive errors. This is because

species estimators rely on the number of observed "rare" items as a proxy for the number of remaining species, and this number can be highly sensitive to a small number of false positive errors. To cope with the problem, we propose a more robust estimator (*SWITCH*) that estimates how the majority consensus on the items would flip.

Ideally, we want to estimate how many errors are still remaining in a data set. Instead, *SWITCH* estimates the total number of expected switches before the majority consensus converges to the ground truth (i.e., assuming workers are better than a random-guesser, the majority will eventually converge to the ground truth with enough votes). Switches act as a proxy to actual errors and, in many cases, might actually be more informative. However, since a record can switch from clean to dirty and then again from dirty to clean, it is not the same as the amount of dirty records or remaining errors in the data set. We estimate the total number of switches as with infinite workers, using the same $Chao92$ estimation technique; using this estimated quantity, we can adjust the current majority consensus to reach the ground truth:

$$majority(\mathcal{I}) + \xi^{+} - \xi^{-} \tag{3}$$

where positive switch ξ^{+} is defined as switches from the "clean" label to the "dirty" label and negative switch ξ^{-} as switches from "dirty" to "clean." It is important to note that this estimator is more robust against false positives, as it becomes less likely that, as the number of votes per item increases, a false positive will flip the consensus.

4 Uncertainty as False Discovery

Extracting insights from data requires repeated analysis and adjustment of hypotheses. With every new hypothesis and insight, the probability of encountering a chance correlation increases. This phenomenon is formally known as the multiple comparisons problem (MCP) and, when done intentionally, is often referred to as "p-hacking" [27] or "data dredging".

4.1 An Illustrative Example

Suppose we are looking for indicators in a census dataset that affects salary distribution. To examine factors such as "age" or "education", we set up the corresponding *null hypothesis* that states the proposed attribute has no correlation with the salary distribution. We then use a statistical test to infer the likelihood of observing a likewise spurious correlation under the null hypothesis. If this likelihood, commonly referred to as the p-value, is lower than the chosen significance level such as 0.05, then the null hypothesis is rejected, and the *alternative hypothesis* that the proposed attribute is correlated with salary is deemed statistically significant.

However, if we keep searching through different indicators in the dataset, we are almost guaranteed to find a statistically significant correlation. For example, choosing a significance level for each test of 0.05 means that statistically we have a 5% chance of falsely rejecting a given null hypothesis; even if the dataset contains completely random data, we would, on average, falsely discover a spurious correlation that passes our significance level after only 20 hypothesis tests.

4.2 Safe Visual Data Exploration

Visual data exploration tools such as Vizdom [18] or Tableau amplify this problem by allowing users to examine lots of visual "hypotheses" (e.g., comparing visualizations) in a short amount of time. In one of our experiments [45], where we used synthetic data sets with known ground truth labels, we found that by not accounting for all comparisons made during exploration, users are left with a high rate of false discoveries even if user-generated insights are followed up with statistical tests. Perhaps even more concerning is the increasing trend towards creating recommendation engines that propose interesting visualizations [41, 22] or automatically test for correlations [10]. Those systems are potentially checking thousands of hypotheses in just a few seconds. As a result,

it is almost guaranteed that such a system will find something "interesting" regardless of whether the observed phenomenon is statistically relevant or not [7].

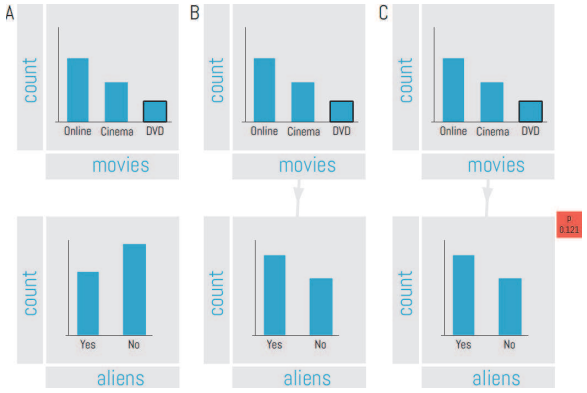

Figure 6: Example of a visualization network where users might be led to false discoveries without automatic hypothesis formulation. (A) two separate visualizations showing preferences for watching movies and how many people believe in alien existence; (B) the two visualizations combined where the bottom one shows proportions of belief in alien existence for only people who like to watch movies on DVD, displaying a noticeable difference compared to the overall population. (C) same visualizations as before but now with automatic hypothesis formulation turned on, highlighting that the observed effect is not statistically significant.

While there exists a variety of statistical techniques to control for the MCP [21, 5] they are not easily applicable in visual data exploration tools as they require knowledge about all the hypotheses being evaluated upfront, whereas in this context, the hypotheses are generated incrementally. To address this, we developed an MCP procedure [46] that allows specifying hypotheses incrementally. Furthermore, we show preliminary results on how such a procedure can be fully integrated into a data exploration system where visual comparisons are automatically tracked and controlled for [47]. Figure 6 shows an example of this. When we analyzed data from a survey on personal habits and opinions [7], we observed that the preference on watching films on DVD produced visually different proportions of belief in aliens, as shown in Figure 6 (A and B). Just by visually examining these charts, users often falsely assumed that people who prefer to watch movies on DVD are more prone to believe in aliens even though this effect is not statistically significant. When automatic testing and tracking is turned on, the system will try to formulate hypotheses for such cases (e.g., when users are comparing subsets against the global population), include them to the MCP control procedure and inform the user about the outcome of the test (Figure 6 C).

5 Uncertainty as Model Quality

ML is one of the most popular tools for uncovering hidden insights and patterns in data analysis. In fact, ensuring model quality leads to more accurate and reliable data analysis results. In this section, we discuss ML model quality as a form of uncertainty in data analysis. Namely, we look at a couple ML model quality issues that are often overlooked in practice. Model validation and quality assurance is an important component for QUDE.

5.1 An Illustrative Example

To ensure that a given model is performing well at a given task, people consider various test performance metrics (e.g., log loss, accuracy, recall, etc.). The problem, which is often overlooked is that the overall performance metrics can fail to reflect the performance on smaller subsets of the data. For example, we want to avoid a model that works well on average with the entire customer data, but fails with a female, teenage demographic in the U.S., especially, if it is one of the key market segments for the application. Here, we present an automated data slicing tool for model validation. The key challenge there is identifying a proper subset or data slice that is large, problematic and interpretable to the user; the search space is exponentially large with the number of features (and their value ranges).

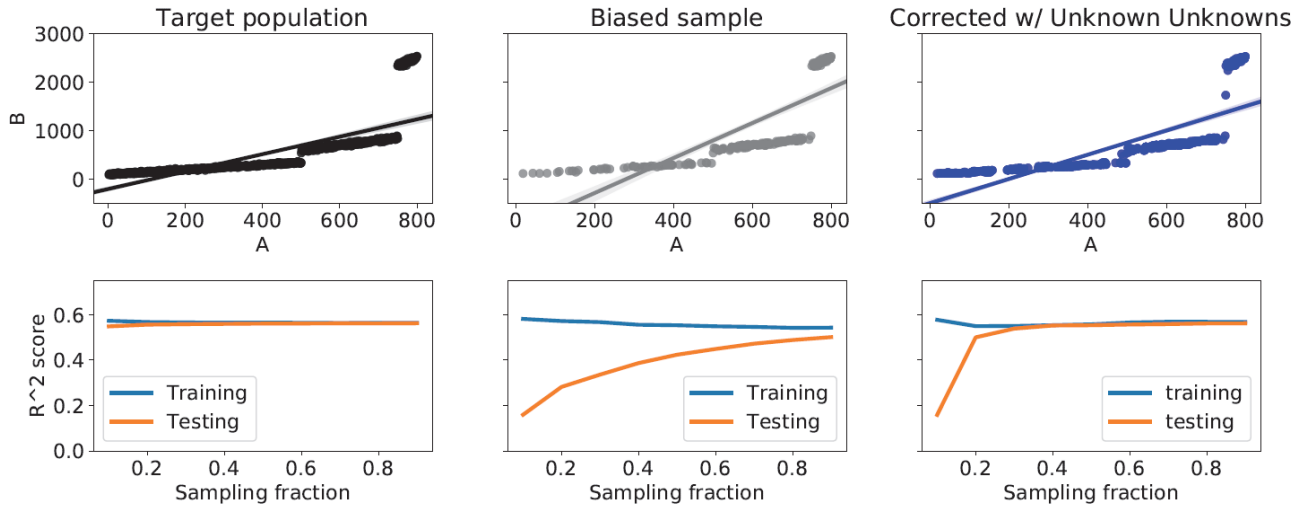

Figure 7: Ideally, we want the generalization gap between the training score and the testing score to be minimal (left); however, the model performs much worse if trained on a biased sample or fails to generalize to the actual testing data (middle). Accounting for *unknown unknowns* can improve the model generalizability (right).

5.2 Automated Data Slicing

While a well-known problem [33], current techniques to determine under-performing slices largely rely on domain experts to define important sub-populations (or at least specify a feature dimension to slice by) [30]. Unfortunately, ML practitioners do not necessarily have the domain expertise to know all important under-performing slices in advance, even after spending a significant amount of time exploring the data. In [13], we present an automated data slicing tool for model validation, called *Slice Finder*. The goal of *Slice Finder* is to identify a handful (e.g., top-K) of the largest problematic slices, that are also interepretable. Larger slices are preferable because they carry more examples, and thus, more impactful to model quality. On the contrary, debugging the model on a tiny slice would not mean much to the overall model quality. Plus, we want to bring the user's attention to the slices that are interpretable. For instance, country = US is more interpretable than country = US $\wedge$ age = 20-40 $\wedge$ zip = 12345, with a fewer number of common features. We find that the interepretability is a key for understanding the model quality problem. The resulting slices are presented via an interactive visualization front-end, which helps users to quickly browse through the slices.

5.3 Unknown Unknowns for ML

Another important aspect of ML model quality is generalizability, which measures how accurately an ML model can predict on new unseen examples. This is also important because ML-model-based analysis is often done to forecast or predict the future instances. Unseen examples during the training present a challenge to any inference model.

Figure 7 illustrates the problem. In the toy example, the target population is hidden (only used for testing), but the training data, which is a biased sample from the population, is missing some of the examples with smaller A values (e.g., smaller companies less likely to be sampled). The fitted regression model can still perform well on the training set, but will fail in testing. The wide gap between the training and the testing scores (middle) indicates this failure of model generalization. Now, by accounting for the *unknown unknowns* (e.g., injecting the generated unseen examples), we can improve the model generalizability. Of course, estimating the number, as well as the values of the unseen examples is not straight-forward.

6 Uncertainty in Data Sharing Environments

QUDE is much more challenging when data is shared or made public. Controlling false discoveries, discussed in Section 4, becomes more problematic when data is shared across institutions or research groups given the difficulty of establishing effective MCP control procedures in sharing environments. Even if just one member deviates from the exact testing protocol, uncertainty is immediately introduced into the results. Moreover, when data is made public, avoiding uncertainty as a result of the MCP (whether introduced intentionally or otherwise) becomes almost impossible given the inordinate amount of coordination and oversight required by all parties using the data. However, these issues do not impede the trend by industry and research institutions to make data publicly available making it imperative to create a method for effectively controlling the MCP in data sharing settings.

6.1 An Illustrative Example

Consider a publicly shared dataset such as MIMIC III [29] published by MIT. The dataset contains de-identified health data associated with $\approx 40,000$ critical care patients. The existing solution for controlling the MCP on such a dataset is to make use of a hold-out. In this case, MIT can release 30K patient records as an exploration dataset (EDS) and hold back 10K as a validation dataset (VDS). The EDS can then be used in arbitrary ways to find interesting insights from the data. However, before a result can be used in a publication, the hypothesis is tested for statistical significance over the VDS.

Unfortunately, there are several issues with such a solution. In order to use the VDS more than once, every hypothesis over the VDS has to be tracked and the MCP controlled. Hence, it becomes necessary for the data owner (e.g., MIT) to provide a "certification" service to validate results obtained over the EDS which is both a burden for the data owner as well as a potential source of bias. Researchers need to trust the data owner to correctly apply MCP control procedures and objectively evaluate their hypotheses.

6.2 Automated Result Certification

Ideally, a data owner publishes a dataset and goes offline (i.e., not have to interact with researchers any further) in order to both minimize the overhead imposed on the data owner as well as eliminate potential bias during the validation phase. An automated solution can be constructed using several cryptographic primitives and is based on the following observation: if all the p-values computed for a dataset are accounted for during the analysis phase, in addition to the order in which they were computed, it is possible to apply an incremental control procedure [46, 23] to control for uncertainty. The solution hinges on the use of *Fully Homomorphic Encryption* (FHE) which was first proposed by Gentry in 2009 [25]. FHE enables computation over encrypted data without revealing any information on the underlying data. Using FHE, any arithmetic operation (and thus any function) can be evaluated on ciphertexts using only the public key such that the result of the evaluation remains encrypted and unknown to the evaluator. If the data is encrypted by the data owner using FHE and the encrypted data made public, researchers are still able to use the encrypted data for analysis (e.g., compute statistical tests) and obtain encrypted results without interacting with the data owner. A researcher may then request the entity in possession of the decryption key (e.g., the data owner) to reveal the result computed locally over the encrypted data.

Consider, once more, the case of the MIMIC III dataset. The data owner (e.g., MIT) makes the encrypted version of the dataset publicly available. Researchers use the encrypted data to compute statistical tests and obtain encrypted p-values. The encrypted p-values are then sent to MIT which proceeds to decrypt and reveal each p-value on a publicly readable database. A p-value obtained in such a fashion can be audited by examining the sequence of test records stored in the database and applying an incremental control procedure. While still requiring participation of the data owner, such a solution is one step closer to the desired goal since it no longer requires the owner to verify hypotheses or otherwise interact with researchers.

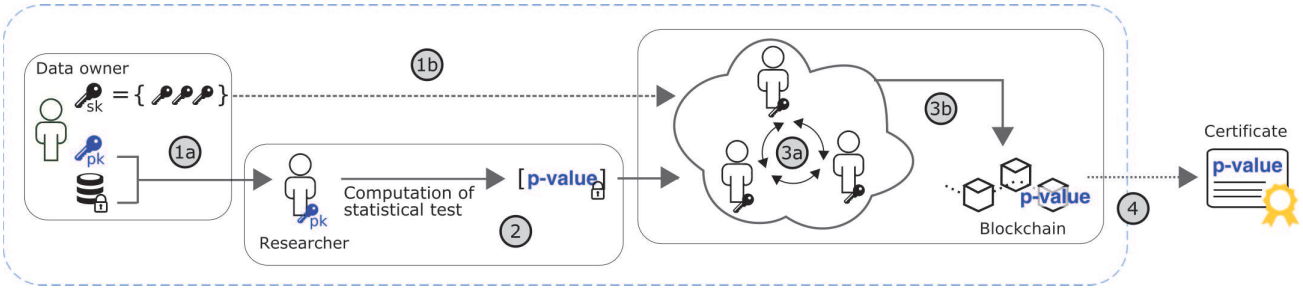

Figure 8: Overview of statistical test certification process: (1a) The data owner encrypts and publishes a dataset along with the public key. (1b) The data owner distributes the secret key to a set of parties that can collectively decrypt using threshold-FHE. (2) A researcher downloads the encrypted data along with the public key and computes a statistical test using FHE. (3a) To obtain the p-value in the clear, the researcher submits the encrypted result to the set of parties who then decrypt the p-value using a consensus protocol. (3b) The parties reveal the result by posting it on a public blockchain along with a timestamp. (4) A publication claiming a significant result can provide proof of valid testing procedures using the records stored on the blockchain.

The responsibilities of the data owner can be distributed to a set of parties (i.e., institutions, research groups, etc.) using threshold-FHE [4] which requires a majority of parties to "agree" on decrypting a result. In combination with a distributed ledger (e.g., a blockchain [34]), which guarantees immutability of recorded transactions, the sequence of tests can be tracked by recording each p-value at the time of decryption. Since the ledger is public and tamper-proof, it can be used as a mathematical proof of result validity. Figure 8 provides a high-level overview of the protocol.

7 Conclusion & Future Work

In this paper, we present several cases for the uncertainty in data analysis and exploration. First, we provide an overview of our work on quantifying the uncertainty as a form of unknown missing data items, undetected data errors in the data set. Next, we point out that any data-driven discoveries should be taken with care, because data analysis and exploration generally requires testing numerous hypotheses, increasing the chance of false discovery. Data analysis in a data sharing environment further complicates this issue of MCP. Finally, we discuss a couple model quality problems that can serve as a source of uncertainty in data analysis.

Our overarching goal is to quantify all types of uncertainty in data analysis and exploration, and in turn, provide measures to correct and validate the analysis results and discoveries. To this end, we plan on implementing the proposed solutions for ML model quality (Section 5) and data-sharing environments (Section 6). We have integrated some of the proposed solutions into an interactive human-in-the-loop data exploration and model building suite [6]. We are also interested in investigating other types of uncertainty (e.g., learning the right feature set for the task and the feature quality assurance) and how they interact with each other. It is interesting to understand the relationship among the different types of uncertainty as the uncertainty compounds over the pipeline.

References

[1] Lessons from next-generation data wrangling tools. https://www.oreilly.com/ideas/lessons-from-next-generation-data-wrangling-tools, 2015.

[2] The 6 components of open-source data science/ machine learning ecosystem; did python declare victory over r? https://www.kdnuggets.com/2018/06/ecosystem-data-science-python-victory.html, 2018.

[3] Z. Abedjan, X. Chu, D. Deng, R. C. Fernandez, I. F. Ilyas, M. Ouzzani, P. Papotti, M. Stonebraker, and N. Tang. Detecting data errors: Where are we and what needs to be done? *PVLDB*, 9(12):993–1004, 2016.

[4] G. Asharov, A. Jain, A. López-Alt, E. Tromer, V. Vaikuntanathan, and D. Wichs. Multiparty computation with low communication, computation and interaction via threshold fhe. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 483–501. Springer, 2012.

[5] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B (Methodological)*, 57(1):289–300, 1995.

[6] C. Binnig, B. Buratti, Y. Chung, C. Cousins, T. Kraska, Z. Shang, E. Upfal, R. C. Zeleznik, and E. Zgraggen. Towards interactive curation & automatic tuning of ml pipelines. In *DEEM@ SIGMOD*, pages 1–1, 2018.

[7] C. Binnig, L. De Stefani, T. Kraska, E. Upfal, E. Zgraggen, and Z. Zhao. Toward sustainable insights, or why polygamy is bad for you. In *CIDR*, 2017.

[8] P. Bohannon, W. Fan, M. Flaster, and R. Rastogi. A cost-based model and effective heuristic for repairing constraints by value modification. In *Proceedings of SIGMOD*, pages 143–154, 2005.

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

[10] F. Chirigati, H. Doraiswamy, T. Damoulas, and J. Freire. Data polygamy: The many-many relationships among urban spatio-temporal data sets. In *SIGMOD*, pages 1011–1025, 2016.

[11] J. Chomicki and J. Marcinkowski. Minimal-change integrity maintenance using tuple deletions. *Information and Computation*, 197(1-2):90–121, 2005.

[12] X. Chu, I. F. Ilyas, S. Krishnan, and J. Wang. Data cleaning: Overview and emerging challenges. In *Proceedings of SIGMOD*, pages 2201–2206, 2016.

[13] Y. Chung, T. Kraska, S. E. Whang, and N. Polyzotis. Slice finder: Automated data slicing for model interpretability. *arXiv preprint arXiv:1807.06068*, 2018.

[14] Y. Chung, S. Krishnan, and T. Kraska. A data quality metric (dqm): how to estimate the number of undetected errors in data sets. *Proceedings of the VLDB Endowment*, 10(10):1094–1105, 2017.

[15] Y. Chung, M. L. Mortensen, C. Binnig, and T. Kraska. Estimating the impact of unknown unknowns on aggregate query results. In *Proceedings of ACM SIGMOD*, pages 861–876, 2016.

[16] Y. Chung, M. L. Mortensen, C. Binnig, and T. Kraska. Estimating the impact of unknown unknowns on aggregate query results. *TODS*, 43, 2018.

[17] G. Cong, W. Fan, F. Geerts, X. Jia, and S. Ma. Improving data quality: Consistency and accuracy. In *PVLDB*, pages 315–326, 2007.

[18] A. Crotty, A. Galakatos, E. Zgraggen, C. Binnig, and T. Kraska. Vizdom: interactive analytics through pen and touch. *Proceedings of the VLDB Endowment*, 8(12):2024–2027, 2015.

[19] Ç. Demiralp, P. J. Haas, S. Parthasarathy, and T. Pedapati. Foresight: Rapid data exploration through guideposts. *arXiv preprint arXiv:1709.10513*, 2017.

[20] X. L. Dong and D. Srivastava. Big data integration. In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*, pages 1245–1248. IEEE, 2013.

[21] O. J. Dunn. Multiple comparisons among means. *Journal of the American Statistical Association*, 56(293):52–64, 1961.

[22] H. Ehsan, M. A. Sharaf, and P. K. Chrysanthis. Muve: Efficient multi-objective view recommendation for visual data exploration. In *ICDE*, pages 731–742, May 2016.

[23] D. P. Foster and R. A. Stine. α-investing: a procedure for sequential control of expected false discoveries. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(2):429–444, 2008.

[24] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. Crowddb: Answering queries with crowdsourcing. In *Proceedings of ACM SIGMOD*, pages 61–72, 2011.

[25] C. Gentry. *A fully homomorphic encryption scheme*. Stanford University, 2009.

[26] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[27] M. L. Head, L. Holman, R. Lanfear, A. T. Kahn, and M. D. Jennions. The extent and consequences of p-hacking in science. *PLoS biology*, 13(3):e1002106, 2015.

[28] J. P. Ioannidis. Why most published research findings are false. *PLoS medicine*, 2(8):e124, 2005.

[29] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

[30] M. Kahng, D. Fang, and D. H. P. Chau. Visual exploration of machine learning results using data cube analysis. In *HILDA*, page 1. ACM, 2016.

[31] R. Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[32] A. Lopatenko and L. Bravo. Efficient approximation algorithms for repairing inconsistent databases. In *Proceedings of ICDE*, pages 216–225, 2007.

[33] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, et al. Ad click prediction: a view from the trenches. In *KDD*, pages 1222–1230, 2013.

[34] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. http://www.bitcoin.org/bitcoin.pdf.

[35] N. M. Nasrabadi. Pattern recognition and machine learning. *Journal of electronic imaging*, 16(4):049901, 2007.

[36] 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.

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

[38] M. Stonebraker, D. Bruckner, I. F. Ilyas, G. Beskales, M. Cherniack, S. B. Zdonik, A. Pagan, and S. Xu. Data curation at scale: The data tamer system. In *CIDR*, 2013.

[39] B. Trushkowsky, T. Kraska, M. J. Franklin, P. Sarkar, and V. Ramachandran. Crowdsourcing enumeration queries: Estimators and interfaces. In *to appear at IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2015.

[40] B. Trushkowsky, T. Kraska, and P. Sarkar. Answering enumeration queries with the crowd. *Commun. ACM*, 59(1):118–127, 2016.

[41] M. Vartak, S. Madden, A. G. Parameswaran, and N. Polyzotis. SEEDB: automatically generating query visualizations. *PVLDB*, 7(13):1581–1584, 2014.

[42] J. Wang, T. Kraska, M. J. Franklin, and J. Feng. Crowder: Crowdsourcing entity resolution. *PVLDB*, 5(11):1483–1494, 2012.

[43] J. Wang, S. Krishnan, M. J. Franklin, K. Goldberg, T. Kraska, and T. Milo. A sample-and-clean framework for fast and accurate query processing on dirty data. In *Proceedings of SIGMOD*, pages 469–480, 2014.

[44] M. Yakout, A. K. Elmagarmid, J. Neville, M. Ouzzani, and I. F. Ilyas. Guided data repair. *PVLDB*, 4(5):279–289, 2011.

[45] E. Zgraggen, Z. Zhao, R. Zeleznik, and T. Kraska. Investigating the effect of the multiple comparisons problem in visual analysis. In *CHI*, page 479. ACM, 2018.

[46] Z. Zhao, L. De Stefani, E. Zgraggen, C. Binnig, E. Upfal, and T. Kraska. Controlling false discoveries during interactive data exploration. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 527–540. ACM, 2017.

[47] Z. Zhao, E. Zgraggen, L. De Stefani, C. Binnig, E. Upfal, and T. Kraska. Safe visual data exploration. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1671–1674. ACM, 2017.