On Explaining Confounding Bias

Brit Youngmann
CSAIL MIT
brity@mit.edu

Michael Cafarella
CSAIL MIT
michjc@csail.mit.edu

Babak Salimi
University of California, San Diego
bsalimi@ucsd.edu

Yuval Moskovitch
Ben Gurion University of the Negev
yuvalmos@bgu.ac.il

ABSTRACT

When analyzing large datasets, analysts are often interested in the explanations for surprising or unexpected results produced by their queries. In this work, we focus on aggregate SQL queries that expose correlations in the data. A major challenge that hinders the interpretation of such queries is *confounding bias*, which can lead to an unexpected correlation. We generate explanations in terms of a set of *confounding variables* that explain the unexpected correlation observed in a query. We propose to mine candidate confounding variables from external sources since, in many real-life scenarios, the explanations are not solely contained in the input data. We present an efficient algorithm that finds the optimal subset of attributes (mined from external sources and the input dataset) that explain the unexpected correlation. This algorithm is embodied in a system called MESA. We demonstrate experimentally over multiple real-life datasets and through a user study that our approach generates insightful explanations, outperforming existing methods that search for explanations only in the input data. We further demonstrate the robustness of our system to missing data and the ability of MESA to handle input datasets containing millions of tuples and an extensive search space of candidate confounding attributes.

1 INTRODUCTION

When analyzing large datasets, analysts often query their data to extract insights. Oftentimes, there is a need to elaborate upon the queries' answers with additional information to assist analysts in understanding unexpected results, especially for aggregate queries, which are harder to interpret [42, 63]. While aggregate query results expose correlations in the data, the human mind cannot avoid a causal interpretation. Thus, we provide explanations for unexpected correlations observed in aggregate queries using causation terms.

In this work, we focus on SQL queries that are aggregating an *outcome attribute* (O) based on some groups of interest indicated by a grouping attribute, referred to as the *exposure* (T) [56]. A major challenge that hinders the interpretation of such queries is *confounding bias* [58] that can lead to a spurious association between T and O and hence perplexing conclusions. Confounding bias occurs when an analyst tries to determine the effect of an exposure on an outcome but unintentionally measures the effect of another factor(s) (i.e., a *confounding variable(s)*) on the outcome. This results in a distortion of the actual association between T and O [56]. We are interested in generating explanations in terms of a set of confounding variables that explain unexpected correlations observed in query results.

Previous work detected uncontrolled confounding variables from the data [63]. However, in many cases, such variables might be found outside the narrow query results that and the database being used [39]. Thus, there is a need to develop automated solutions that can explain unexpected correlations observed in query results to analysts, which goes beyond just the data accessed by the query. To illustrate, consider the following example.

Example 1.1. *Ann is a data analyst in the WHO organization who aims to understand the coronavirus pandemic for improved policy-making. She examines a dataset containing information describing Covid-19-related facts in multiple cities worldwide. It consists of the number of deaths-/recovered-/active-/new- per-100-cases in each city. Ann evaluates the following query over this dataset:*

```
SELECT Country, avg(Deaths_per_100_cases)
FROM Covid-Data
GROUP BY Country
```

A visualization of the query results is given in Figure 1. Here, the exposure is Country *and the outcome is* Deaths_per_100_cases*. Ann observes a puzzling correlation between the exposure and outcome; namely, she wonders why the choice of the country has such a substantial effect on the death rate. She is interested in finding a set of confounding variables that explain this association. She sees that the attribute* Confirmed_cases *from* Covid-data *is correlated with* Deaths_per_100_cases*. However, this attribute alone is not enough to explain the correlation. For example, she sees that while Germany had the fifth-most confirmed cases worldwide, it had only a fraction of the death toll in other countries. Ann understands that other factors (that are not in this data) affect the association between death rate and country. She remembers reading in the news that as a country's success (defined by multiple variables, including GDP[1] and HDI[2]) grows, the death rate decreases [36, 69]. However, such economic features of countries are not available in her dataset but could be extracted from external sources.*

We propose to mine unobserved confounding attributes from external sources. In general, our framework can extract candidate confounders from any knowledge source (e.g., related tables, data lakes, web tables) as long as it can be integrated with the input data. This paper focuses on mining attributes from a Knowledge Graph (KG) for the following reasons. KGs are an emerging type of knowledge representation [13, 25, 75], that can effectively organize and represent a large amount of data. KGs have been efficiently

[1] Gross domestic product (GDP) is the monetary value of all goods and services made within a country during a specific period.
[2] The Human Development Index (HDI) is a statistic composite index of life expectancy, education, and per capita income indicators.

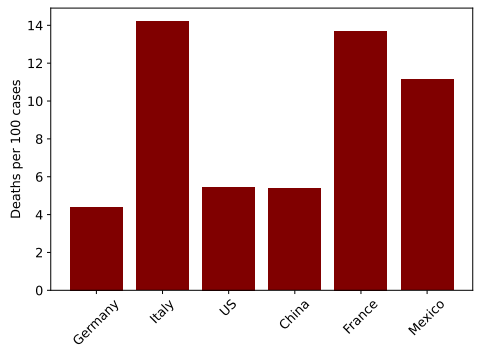


Figure 1: Visualization of the results of the query Q.

utilized in various tasks, such as question-answering and recommendation [22]. Further, attribute names in KGs are typically highly informative, allowing analysts to reason about the generated explanations. However, the sheer breadth of coverage that makes KGs potentially valuable also creates the need to automate the process of mining relevant confounding variables. There are multiple general-purpose (e.g., Wikidata [11], DBpedia [8], Yago [60]) or domain-specific (e.g., for medical proteomics [65], or protein discovery [51]) KGs that act as central storage for data collected from multiple sources. We argue that such valuable data could be utilized for explaining unexpected correlations observed in user queries in a wide range of scenarios.

To this end, we present an efficient algorithm that finds a subset of confounding attributes (mined from external sources and the input dataset) that explain the unexpected correlation observed in a given query. This algorithm is embodied in a system called MESA, which automatically mines candidate attributes from a knowledge source. This source may be provided by the analyst (for a specific domain) or could be any publicly available knowledge source.

Example 1.2. *Ann uses MESA to search for an explanation for her query. MESA mines all available attributes about countries that appear in her data from DBpedia. She learns that besides* Confirmed_cases*, the attributes HDI, and GDP are uncontrolled confounding attributes. She sees that the death rate is similar in countries with a similar number of confirmed cases, HDI, and GDP. She is pleased because she found a plausible real-world explanation for her query results [36, 69].*

Previous work provides explanations for trends and anomalies in query results in terms of predicates on attributes that are shared by one (group of) tuple in the results but not by another (group of) tuple [30, 39, 61, 62, 70]. However, those methods do not account for correlations among attributes, and are thus inapplicable for explaining the correlation between the outcome and the exposure. [63] presented a system that provides explanations based on causal analysis, measured by correlation among attributes. However, this system only considers the input dataset, and its running times are exponential in the number of candidate confounding attributes. We share with CajaDE [39] the motivation of considering explanations that are not solely drawn from the input table. CajaDE is a system that generates insightful explanations based on contextual information mined from tables related to the table accessed by the query. Their explanations are a set of patterns that are unevenly distributed among groups in the query results, and are *independent of the outcome attribute*. Thus, CajaDE may generate explanations that are irrelevant for understanding the correlation between the exposure and outcome.

Our framework supports a rich class of aggregate SQL queries that compare among subgroups, investigating the relationship between an aggregated attribute O and a grouping attribute T. To explain the correlation between T and O observed in the results of a query Q, we formalize the Correlation-Explanation problem that seeks a set of confounding attributes (extracted from external sources or the input database), which minimizes the partial correlation between T and O (to measure the correlation between T and O, while controlling for the effect of confounding variables). Further, MESA enables analysts to learn the individual responsibility of selected attributes and to automatically identify unexplained data subgroups (correspond to refinements of Q) in which different explanations are required.

Given an input database $\mathcal{D}$ and a knowledge source, we extract a set of attributes representing additional properties of entities from $\mathcal{D}$. The attributes are extracted only after the query arrives (as the knowledge source may be a part of the input). Extracted attributes may contain many missing values, especially ones extracted from a KG where data is sparse. Previous work showed that common approaches for handling missing data could cause substantial *selection bias* [66] (which occurs when the obtained data fails to properly represent the population intended to be analyzed) if many values are missing [66]. In contrast to prediction, explanations quality is more sensitive to missing data [52]. We, therefore, present a principled way of handling missing values, ensuring the explanations are robust to missing data. We provide sufficient conditions to detect selection bias and an algorithmic approach to handle it properly.

There are potentially hundreds of attributes that could be extracted from external sources. Thus, there is a need to develop an efficient algorithm to search for the optimal attribute set (i.e., explanation) in this extensive search space. Further, the search for the optimal attribute set involves estimating partial correlation for high-dimensional conditioning sets, which is notoriously difficult [40]. To this end, we propose the MCIMR algorithm, which does not require iterating over all possible attribute sets, and avoids estimating high-dimensional conditioning sets. It selects attributes based on Min-Conditional-mutual-Information (a common measure for partial correlation) and Min-Redundancy criteria, yielding a PTIME algorithm that finds the optimal k-size explanation where k is given. We then define a stopping criterion, allowing the algorithm to stop when no further improvement is found. We propose multiple pruning techniques to speed up the computation.

We conduct an experimental study based on four commonly used datasets that evaluate the quality and efficiency of the MCIMR algorithm. Our approach is effective whenever the explanation can be found in a given knowledge source. We show that this was the case in 72.5% of random aggregate queries evaluated on these datasets, setting the knowledge source to be the DBPedia KG [8]. To evaluate the explanations quality, we focus on 14 representative queries suffering from confounding bias. These queries are inspired by real-life analysis reports, such as Stack Overflow annual reports [5] and academic papers [69]. We ran a user study consisting of 150 subjects to evaluate the quality of our explanations compared with six approaches. We show that the explanations generated by MCIMR

are almost as good as those of a computationally infeasible naïve method that iterates over all attribute subsets and are much better than those of feasible competitors. We also show that previous findings in each domain support our substantive explanations. Our experimental results demonstrate the robustness of our solution to missing data and indicate the effectiveness of our algorithm in finding explanations in less than 10s for queries evaluated on datasets containing more than 5M tuples.

Our main contributions are summarized as follows:

- We formalize the Correlation-Explanation problem that seeks a subset of attributes that explains unexpected correlations observed in SQL queries (Section 2).
- We propose to extract unobserved confounding attributes from external sources and focus on KGs. We develop a principled way to avoid selection bias (Section 3).
- We devise an efficient algorithm that computes the optimal explanation for the Correlation-Explanation problem. We embody this algorithm in a system called MESA which enables analysts to automatically identify unexplained data subgroups (Section 4).
- We qualitatively evaluated the explanations produced by MESA and existing solutions over real-life datasets through a user study. We further conducted performance experiments to assess scalability (Section 5).

Related work is presented in Section 6 and we conclude in Section 7.

2 MODEL AND PROBLEM FORMULATION

2.1 Data Model

We operate on a standard multi-relational dataset $\mathcal{D}$. To simplify the exposition, we assume $\mathcal{D}$ consists of a single relational table, however, our definitions and results apply to the general case. The table's attributes are denoted by $\mathcal{A}$. For an attribute A_i we denote its domain by $Dom(A_i)$. We use bold letters for sets of attributes $\boldsymbol{A}\subseteq\mathcal{A}$. We expect the reader is familiar with basic information theory measures, such as entropy and conditional mutual information.

Our framework supports a rich class of SQL queries that involve groping, joins and different aggregations to support complex real-world scenarios. The queries we examine compare among subgroups, investigating the relationship between an aggregated attribute O (referred to as the outcome) and an grouping attribute T (referred to as the exposure). To simplify the exposition, we assume a single grouping attribute. However, our results can be naturally generalized for multiple grouping attributes. To handle a numerical exposure, one may bin this attribute. We call the condition C (given by the WHERE clause) the context for the query. Given C, we aim to explain the difference among $agg(O)$ for each $T{=}t_i$, where $t_i{\in}Dom(T)$. If the attribute O belongs to a different table from the one containing the exposure T, the query Q describes how these two tables are combined in the join condition.

We use the following example based on the Stack Overflow (SO) dataset throughout this paper. In our experiments, we demonstrate the operation of MESA over four datasets, including Covid-Data.

Example 2.1. *SO dataset contains information about people who code around the world, such as their age, gender, income, and country. Consider the following query:*

```
SELECT Country, avg(Salary)
FROM SO
WHERE Continent = Europe
GROUP BY Country
```

Here, O is Salary, T is Country, the context C is Continent = Europe, and the aggregation function is average. We aim to explain the difference in the average salary of developers from each country in Europe. While some attributes from the dataset may partially explain this (e.g., Gender, DevType), other important attributes that can cast light on this difference cannot be found in this dataset.

Knowledge Extraction. In general, MESA can extract attributes from any external source, such as related tables, data lakes, unstructured data (e.g., images), or Knowledge Graphs (KGs), as long as it can be integrated with the input dataset. This paper focuses on mining attributes from a KG for the following reasons. First, KGs can effectively organize a large amount of (domain specific or general) data, and have been successfully utilized in various downstream applications, such as question-answering systems, search engines, and recommendation systems [22]. Second, one of the strengths of KGs is that most of the attributes are already reconciled. Namely, we will not have to match different versions of attributes across different entities. Last, the attribute names are typically highly informative, allowing users to reason about the generated explanations. We note that extracting attributes from other sources poses a series of additional challenges, including handling many-to-many relations and uninformative attribute names. We leave these extensions for future research.

Attributes extracted from a knowledge source may be irrelevant for a given query. We thus let the analyst decide which source MESA should use. Given a knowledge source (e.g., domain specific KG [51, 65], publicly available KG [8, 11, 60], data lake), we extract a set of attributes $\mathcal{E}$ representing additional properties of entities from $\mathcal{D}$.

Continuing with our example, $\mathcal{E}$ could be a set of properties of countries extracted from a KG, such as their density, and HDI. We can potentially join $\mathcal{E}$ and $\mathcal{T}$, by linking values from $\mathcal{T}$ with their corresponding entities in $\mathcal{G}$ that were used for attributes extraction. However, $\mathcal{E}$ may contain many attributes, most of them are irrelevant for explaining the query results.

2.2 Problem Formulation

Given a query, the analyst observes an unexpected correlation between the exposure T and the outcome O attributes that she would like to explain. We assume there is *confounding bias* that causes a spurious association between T and O. Confounding bias is a systematic error due to the uneven or unbalanced distribution of a third variable(s), known as the confounding variable(s) in the competing groups. Uncontrolled confounding variables lead to an inaccurate estimate of the true association between T and O. Our goal is to discover the confounding variables. Let $\mathcal{A}$ denote $\mathcal{E}\cup\mathcal{T}\setminus\{O,T\}$, referred to as the candidate attributes. $\mathcal{A}$ contains confounding attributes that affect both T and O. We aim to find an attribute set $\boldsymbol{E}\subseteq\mathcal{A}$ that control the correlation between O and T, i.e., when

conditioning on $\boldsymbol{E}$, the correlation between O and T is diminished. We call such a set the correlation explanation.

Example 2.2. *It is very likely that countries' economic features (such as GDP, Gini, and HDI) affect developers' salaries. To unearth the association between* Country *and* Salary*, one must measure the correlation while controlling for such attributes. This will allow users to understand which factors affect the differences in developers' salaries. Intuitively, we expect the average developers' salaries to be similar in countries with similar economic characteristics.*

Ideally, we look for a minimal-size set of attributes $\boldsymbol{E}\subseteq\mathcal{A}$ s.t: $(O\perp\!\!\!\perp T|\boldsymbol{E},C)$. However, in practice, we may not find such perfect explanations (that entirely explains away the correlation), hence we search for a minimal-size set of attributes that *minimizes the partial correlation between T and O*. Partial correlation measures the strength of a relationship between two variables, while controlling for the effect of other variables. A common measure of partial correlation is multiple linear regression, which is sensitive only to linear relationship. Other partial correlation measures, such as Spearman's coefficient, are more sensitive to nonlinear relationships [24, 31]. Here we use *Conditional Mutual Information* (CMI), a common measure of the mutual dependence between two variables, given the value of a third. We chose CMI because (1) it is a widely used non-parametric measure for partial correlation [20], (2) there is a plethora of techniques for estimating it from data [63], (3) it also allows us to develop information-theoretic optimizations. CMI may suffer from underestimation, especially when quantifying dependencies among variables with high associations [74]. However, we avoid such cases since, as we explain in Section 4.2, we discard all attributes that are logically dependent on T or O. Note that $(O\perp\!\!\!\perp T|\boldsymbol{E},C)$ holds iff $I(O;T|\boldsymbol{E},C)$=0, where $I(O;T|\boldsymbol{E},C)$ is the mutual information of O and T while conditioning on $\boldsymbol{E}$ and the context C. Thus, we formalize the Correlation-Explanation problem as follows:

Definition 2.1 (**Correlation-Explanation**). *Given a set of candidate attributes $\mathcal{A}$ and a query Q, find a set of attributes $\boldsymbol{E}^*$ s.t: $\boldsymbol{E}^* = argmin_{\boldsymbol{E}\subseteq\mathcal{A}} I(O;T|\boldsymbol{E},C)\cdot|\boldsymbol{E}|$.*

Following previous work [43, 59, 62], besides the explanatory power, we also consider the cardinality of the sets.

Example 2.3. *Among other attributes, we extracted from a KG the* Gini *(E_1), Density (E_2), and HDI (E_3) attributes. An attribute from SO is the developers* Gender *(E_4). According to our data, we have $I(O;T|C)$=2.6. When conditioning on E_1, we get: $I(O;T|C,E_1)$=1.3. Namely, in countries with a similar Gini index, there is less correlation between the country of developers and their salaries. When also considering Density, we get: $I(O;T|C,E_1,E_2)$=0.03. Thus, this set of attributes explains away the correlation in Q_{so}. When conditioning on HDI, on the other hand, we get: $I(O;T|C,E_3)$=2.5. Since the HDI of all countries in Europe is similar[3], this attribute does not explain the observed correlation. Similarly, when conditioning on Gender we get: $I(O;T|C,E_4)$=2.3, implying that the developers gender cannot explain the correlation in Q_{so}.*

To assist analysts in interpreting the results, we enable them to learn the individual *responsibility* of selected attributes. Given an explanation $\boldsymbol{E}$, we rank the attributes in $\boldsymbol{E}$ in terms of their responsibilities as follows:

Definition 2.2 (Degree of responsibility). *Given a query Q and set of attributes $\boldsymbol{E}$, the degree of responsibility of an attribute $E_i\in\boldsymbol{E}$ is defined as follows:*

$$Resp(E_i) := \frac{I(O;T|\boldsymbol{E}\backslash\{E_i\},C) - I(O;T|\boldsymbol{E},C)}{\sum_{E_j\in\boldsymbol{E}}(I(O;T|\boldsymbol{E}\backslash\{E_j\},C) - I(O;T|\boldsymbol{E},C))}$$

The responsibility of an attribute E_i is the normalized value of its individual contribution. When all attributes in $\boldsymbol{E}$ contribute to the explanations (i.e., the numerator is positive), the denominator is non-negative. The responsibility of E_i is positive if E_i contributes to the explanation. Thus, a negative responsibility indicates that E_i only harms the explanation (it happens since E_i has a negative interaction information with O and T). The higher the responsibility of an attribute, the greater is its individual explanation power.

Example 2.4. *Recall that E_1= Gini, and E_2= Density. Let $\boldsymbol{E}$=$\{E_1,E_2\}$. According to our data we have: $I(O;T|C,E_2)$=1.51. We get: $Resp(E_1)$=0.54, and $Resp(E_2)$=0.46. The attribute Hobby (E_5) indicates whether a developer is coding as an hobby. It has a negative interaction information with O and T. We have $I(O;T|C,E_5)$=2.7 >$I(O;T|C)$. Let $\boldsymbol{E}$=$\{E_1,E_5\}$. We get: $I(O;T|C,\boldsymbol{E})$=1.5, $Resp(E_1)$=1.2, and $Resp(E_5)$= − 0.2. Since E_5 did not contribute to the explanation, its responsibility is negative.*

Key Assumption. We generally believe that attributes with low responsibility are of little interest to analysts and that XOR-like explanations (in which the explanation power of each individual attribute is low, but their combination makes a good explanation) are hard to understand; thus, they are less likely to be considered good explanations. Our view is motivated by [44]. A similar assumption is often made in feature selection [19, 68], where they assume the optimal feature set does not contain multivariate associations among features, which are individually irrelevant to a target class but become relevant in the presence of others. We further believe true XOR phenomena are likely to be uncommon in real datasets; the practical success of feature selection methods that make this assumption [20] is some evidence for this view. Further, generating XOR explanations would be a substantial additional technical challenge. It would eliminate our ability to prune low-relevance attributes and to define a stopping criterion for our algorithm (see Section 4). Also, extending our algorithm to consider XOR explanations would mean estimating CMI for a high-dimensional conditioning set, which is notoriously difficult [40].

3 ATTRIBUTES EXTRACTION

3.1 Extracting the Candidate Attributes

MESA extracts attributes representing additional properties of entities from $\mathcal{D}$ from a given knowledge source. In general, we may extract attributes from any given source as long as it can be integrated with the input dataset. For example, we may extract attributes from a data lake, leveraging existing methods to join or union an input table with other tables [31, 53, 64, 72, 76]. As mentioned in Section 2.1, here we focus on extracting attributes from a given KG.

Extracting Attributes from a KG: Given a KG, the first step is to map values that appear in the table $\mathcal{T}$ to their corresponding

[3] As reflected in https://en.populationdata.net/rankings/hdi/europe/.

unique entities in the KG $\mathcal{G}$. This task is often referred to as the Named Entity Disambiguation (NED) problem [54]. We can use any off-the-shelf NED algorithm (e.g., [54, 78]) to match any non-numerical value in $\mathcal{T}$ to an entity in $\mathcal{G}$. Next, given an entity from $\mathcal{T}$, we extract all of its properties from $\mathcal{G}$. We then organize all the extracted properties into a table, setting a null value to all properties whose values were missing. This process is equivalent to building the *universal relation* [32] out of all of the entity specific relations that were derived from $\mathcal{G}$.

To extract more attributes and potentially improve the explanations, one may "follow" links in $\mathcal{G}$. Namely, extract also properties of values which are entities in $\mathcal{G}$ as well. This process can be done up to any number of hops in $\mathcal{G}$. All properties are then flattened and stored as a single table.

Accommodating One-to-Many Relations: The process described above assumes that each entity is associated with a single value. However, real-world data often contain multiple categorical values (see Example 3.1). Because correlation is only defined for sets of paired values, downstream applications typically aggregate the values into a single number [64]. MESA supports any user-defined function (e.g., mean, sum, max, first, or any representation-learning-based technique [16]) to perform the aggregation.

Example 3.1. *A country's leader is an attribute extracted for each country. We can extract properties of the leaders, such as their age and gender, adding to $\mathcal{E}$ additional properties such as Leader Age, and Leader Gender. Other properties may point to multiple entities. The US entity has the property Ethnic-Group, which points to different ethnic groups. Each ethnic group is also an entity, and has the property Population size. One may add the property Avg Population size of Ethnic-Group to $\mathcal{E}$ by averaging the population sizes.*

3.2 Handling Missing Data

Extracted attributes, especially ones from KGs where data is sparse, may contain missing values. Our goal is to develop a principled approach to ensure the generated explanations are robust to missing data. Handling missing data is an enduring problem for many systems [28]. The simplest approach to dealing with missing values is to restrict the analysis to complete cases, i.e., discard cases that have missing values.However, this can induce *selection bias* if the excluded tuples are systematically different from those included. For example, if the HDI values of only countries with a very high HDI are missing, restricting the analysis only to complete cases may lead to misleading explanations. A common solution is to impute missing values. Data imputation is unlikely to cause substantial bias if few data are missing, but bias may increase as the number of missing data increases [66]. Another common approach is Multiple Imputations (MI) [55]. While MI is useful in supervised learning as long as it leads to models with an acceptable level of accuracy, MI makes a missing-at-random assumption [28], which is often not the case in our setting. The approach that we followed is Inverse Probability Weighting (IPW), a commonly used method to correct selection bias [66]. In IPW, we consider only complete cases, but more weight is given to some complete cases than others. We next explain how to adapt IPW into our setting.

For simplicity of presentation, we assume that $\mathcal{T}$ and $\mathcal{E}$ have been joined into a single table. As we will explain in Section 4, for an attribute $E \in \mathcal{E}$ we estimate $I(O;T|E,C)$ and $I(E;E')$ for $E' \in \mathcal{E}$. Therefore, we need to recover the probabilities $P(O|C,E)$, $P(O|C,T,E)$, $P(E)$, and $P(E|E')$. But since E may contain missing values, we must ensure that those probabilities are *recoverable*. Given an attribute E, let R_E denote a selection attribute that indicates if the values of E for the i-th tuple in the results of Q is missing. I.e., $R_E[i]$=1 if the value of E for the i-th tuple was extracted, and $R_E[i]$=0 otherwise. A complete cases analysis means that we examine only cases in which $R_E[i]$=1. Let R_E=1 denote the selection of all tuples in which for them $R_E[i]$=1 holds. We say the probability of an event X which involves E (e.g., $P(O|E)$) is recoverable if: $P(X)$=$P(X|R_E$=1).

We prove that $I(O;T|C,E)$ is recoverable if the complete cases are a representative sample of the original data, and each complete case is a random sample from the population of individuals with the same E and T values.

Proposition 3.1. *If $(O \perp\!\!\!\perp R_E = 1|E,C)$ and $(O \perp\!\!\!\perp R_E = 1|E,T,C)$, then $I(O;T|C,E)$ is recoverable.*

We prove $I(E;E')$ is recoverable if the completeness of a case is independent of E, and remains independent given E'.

Proposition 3.2. *If $(E_i \perp\!\!\!\perp R_{E_i}$=1, R_{E_j}=1) and $(E_i \perp\!\!\!\perp R_{E_i}$=1, R_{E_j}=1$|E_j)$, then $I(E;E')$ is recoverable.*

In situations other than described above, the probabilities will generally not be recoverable. Following the IPW approach, we assign weights to complete cases, where the weight $W(X)$ of an event X is defined as $W(X)$= $P(R_E$=1$)/P(R_E$=1$|X)$. However, since E contains missing values, $P(X)$ is unknown. We thus estimate $P(X)$. Commonly, a logistic regression model is fitted [35, 37]. Data available for this are the values of the attributes in $\mathcal{D}$. We therefore employ a logistic regression (at pre-processing) to estimate $P(X)$. We note that although, as in MI, we predict missing values, we only use those predicted values for weights computation and not for the entire analysis.

4 ALGORITHMS

4.1 The MCIMR Algorithm

We present the MCIMR algorithm for the Correlation-Explanation problem. We show that MCIMR is a PTIME algorithm that finds the optimal k-size solution where k is given. We then define a stopping criterion, allowing it to stop when no further improvement is found.

When k equals 1, the optimal solution to Correlation-Explanation is the attribute $E \in \mathcal{A}$ that minimizes $I(O;T|C,E)$. When $k \geq 1$, a simple incremental solution is to add one attribute at a time: Given the explanation obtained at the $(k-1)$-th iteration $\boldsymbol{E_{k-1}}$, the k-th attribute to be added, denoted as E_k, is the one that contributes to the largest decrease of $I(O;T|C,\boldsymbol{E_{k-1}})$. Formally,

$$E_k = argmin_{E \in \mathcal{A} \setminus \boldsymbol{E_{k-1}}} I(O;T|C,\boldsymbol{E_k}) \quad (1)$$

where $\boldsymbol{E_k}$=$\boldsymbol{E_{k-1}} \cup \{E_k\}$.

It is difficult to get an accurate estimation for multivariate mutual information [57], as in Equation (1). Instead, MCIMR calculates only bivariate probabilities, which is much more accurate, by incrementally selecting attributes based on Minimal-Conditional-mutual-Information (MCI) and Minimal-Redundancy (MR) criteria.

The idea behind MCI is to search a k-size attribute set $\boldsymbol{E_k}$ that satisfies Equation 2, which approximates Equation 1 with the mean

Algorithm 1: The MCIMR Algorithm.

```
input : A number k, a set of attributes 𝒜, the outcome, treatment attributes O and T,
        and the context C
output: An explanation E.
1  MCIMR(k, 𝒜, O, T, C):
2  E ← ∅.
3  for i ∈ [1, k] do
4      E_i ← NextBestAtt (O, T, C, E, 𝒜)
5      if O ⫫ E_i | E                    // The responsibility test for E_i
6      then
7          return E
8      E ← E ⋃ {E_i}
9  return E
10 NextBestAtt (O, T, C, E, 𝒜):
11 E* ←None, v ← ∞
12 foreach E ∈ 𝒜 \ E do
13     /* Weights are added if selection bias was detected      */
14     v_1 ← I(O; T|C, E), v_2 ← 0                  // Min CI computation
15     foreach E' ∈ E do
16         /* Weights are added if selection bias was detected  */
17         v_2 ← v_2 + I(E; E')              // Min redundancy computation
18     if v_1 + v_2/|E| < v then
19         E* ← E, v ← v_1 + v_2/|E|
20 return E*
```

value of all CMI values between the individual attributes in $\boldsymbol{E_k}$ and O and T:

$$\boldsymbol{E_k} = argmin_{\boldsymbol{E_k} \subseteq \mathcal{A}} CI(O, T, C, \boldsymbol{E_k}) \quad (2)$$

where $CI(O, T, C, \boldsymbol{E_k}) = \frac{1}{k} \sum_{E \in \boldsymbol{E_k}} I(O; T|C, E)$.

However, it is likely that attributes selected according to MCI are redundant. Thus, the following minimal redundancy condition is added:

$$\boldsymbol{E_k} = argmin_{\boldsymbol{E_k} \subseteq \mathcal{A}} Rd(\boldsymbol{E_k}) \quad (3)$$

where $Rd(\boldsymbol{E_k}) = \frac{1}{k^2} \sum_{E_i, E_j \in \boldsymbol{E_k}} I(E_i; E_j)$.

Our goal is to minimize CI and Rd simultaneously. Namely, we look for a k-size attribute set $\boldsymbol{E_k^*} \subseteq \mathcal{A}$ such that:

$$\boldsymbol{E_k^*} = argmin_{\boldsymbol{E_k} \subseteq \mathcal{A}} \left[CI(O, T, C, \boldsymbol{E_k}) + Rd(\boldsymbol{E_k}) \right] \quad (4)$$

The MCIMR algorithm selects attributes incrementally as follows (as defined in Equation 4). In the k-th iteration we have the $k-1$-size attribute set $\boldsymbol{E_{k-1}}$. The k-th attribute to be added is the attribute that minimizes the following condition:

$$E_k = argmin_{E \in \mathcal{A} \setminus \boldsymbol{E_{k-1}}} \left[I(O; T|C, E) + \frac{1}{k-1} \sum_{E_i \in \boldsymbol{E_{k-1}}} I(E; E_i) \right] \quad (5)$$

We prove that the combination of the MCI and MRd criteria is equivalent to Equation 1. Namely, the MCIMR algorithm correctly computes the optimal k-size solution.

Theorem 4.1. *The MCIMR algorithm yields the optimal k-size solution to Equation 1.*

Stopping Criteria. Up until this point we assumed that the size of the explanation k is given. However, given two consecutive solutions of sizes k and $k+1$, we can not say if $I(O; T|C, \boldsymbol{E_k}) < I(O; T|C, \boldsymbol{E_{k+1}})$ or vice versa. As mentioned, we assume that attributes in which their marginal explanation power is small are of no interest to analysts. We thus stop the algorithm after the first iteration in which the responsibility of the new attribute to be added is ≈ 0. Namely, we treat k as an upper bound on the explanation size. To this end, we propose the *responsibility test*. Given the set of selected attributes $\boldsymbol{E_k}$, this test verifies if the responsibility of a candidate attribute E_{k+1} is ≈ 0.

Lemma 4.2 (Responsibility test). *If $O \perp\!\!\!\perp E_{k+1} | \boldsymbol{E_k}$ then $Resp(E_{k+1}) \leq 0$.*

We measure conditional independence using the highly efficient independence test proposed in [63].

The MCIMR algorithm is depicted in Algorithm 1. First, it initializes the attribute set $\boldsymbol{E}$ to be returned with the empty set (line 2). Then, new attributes are iteratively added according to the NextBestAtt procedure (line 4). The algorithm then applies the responsibility test to a selected attribute. If the responsibility of this attribute is ≈ 0, the algorithm terminates and returns the solution obtained until this point (lines 5-7). Otherwise, it terminates after k iterations (line 9). Given the attribute set selected up until the i-th iteration, the NextBestAtt procedure finds the i-th attribute to be added. It implements Equation 5, by iterating over all candidate attributes and computing their individual explanation power (line 14), and their redundancy with selected attributes (lines 16-18). For simplicity, we omitted parts dedicated to handling missing data from presentation. In our implementation, before executing lines 14 and 18, we check if weights are needed to be added and adjust the computation accordingly.

Proposition 4.3. *The time complexity of the incremental MCIMR algorithm is $O(k|\mathcal{A}|)$.*

The size of $\mathcal{A}$ is potentially very large. Thus, in the next section, we propose several optimizations to reduce it.

4.2 Pruning Optimizations

We propose several optimizations to reduce the size of $\mathcal{A}$ and thereby reduce execution times. These optimizations are used to prune attributes that are either uninteresting as an explanation or cannot be a part of the optimal solution, and significantly improve running times. We propose two types of optimizations: **Across-queries optimizations** that could be executed at pre-processing; and **Query-specific optimizations** that could be done only once O and T are known and are executed before running the MCIMR algorithm.

Preprocessing pruning. Attributes discarded at this phase either have a fixed value, a unique value for each tuple, or lots of missing values. Thus, such attributes are uninteresting as an explanation [39, 63]. **Simple Filtering**: We drop all attributes with a constant value (e.g., the attribute Type which has the value *Country* to all countries), and attributes in which the percentage of missing values is >90%. **High Entropy**: we discard attributes such as wikiID, that have high entropy and (almost) a unique value for each tuple (as was done in [63]).

Online pruning. **Logical Dependencies**: Logical dependencies can lead to a misleading conclusion that we found a confounding attribute, where we are, in fact, conditioning on an attribute that is functionally dependent on T or O (see proof in [10]). We thus discard all attributes E s.t. $H(T|E) \approx H(E|T) \approx 0$ (resp., for O). These tests correspond to approximate functional dependencies [63], such as CountryCode $\Rightarrow$ Country. **Low Relevance**: As mentioned, we assume that the optimal explanation does not contain attributes which are individually unimportant but become important in the context of others. We leverage this assumption to prune attributes in which their individual explanation power is low (tested using conditional entropy, see full details in [10]).

Another possible optimization is to cluster attributes that are highly correlated, such as HDI and HDI Rank, to reduce the redundancy among attributes [39]. However, we found this optimization to be not useful because of: (1) It could only be done after the query arrives, namely after we are done filtering, and the clustering process took longer than running our algorithm on all attributes. (2) We found that attributes clustered together were not necessarily semantically related.

4.3 Identifying Unexplained Subgroups

The MCIMR algorithm finds the explanations for the correlation between T and O. While the generated explanation is optimal considering the whole data, it may be insufficient for some parts in the data. We thus propose an algorithm the analyst may use after getting the explanation, to identify unexplained data subgroups. It receives the original query Q and its generated explanation. The output is a set of data groups correspond to context refinements of Q, in which a different explanation is required and thus may be of interest to the analyst.

Example 4.1. *Consider a query compare the average salary of developers among countries. The explanation found by MESA is* $\boldsymbol{E}$=*{HDI, Gini}. As mentioned, the HDI of all countries in Europe is similar. Thus, for countries in Europe, it is likely that* $\boldsymbol{E}$ *is not a satisfactory explanation.*

For simplicity, numerical attributes are assumed to be binned. Data groups are defined by a set of attribute-value assignments and correspond to refinements of the context C of Q. Treating the context C as a set of conditions, a refinement C' of C is a set s.t. $C' \subset C$. We search for the largest data groups s.t. $\boldsymbol{E}$ can not serve as their explanation. Formally, given an explanation $\boldsymbol{E}$, $I(O;T|C,\boldsymbol{E})$ is referred to as the explanation score for C. We are inserted in the top-k data groups (in terms of size), each correspond to a context refinement C' of C, s.t. their explanation score is $>\tau$ for some threshold τ (τ can be set based on the initial explanation score).

Example 4.2. *Continuing with Example 4.1, we refine Q by adding a WHERE clause selecting only countries in Europe (*$C' = \{$Continent = Europe$\}$*). Let* Q_{EU} *denote this query. We get:* $I(O;T|C',\boldsymbol{E})$=2.13. *As mentioned in Example* 2.3*, the optimal explanation for* Q_{EU} *is* {Gini, Density}.

A naive algorithm would traverse over all possible contexts refinements C', check if the explanation score is $>\tau$, and will choose the largest data groups for which $\boldsymbol{E}$ is not a satisfactory explanation. We propose an efficient algorithm, exploiting the notion of pattern graph traversal [15]. Intuitively, the set of all context refinements can be represented as a graph where nodes correspond to refinements and there is an edge between C and C' if C' can be obtained from C by adding a single value assignment. This graph can be traversed in a top-down fashion, while generating each node at most once (see [10]).

Algorithm 2 depicts the search for the largest k data groups that for which $\boldsymbol{E}$ is not a satisfactory explanation. It traverses the refinements graph in a top-down manner, starting for the children of C. It uses a max heap *MaxHeap* to iterate over the refinements by their size. It first initialize the result set $\mathcal{R}$ (line 1) and *MaxHeap* with the children of C (line 2). Then, while the $\mathcal{R}$ consists of less

Algorithm 2: Top-k unexplained data groups.

input : A number k, a set of attributes $\mathcal{A}$, the attributes O and T, the context C, an explanation $\boldsymbol{E}$, and a threshold τ.
output: Context refinements $\{C_1, \ldots, C_k\}$ s.t. the corresponding groups are the largest k groups and $I(O;T|C_i,\boldsymbol{E})>\tau$

```
1  R ← ∅
2  MaxHeap ← GenChildren(C)
3  while |R| < k or MaxHeap.isEmpty() do
4      C' ← MaxHeap.exatractMax()
5      if I(O;T|C',E) > τ then
6          update(R, C')   // If none of the ancestors of C' are in R,
                           insert C' into R.
7      else
8          for C'' ∈ GenChildren(C') do
9              MaxHeap.insert(C'')
10 return R
```

Table 1: Examined Datasets.

Dataset	n	$\lvert\mathcal{E}\rvert$	Columns used for extraction
SO [7]	47623	461	Country, Continent
COVID-19 [3]	188	463	Country, WHO-Region
Flights [4]	5819079	704	Airline, Origin/Destination city/state
Forbes [6]	1647	708	Name

then k refinements (line 3), the algorithm extracts the largest (by data size) refinement C' (line 4) and computes $I(O;T|C',\boldsymbol{E})$. If it exceeds the threshold τ (line 5), C' is used to update $\mathcal{R}$ (line 6). The procedure `update` checks whether any ancestor of C' is already in $\mathcal{R}$ (this could happen because the way the algorithm traverses the graph). If not, C' is added to $\mathcal{R}$. If $I(O;T|C',\boldsymbol{E})\leq\tau$ (line 5), the children of C' are added to the heap (lines 8– 9).

Proposition 4.4. *Algorithm 2 yields the top-k largest data groups in which their explanation score is grater than* τ.

In the worst case, there are no such k data groups and hence the algorithm traverses over every possible context refinement of Q, which is polynomial in the number of attributes and (binned) values. However, as we show, in practice this algorithm efficiently identifies the data groups of interest, while exploring only an handful of context refinements.

5 EXPERIMENTAL STUDY

We present experiments that evaluate the effectiveness and efficiency of our solution. We aim to address the following research questions. $Q1$: What is the quality of our explanations, and how does it compare to that of existing methods? $Q2$: How robust are the explanations to missing data? $Q3$ What is the efficiency of the proposed algorithm and the optimization techniques? $Q4$: How useful are our proposed extensions?

Our code and datasets are available at [10]. We used DBPedia KG [8] for attribute extraction, and the Pyitlib library [9] for information-theoretic computations. The experiments were executed on a PC with a 4.8GHz CPU, and 16GB memory.

Datasets. We examine four commonly used datasets: **(1) SO**: Stack Overfow's annual developer survey is a survey of people who code around the world. It has more than 47K records containing information about the developers' such as their age, income, and country. **(2) Covid-19**: This dataset includes information such as number of confirmed, death, and new cases in 2020 across the globe. **(3) Flights Delay**: This dataset contains transportation statistics

of over 5.8M domestic flights operated by large air carriers in the USA. **(4) Forbes**: This dataset contains annual earning information of 1.6K celebrities between 2005 – 2015 It contains the celebrities' annual pay, and category (e.g., Actors, Producers).

The attributes used for property extraction and the number of extracted attributes in each dataset are given in Table 1.

Baseline Algorithms. We compare MESA against the following baselines: **(1) Brute-Force**: The optimal solution according to Def. 2.1. This algorithm implements an exhaustive search over all subsets of attributes. To make it feasible, we run it after employing our pruning optimizations. **(2) Top-K**: This naive algorithm ranks the attributes according to their individual explanation power (equivalent to Max-Relevance only). **(3) Linear Regression** (LR): This baseline employs the OLS method to estimate the coefficients of a linear regression describing the relationship between the outcome and the candidate attributes. The explanations are defined as the top-k attributes with the highest coefficients (s.t. the p value is $<.05$). Note that Pearson's r is the standardized slope of LR and thus can be viewed as part of our competing baselines. **(4) HypDB** [63]: This system employs an algorithm for confounding variable detection based on causal analysis. The explanations are defined as the top-k attributes with the highest responsibility scores. **(5) MESA$^-$**: Last, to examine how pruning affects the explanation, we examine the explanation generated by MESA without the pruning optimizations.

We also examined the explanations generated by CajaDE [39], a system that generates query results explanations based on augmented provenance information. However, since in all cases, CajaDE generated explanations obtained the lowest scores, we omit its results from presentation. The reason for that CajaDE explanations are a set of patterns that are unevenly distributed among groups in the query results, which are independent of the outcome variable. Thus, it cannot generate explanations that explain the correlation between T and O.

Unless mentioned otherwise, we set the maximal explanation size, k, to 5 and extracted attributes for 1-hop in the KG. For a fair comparison, we run all baselines (except for MESA$^-$) after employing our pruning optimizations.

5.1 Quality Evaluation (Q1)

We validate our intuition that attributes extracted from KGs can explain correlations in common scenarios. To this end, we randomly generated 40 SQL queries (10 from each dataset) as follows. We set T to be one of the attributes used for attribute extraction (as listed in Table 1). We set O to be a numerical attribute that could be predicted from the data (e.g., Departure/Arrival Delay in Flights, New/Death Cases in Covid-19). We then added a WHERE clause by randomly picking another attribute and one of its values, ensuring selected subsets contain more than 10% of the tuples in the original dataset. Full details are given in the Appendix. We say our approach was useful if (1) the partial correlation between the exposure and outcome (while conditioning on an explanation generated by MESA) is lower than the original correlation, and (2) the explanation contains at least one extracted attribute. We report this was the case in 72.5% percent of the queries.

Next, we aim to asses the quality of the generated explanations to validate our problem definition. To this end, we present a user study consisting of explanations produced by each algorithm. Since a standard benchmark for results explanation does not exist, we consider 14 representative queries suffering from confounding bias, as shown in Table 2. Our queries are inspired by real-life sources, such SO annual reports [5], news and media websites (e.g., Vanity Fair [1], USA Today[2] for Forbes and Flights), and academic papers [36, 69]. Similar experiments were conducted in [39, 42, 63]. To compare the generated explanations with the "ground-truth" explanations, we will show that our explanations are supported by previous findings. A similar approach was taken in [63].

We recruited 150 subjects on Amazon MTurk. This sample size enables us to observe a 95% confidence level with a 10% margin of error. Subjects were asked to rank each explanation of each method (shown together with its corresponding query) on a scale of 1–5, where 1 indicates that it does not make sense and 5 indicates that the explanation is highly convincing. The form we gave to the subjects is available at [10].

HypDB's time complexity is exponential in the size of $\mathcal{A}$ [63]. We run it over all attributes in $\mathcal{A}$ (after pruning) and report that it never terminates within 10 hours. Thus, we have no choice but to limit the number of attributes for HypDB, to allow it to generate explanations in a reasonable time. For HypDB, besides pruning, we omitted candidate attributes uniformly at random, ensuring that $|\mathcal{A}|\leq 50$. We only report the results of Brute-Force for the small Covid-19 and Forbes datasets, as it was infeasible to compute them for the larger datasets. We do not randomly drop attributes for computational efficiency here because Brute-Force is intended to be an optimal solution for our problem definition against which our algorithm is judged. The explanations generated by different methods are given in Table 2, and the average explanation scores given by the subjects are depicted in Table 3.

We summarize our main finding as follows:

- The subjects found the explanations generated by Brute-Force, MESA$^-$, and MESA to be the most convincing. This supports our mathematical definition (Def 2.1) of what constitutes a good explanation.
- MESA explanations are supported by previous in-domain findings, which serve as "ground-truth" explanations.
- Our pruning has little effect on explanation quality.
- The next best competitor is HypDB. However, it is unable to scale to a large number of candidate attributes.
- As expected, Top-k yields redundancy in selected attributes.

First, subjects found the explanations generated by Brute-Force, MESA$^-$, and MESA to be the most convincing. The pairwise differences between the average scores of these 3 methods are not statistically significant. Previous in-domain findings also support these explanations. For example, in SO Q_1, it was shown in [5] that there is a correlation between developers salary and countries' economies (reflected in the HDI and Gini values). For Flights Q_1, it was stated in [2] that weather is one of the top reasons for flights delay in the US. For Covid-19 Q_1, it was shown that there is a correlation between countries' economies and Covid-19 death rate [36, 69]. More details can be found in the Appendix. In all cases where the results of Brute-Force and MESA are different, it happens because MESA drops attributes with insignificant responsibility

Table 2: User study: The best and second best explanations are marked in red and blue, resp.

Dataset		Query	Brute-Force	MESA-	MESA	Top-K	LR	HypDB
SO	Q_1	Average salary per country	-	**HDI Rank, Gini**	**HDI, Gini**	HDI, Established Date	Population Census, Language	GDP
	Q_2	Average salary per continent	-	**GDP Rank, Density**	**GDP,Density**	GDP,Area rank	GDP, Area Rank	GDP
	Q_3	Average salary per country in Europe	-	**Population Census, Gini Rank**	**Population Census, Gini**	Population Census, Population Estimate	Population Census, Language	Gini, Area Rank
Flights	Q_1	Average delay per origin city	-	**Precipitation Days, Year UV, Airline**	Population urban, Year Low F, Airline	Year Low F, Year Avg F, December Low F	Year Low F, December percent sun, Day	**Year Low F, May Precipitation Inch, Airline**
	Q_2	Average delay per origin state	-	**Density, Year Snow, Airline**	**Population estimation, Year Low F, Airline**	Population estimation, Population Urban, Population Rank	Population estimation, Median Household Income, Distance	Record Low F, Population estimation, Day
	Q_3	Average delay per origin cities in CA	-	**Density, Population Metropolitan, Security Delay**	Density, Population Total,Security Delay	Population Metropolitan,Security Delay	-	**Density, Population Ranking, Cancelled**
	Q_4	Average delay per origin state and airline	-	**Population Total, Fleet size**	**Population Ranking, Fleet size**	Density, Population Total	-	Revenue, Dec Record Low F
	Q_5	Average delay per airline	-	**Equity, Fleet Size**	**Equity, Fleet Size**	Equity, Net Income	**Equity, Fleet Size**	**Num of Employees, Revenue**
Covid-19	Q_1	Deaths per country	**HDI, GDP, Confirmed cases**	**HDI, GDP Rank, Confirmed cases**	**HDI, GDP, Confirmed cases**	GDP Rank, GDP Nominal, HDI	Area Rank, Currency, Recovered cases	Density, Time Zone, Confirmed cases
	Q_2	Deaths per country in Europe	**Gini, Population Census, Confirmed cases**	**Gini Rank, Density, Confirmed cases**	**Gini, Population Census, Confirmed cases**	Gini Rank, Gini, GDP	Area Rank, Currency, Population Total	Currency, GDP, New cases
	Q_3	Average deaths per WHO-Region	**Density, Confirmed Cases**	**Density,Confirmed Cases**	**Density,Confirmed Cases**	**Density,Confirmed Cases**	-	**Area Km,Confirmed Cases**
Forbes	Q_1	Salary of Actors	**Net Worth, Gender, Age**	**Net Worth, ActiveSince, Gender**	Net Worth, Gender	Net worth, Awards	Citizenship, Honors	Gender, Honors
	Q_2	Salary of Directors/Producers	**Net Worth, Awards**	**Years Active, Net Worth**	**Net Worth, Awards**	Net Worth, Age	-	Years Active
	Q_3	Salary of Athletes	**Cups, Draft Pick, Active Years**	National Cups, Draft Pick	Cups, Draft Pick	Total Cups, National Cups	-	**Cups, Active Years**

Table 3: Avg. explanation scores according to the subjects.

Baseline	Average Score	Average Variance
Brute-Force	3.8	0.8
MESA-	3.7	1.1
MESA	3.5	0.9
HypDB	2.8	1.1
Top-K	2.1	0.8
LR	1.8	0.6

(according to the responsibility test). For example, in Forbes Q_1, MESA dropped Age. The low difference between the results of MESA$^-$ and MESA indicates that pruning has little effect on explanations quality. Namely, *MESA is able to execute efficiently without compromising on explanation quality.*

The explanations of all methods consist of attributes extracted from the KG. This validates our assumptions that KGs can serve as valuable sources for results explanations. The next best competitor is HypDB (the average score is worse than that of MESA. This difference is statistically significant, $p<.05$). This is not surprising as HypDB finds confounding attributes using causal analysis. However, its main disadvantage is its ability to scale for large number of attributes. In cases where HypDB generated explanations that were considered not convincing, it was mainly because important attributes were dropped (as we limited the number attributes to enable feasible execution times). Not surprisingly, the explanations generated by Top-K and LR were considered to be less convincing (their average scores are statistically significant from all other methods, $p<.05$). For Top-K, this is substantially because it ignores redundancy among attributes. For example, in Flights Q_1, it chose the attributes Year Low F and Year Average F, which are highly correlated. For LR, in many cases, it failed to generate explanations, as there were no attributes with low enough p-values. Even when it succeeded, the subjects found them to be not convincing. The reason is that LR focuses on finding linear correlations.

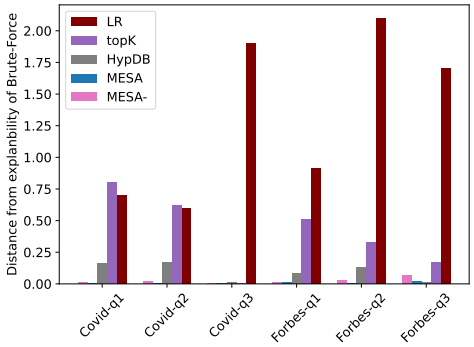
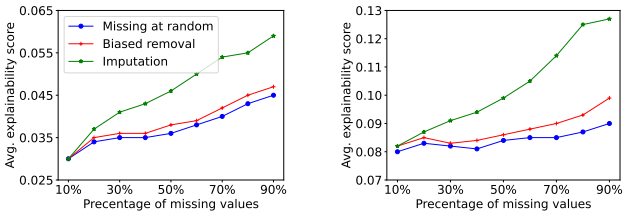


Figure 2: Distance from explainability scores of Brute-Force.

Explainability scores. Let $\boldsymbol{E}$ denote the explanation found by an algorithm. We call $I(O;T|\boldsymbol{E})$ the explainability score. Explainability score equal to 0 means that $\boldsymbol{E}$ perfectly explains the correlation between O and T. The explainability scores of Brute-Force serve as the gold standard (as by definition, it aims to minimize this score). In some cases, the explanations generated by all algorithms, including Brute-Force, cannot fully explain the correlations. E.g., in Flights Q_2, the explainability score of Brute-Force is 0.25. This means that other factors that affect flight delays may not exist in the KG (e.g., labor problems). The results are depicted in Figure 2. The y-axis is the distance between the explainability scores of each method and Brute-Force. The lower the distance the better is the explanation. Observe that the explainability scores of MESA are almost as good as the ones of Brute-Force and MESA$^-$, and are much better than those of the competitors.

Additional experiments can be found in the Appendix.



SO **Covid-19**

Figure 3: Explainability as a function of missing data.

5.2 Robustness to Missing Data (Q2)

On average, the percentage of missing values in extracted attributes is 37%, 42%, 45% and 73% in Covid-19, SO, Flights and Forbes, resp. The high prevalence of missing values in Forbes is because DBpedia uses different attributes to describe a person from each category (e.g., actors, authors). In Covid-19, SO, Flights, and Forbes, the percentage of attributes with selection bias is 13.3%, 14.1%, 24.2%, and 29.4%, resp. *This verifies that selection bias exists in attributes extracted from KGs, and thus should be appropriately handled.*

We examine the robustness of our explanations to missing data, by varying the percentage of missing values from the top 10 most relevant (w.r.t. the outcome) attributes. We examine two ways to omit values: missing-at-random and biased removal, where the top-x highest values from examined attributes were omitted (when varying x). We examine the effect on our generated explanations average explainability score. Explainability should not be affected if an explanation is robust to missing data. We also examine the effect on the explainability scores while imputing missing values (using the common mean imputation technique [73]). The results for the SO and Covid datasets are depicted in Figure 3. As expected, data imputation has huge negative effect on explainability. Our approach is much less sensitive to missing data: Even with 50% missing values (at random or not), the explainability scores have hardly changed. When the percentage of missing values is above 50%, a lot of the information is lost, and thus it is harder to estimate partial correlation correctly.

5.3 Efficiency Evaluation (Q3)

To examine the contribution of our optimizations, we report the running times of the following baselines: **No Pruning** —the MCIMR algorithm without pruning; **Offline Pruning** —MCIMR with only offline pruning. We study the effect of multiple parameters on running times. For each dataset, we report the average execution time of the queries presented in Section 5.1. In all cases, the execution time of MCIMR was less than 10 seconds, a reasonable response time for an interactive system. We omit the results obtained on the (smallest) Covid-19 dataset from presentation, as the results demonstrated similar trends to those of Forbes.

Candidate Attributes. In this experiment, we omitted from consideration attributes from $\mathcal{A}$ uniformly at random. The results are depicted in Figure 4. In all dataset, we exhibit a (near) linear growth in running times as a function of the size of $\mathcal{A}$. The execution times of No-Pruning are significantly higher than those of Offline Pruning and MCIMR, *indicating the usefulness of the offline pruning.*

Table 4: Top-5 unexplained groups for SO Q1.

Rank	Size	Data group
1	18342	Continent = Europe
2	17899	Continent = Asia
3	15466	Continent = North America
4	14788	Currency = Euro
5	12754	Continent = Africa

The difference in times across datasets is due to their size. Estimating CMI on large datasets (e.g., Flights, SO) takes longer than on small datasets (e.g., Forbes). In Forbes, Offline Pruning is faster than MCIMR, implying that in small datasets online pruning is not necessary, as it takes longer that running MCIMR.

Data Size. We vary the number of tuples in $\mathcal{D}$, by removing tuples uniformly at random. The results are depicted in Figure 5. In SO and Flights, observe that the dataset size has a little effect on running times. This is because the size of the subgroups in the group-by queries were big. Thus, when randomly omitting tuples from the datasets, the number of considered groups is almost unchanged. On the other hand, since in Forbes each group contained only a few records, we exhibit a (near) linear growth in running times.

Explanation size. We vary the bound on the explanation size. Recall that given a bound k, MCIMR returns an explanation of size $\leq k$. It may return an explanation of size $l<k$ if the responsibility of the l+1 attribute is $\approx$0. The results are shown in Figure 6. In all cases, the size of the explanations was no bigger than 3. Thus, k has almost no effect on running times, as the algorithms terminate after no more than 4 iterations.

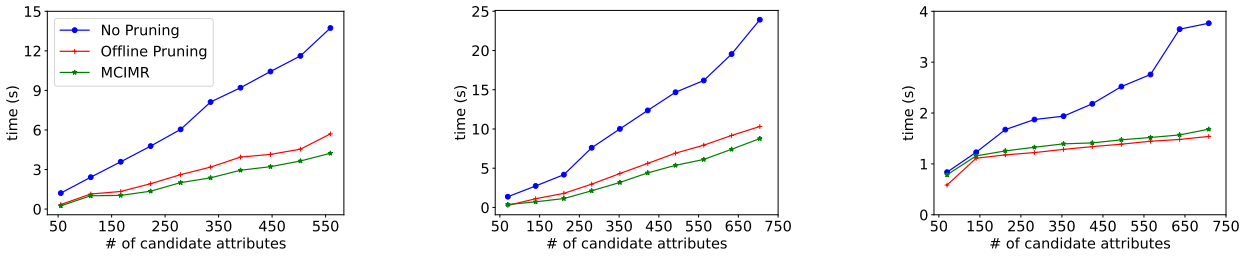
5.4 Extensions (Q4)

We examine the effect of extracting attributes following more than one hop in the KG. We report that in the vast majority of cases, MESA's explanations were unaffected, indicating that most of the relevant information can be found in the first hop. Further details can be found in the Appendix .

Unexplained Subgroups. We demonstrate the effectiveness of the Top-K unexplained groups algorithm by focusing on SO Q_1, setting τ>0.2. The top-5 largest unexplained data groups are given in Table 4. Observe that economy-related attributes (e.g., GDP, HDI) of selected data groups are internally consistent (e.g., the HDI of countries in Europe is similar). Thus, it makes sense that the explanation for SO Q_1 ({HDI, Gini}) will not be a satisfactory explanation for these data groups. Indeed, as shown in Table 2, the explanation of MESA for the top-1 unexplained group (SO Q_3) is different from the one found for all countries. We ran this algorithm over the other queries as well. The average execution time is 4.4s. This demonstrates the ability of our algorithm to efficiently identify data subgroups that are likely to be of interest to users.

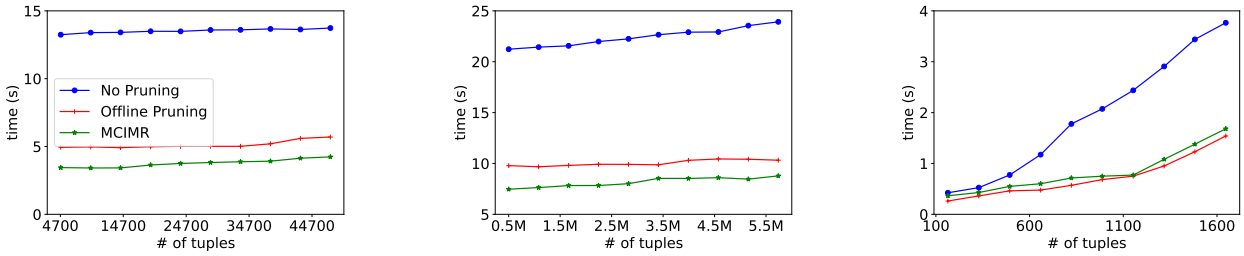
6 RELATED WORK

Results Explanations. Methods explaining why data is missing or mistakenly included in query results have been studied in [18, 21, 38, 67]. Explanations for unexpected query results have been presented in [17, 49]. Those works are orthogonal to our work, as we aim to explain unexpected correlations. Another line of work provides explanations on how a query result was derived by analyzing its provenance and pointing out tuples that significantly affect the results [46, 47, 50]. Those methods are designed to generate tuple-level



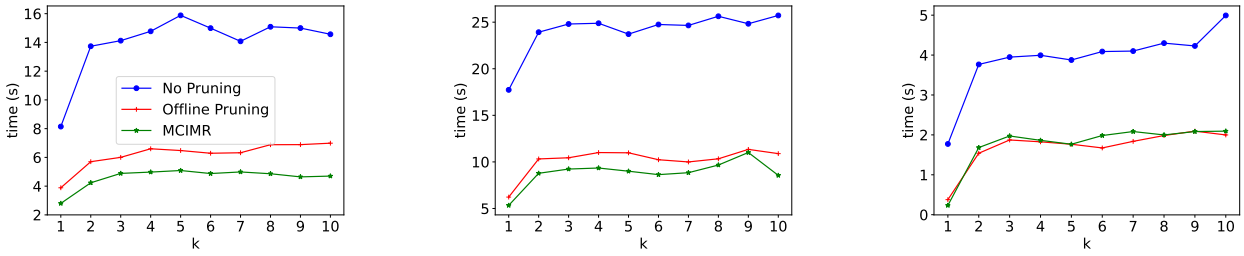
SO **Flights** **Forbes**

Figure 4: Running times as a function of the number of candidate attributes.



SO **Flights** **Forbes**

Figure 5: Running times as a function of the number of rows in the dataset.



SO **Flights** **Forbes**

Figure 6: Running times as a function of the bound on the explanation size.

explanations and not attribute-level explanations that are required for unearthing correlations. Another type of explanation for query results is a set of patterns that are shared by one (group of) tuple but not by another (group of) tuple [30, 39, 61, 62, 70]. However, those works as well do not account for correlations among attributes. [63] presented HypDB, a system that identifies confounding bias in SQL queries for improved decision making, detected using causal analysis. However, as mentioned, HypDB only considers attributes from the input table, and it cannot efficiently handle a large amount of candidate attributes.

We share with [39] the motivation for considering explanations that are not solely drawn from the input table. [39] presented CajaDE, a system that generates explanations of query results based on information from tables related to the table accessed by the query. However, related tables often do not exist. Moreover, as mentioned in Section 5, their explanations are *independent of the outcome*. Thus, even if CajaDE is given the attributes mined from other sources, it may generate explanations that are irrelevant to the correlation between the exposure and outcome.

Dataset Discovery. Given an input dataset, dataset discovery methods find related tables that can be integrated via join or union operations. Existing methods estimate how joinable or unionable two datasets are [53, 71, 76, 77]. Other works focused on automating the data augmentation task to discover relevant features for ML models [23]. While these works focus on finding datasets that are joinable or unionable, we aim to find unobserved attributes that explain unexpected correlations. Recent work proposed solutions to discover datasets that can be joined with an input dataset and contain a column that is correlated with a target column [31, 64]. Such techniques can be integrated into our system for extracting candidate attributes from tabular data. We focus on finding attributes that minimize the partial correlation between two columns rather than finding columns that are correlated with a target column. Thus, future work will extend these techniques to support our goal.

Feature Selection. The Correlation-Explanation problem is closely related to the well-studied Feature Selection (FS) problem [20, 34, 40]. FS methods select a concise and diverse set of

attributes relevant to a target attribute for use in model construction [20], whereas we aim to select a conciseness and no-redundant set of attributes that are correlated to the outcome and exposure. Closest to our project is a line of work using information-theoretic methods for FS [40]. Algorithms in this family [29, 33, 41, 48] define different criteria to maximize feature relevance and minimize redundancy. Relevance is typically measured by the feature correlation with the target attribute. Of particular note, the MRMR algorithm [57] selects features based on Max-Relevance and Min-Redundancy criteria. A main difference in MCIMR is that instead of the Max-Relevance criterion, we use a Min-CMI criterion. Another critical difference is the stopping condition. While in MRMR, the size k of the selected feature set is determined using the underlying learning model, in MCIMR, we set k using responsibility scores.

Explainable AI. A related line of work is Explainable AI (XAI), an emerging field in machine learning that aims to address how black box decisions of AI systems are made [12, 27]. Similar to our approach, XAI can be used to learn new facts, to gather information and thus to gain knowledge [12]. We share the motivation with posthoc XAI methods [14, 26, 45], which extract explanations from already learned models. The advantage of this approach is that it does not impact the performance of the model, which is treated as a black box. In MESA as well, we generate explanations after the SQL query was executed, independently from the database engine.

7 CONCLUSION AND LIMITATIONS

This paper presented the Correlation-Explanation problem, whose goal is to identify uncontrolled confounding attributes that explain unexpected correlations observed in query results. We developed an efficient algorithm that finds the optimal subset of attributes. This algorithm is embodied in a system called MESA, which adapts the IPW technique for handling missing data. MESA is applicable for cases where explanations can be found in a given external knowledge source. In this paper we focused on extracting attributes from KGs. Future work would extracted candidate attributes from other sources, such as unstructured data (e.g., text documents). Another interesting future work is to identify which links in a KG are relevant to the explanation and worthy to follow.

REFERENCES

[1] 2018. The Vanity Fair. https://www.vanityfair.com/hollywood/2018/02/hollywood-movie-salaries-wage-gap-equality.
[2] 2019. The USA Today. https://www.usatoday.com/story/travel/airline-news/2022/06/19/why-us-flights-canceled-delayed-sunday/7677552001/.
[3] 2020. COVID-19 Dataset. https://www.kaggle.com/imdevskp/corona-virus-report?select=usa_county_wise.csv.
[4] 2020. Flights Delay Dataset. https://www.kaggle.com/usdot/flight-delays?select=flights.csv.
[5] 2021. 2021 Stackoverflow Developer Survey. https://insights.stackoverflow.com/survey/2021.
[6] 2021. Forbes Dataset. https://www.kaggle.com/datasets/slayomer/forbes-celebrity-100-since-2005.
[7] 2021. Stack Overfow developer survey. https://insights.stackoverflow.com/survey.
[8] 2022. DBPedia. https://www.dbpedia.org/.
[9] 2022. Pyitlib library. https://pypi.org/project/pyitlib/.
[10] 2022. Technical Report. https://github.com/ResultsExplanations/ExplanationsFromKG.
[11] 2022. Wikidata. https://www.wikidata.org/wiki/Wikidata:Main_Page.
[12] Amina Adadi and Mohammed Berrada. 2018. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE access* 6 (2018), 52138–52160.
[13] Farahnaz Akrami, Mohammed Samiul Saeef, Qingheng Zhang, Wei Hu, and Chengkai Li. 2020. Realistic re-evaluation of knowledge graph completion methods: An experimental study. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 1995–2010.
[14] Anneleen Van Assche and Hendrik Blockeel. 2007. Seeing the forest through the trees: Learning a comprehensible model from an ensemble. In *European Conference on machine learning*. Springer, 418–429.
[15] Abolfazl Asudeh, Zhongjun Jin, and H. V. Jagadish. 2019. Assessing and Remedying Coverage for a Given Dataset. In *35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019*. IEEE, 554–565. https://doi.org/10.1109/ICDE.2019.00056
[16] Y. Bengio et al. 2013. Representation learning: A review and new perspectives. *IEEE PAMI* 35, 8 (2013), 1798–1828.
[17] Aline Bessa, Juliana Freire, Tamraparni Dasu, and Divesh Srivastava. 2020. Effective Discovery of Meaningful Outlier Relationships. *ACM Transactions on Data Science* 1, 2 (2020), 1–33.
[18] Nicole Bidoit, Melanie Herschel, and Katerina Tzompanaki. 2014. Query-based why-not provenance with nedexplain. In *Extending database technology (EDBT)*.
[19] Laura E Brown and Ioannis Tsamardinos. 2008. Markov blanket-based variable selection in feature space.
[20] Girish Chandrashekar and Ferat Sahin. 2014. A survey on feature selection methods. *Computers & Electrical Engineering* 40, 1 (2014), 16–28.
[21] Adriane Chapman and HV Jagadish. 2009. Why not?. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*. 523–534.
[22] Xiaojun Chen, Shengbin Jia, and Yang Xiang. 2020. A review: Knowledge reasoning over knowledge graph. *Expert Systems with Applications* 141 (2020), 112948.
[23] Nadiia Chepurko, Ryan Marcus, Emanuel Zgraggen, Raul Castro Fernandez, Tim Kraska, and David Karger. 2020. ARDA: automatic relational data augmentation for machine learning. *arXiv preprint arXiv:2003.09758* (2020).
[24] Frederick E Croxton and Dudley J Cowden. 1939. Applied general statistics. (1939).
[25] Christopher De Sa, Alex Ratner, Christopher Ré, Jaeho Shin, Feiran Wang, Sen Wu, and Ce Zhang. 2016. Deepdive: Declarative knowledge base construction. *ACM SIGMOD Record* 45, 1 (2016), 60–67.
[26] Yinpeng Dong, Hang Su, Jun Zhu, and Fan Bao. 2017. Towards interpretable deep neural networks by leveraging adversarial examples. *arXiv preprint arXiv:1708.05493* (2017).
[27] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. 2018. Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*. IEEE, 0210–0215.
[28] Bradley Efron. 1994. Missing data, imputation, and the bootstrap. *J. Amer. Statist. Assoc.* 89, 426 (1994), 463–475.
[29] Ali El Akadi, Abdeljalil El Ouardighi, and Driss Aboutajdine. 2008. A powerful feature selection approach based on mutual information. *International Journal of Computer Science and Network Security* 8, 4 (2008), 116.
[30] Kareem El Gebaly, Parag Agrawal, Lukasz Golab, Flip Korn, and Divesh Srivastava. 2014. Interpretable and informative explanations of outcomes. *Proceedings of the VLDB Endowment* 8, 1 (2014), 61–72.
[31] Mahdi Esmailoghli, Jorge-Arnulfo Quiané-Ruiz, and Ziawasch Abedjan. 2021. COCOA: COrrelation COefficient-Aware Data Augmentation.. In *EDBT*. 331–336.
[32] Ronald Fagin, Alberto O Mendelzon, and Jeffrey D Ullman. 1982. A simplied universal relation assumption and its properties. *ACM Transactions on Database Systems (TODS)* 7, 3 (1982), 343–360.
[33] François Fleuret. 2004. Fast binary feature selection with conditional mutual information. *Journal of Machine learning research* 5, 9 (2004).
[34] Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *Journal of machine learning research* 3, Mar (2003), 1157–1182.
[35] David Hinkley. 1985. Transformation diagnostics for linear models. *Biometrika* 72, 3 (1985), 487–496.
[36] A Kaklauskas, V Milevicius, and L Kaklauskiene. 2022. Effects of country success on COVID-19 cumulative cases and excess deaths in 169 countries. *Ecological indicators* (2022), 108703.
[37] Joseph DY Kang and Joseph L Schafer. 2007. Demystifying double robustness: A comparison of alternative strategies for estimating a population mean from incomplete data. *Statistical science* 22, 4 (2007), 523–539.
[38] Seokki Lee, Bertram Ludäscher, and Boris Glavic. 2020. Approximate summaries for why and why-not provenance (extended version). *arXiv preprint arXiv:2002.00084* (2020).
[39] Chenjie Li, Zhengjie Miao, Qitian Zeng, Boris Glavic, and Sudeepa Roy. 2021. Putting Things into Context: Rich Explanations for Query Answers using Join Graphs. In *Proceedings of the 2021 International Conference on Management of Data*. 1051–1063.
[40] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. 2017. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)* 50, 6 (2017), 1–45.

[41] Dahua Lin and Xiaoou Tang. 2006. Conditional infomax learning: An integrated framework for feature extraction and fusion. In *European conference on computer vision*. Springer, 68–82.

[42] Yin Lin, Brit Youngmann, Yuval Moskovitch, HV Jagadish, and Tova Milo. 2021. On detecting cherry-picked generalizations. *Proceedings of the VLDB Endowment* 15, 1 (2021), 59–71.

[43] Brandon Lockhart, Jinglin Peng, Weiyuan Wu, Jiannan Wang, and Eugene Wu. 2021. Explaining Inference Queries with Bayesian Optimization. *Proc. VLDB Endow.* 14, 11 (2021), 2576–2585.

[44] Tania Lombrozo. 2007. Simplicity and probability in causal explanation. *Cognitive psychology* 55, 3 (2007), 232–257.

[45] David Martens, Bart Baesens, Tony Van Gestel, and Jan Vanthienen. 2007. Comprehensible credit scoring models using rule extraction from support vector machines. *European journal of operational research* 183, 3 (2007), 1466–1476.

[46] Alexandra Meliou, Wolfgang Gatterbauer, Katherine F Moore, and Dan Suciu. 2009. Why so? or why no? functional causality for explaining query answers. *arXiv preprint arXiv:0912.5340* (2009).

[47] Alexandra Meliou, Wolfgang Gatterbauer, Katherine F Moore, and Dan Suciu. 2010. The complexity of causality and responsibility for query answers and non-answers. *arXiv preprint arXiv:1009.2021* (2010).

[48] Patrick Emmanuel Meyer, Colas Schretter, and Gianluca Bontempi. 2008. Information-theoretic feature selection in microarray data using variable complementarity. *IEEE Journal of Selected Topics in Signal Processing* 2, 3 (2008), 261–274.

[49] Zhengjie Miao, Qitian Zeng, Boris Glavic, and Sudeepa Roy. 2019. Going beyond provenance: Explaining query answers with pattern-based counterbalances. In *Proceedings of the 2019 International Conference on Management of Data*. 485–502.

[50] Tova Milo, Yuval Moskovitch, and Brit Youngmann. 2020. Contribution Maximization in Probabilistic Datalog. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 817–828.

[51] Sameh K Mohamed, Vít Nováček, and Aayah Nounu. 2020. Discovering protein drug targets using knowledge graph embeddings. *Bioinformatics* 36, 2 (2020), 603–610.

[52] Karthika Mohan, Felix Thoemmes, and Judea Pearl. 2018. Estimation with incomplete data: The linear case. In *Proceedings of the International Joint Conferences on Artificial Intelligence Organization*.

[53] Fatemeh Nargesian, Erkang Zhu, Ken Q Pu, and Renée J Miller. 2018. Table union search on open data. *Proceedings of the VLDB Endowment* 11, 7 (2018), 813–825.

[54] Alberto Parravicini, Rhicheek Patra, Davide B Bartolini, and Marco D Santambrogio. 2019. Fast and accurate entity linking via graph embedding. In *Proceedings of the 2nd Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*. 1–9.

[55] Patricia A Patrician. 2002. Multiple imputation for missing data. *Research in nursing & health* 25, 1 (2002), 76–84.

[56] Judea Pearl. 2009. *Causality*. Cambridge university press.

[57] Hanchuan Peng, Fuhui Long, and Chris Ding. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence* 27, 8 (2005), 1226–1238.

[58] Mohamad Amin Pourhoseingholi, Ahmad Reza Baghestani, and Mohsen Vahedi. 2012. How to control confounding effects by statistical analysis. *Gastroenterology and hepatology from bed to bench* 5, 2 (2012), 79.

[59] Romila Pradhan, Jiongli Zhu, Boris Glavic, and Babak Salimi. 2021. Interpretable Data-Based Explanations for Fairness Debugging. *arXiv preprint arXiv:2112.09745* (2021).

[60] Thomas Rebele, Fabian Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. 2016. YAGO: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *International semantic web conference*. Springer, 177–185.

[61] Sudeepa Roy, Laurel Orr, and Dan Suciu. 2015. Explaining query answers with explanation-ready databases. *Proceedings of the VLDB Endowment* 9, 4 (2015), 348–359.

[62] Sudeepa Roy and Dan Suciu. 2014. A formal approach to finding explanations for database queries. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. 1579–1590.

[63] Babak Salimi, Johannes Gehrke, and Dan Suciu. 2018. Bias in olap queries: Detection, explanation, and removal. In *Proceedings of the 2018 International Conference on Management of Data*. 1021–1035.

[64] Aécio Santos, Aline Bessa, Fernando Chirigati, Christopher Musco, and Juliana Freire. 2021. Correlation sketches for approximate join-correlation queries. In *Proceedings of the 2021 International Conference on Management of Data*. 1531–1544.

[65] Alberto Santos, Ana R Colaço, Annelaura B Nielsen, Lili Niu, Maximilian Strauss, Philipp E Geyer, Fabian Coscia, Nicolai J Wewer Albrechtsen, Filip Mundt, Lars Juhl Jensen, et al. 2022. A knowledge graph to interpret clinical proteomics data. *Nature Biotechnology* (2022), 1–11.

[66] Shaun R Seaman and Ian R White. 2013. Review of inverse probability weighting for dealing with missing data. *Statistical methods in medical research* 22, 3 (2013), 278–295.

[67] Balder ten Cate, Cristina Civili, Evgeny Sherkhonov, and Wang-Chiew Tan. 2015. High-level why-not explanations using ontologies. In *Proceedings of the 34th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*. 31–43.

[68] Ioannis Tsamardinos, Constantin F Aliferis, Alexander R Statnikov, and Er Statnikov. 2003. Algorithms for large scale Markov blanket discovery.. In *FLAIRS conference*, Vol. 2. St. Augustine, FL, 376–380.

[69] Ashwini Kumar Upadhyay and Shreyanshi Shukla. 2021. Correlation study to identify the factors affecting COVID-19 case fatality rates in India. *Diabetes & Metabolic Syndrome: Clinical Research & Reviews* 15, 3 (2021), 993–999.

[70] Eugene Wu and Samuel Madden. 2013. Scorpion: Explaining away outliers in aggregate queries. (2013).

[71] Yang Yang, Ying Zhang, Wenjie Zhang, and Zengfeng Huang. 2019. Gb-kmv: An augmented kmv sketch for approximate containment similarity search. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 458–469.

[72] Y Zhang and Z. Ives. [n.d.]. Finding related tables in data lakes for interactive data science. In *SIGMOD*. 1951–1966.

[73] Zhongheng Zhang. 2016. Missing data imputation: focusing on single imputation. *Annals of translational medicine* 4, 1 (2016).

[74] Juan Zhao, Yiwei Zhou, Xiujun Zhang, and Luonan Chen. 2016. Part mutual information for quantifying direct associations in networks. *Proceedings of the National Academy of Sciences* 113, 18 (2016), 5130–5135.

[75] Weiguo Zheng, Jeffrey Xu Yu, Lei Zou, and Hong Cheng. 2018. Question answering over knowledge graphs: question understanding via template decomposition. *Proceedings of the VLDB Endowment* 11, 11 (2018), 1373–1386.

[76] Erkang Zhu, Dong Deng, Fatemeh Nargesian, and Renée J Miller. 2019. Josie: Overlap set similarity search for finding joinable tables in data lakes. In *Proceedings of the 2019 International Conference on Management of Data*. 847–864.

[77] Erkang Zhu, Fatemeh Nargesian, Ken Q Pu, and Renée J Miller. 2016. LSH ensemble: Internet-scale domain search. *arXiv preprint arXiv:1603.07410* (2016).

[78] Ganggao Zhu and Carlos A Iglesias. 2018. Exploiting semantic similarity for named entity disambiguation in knowledge graphs. *Expert Systems with Applications* 101 (2018), 8–24.

A MISSING PROOFS

In this part we provide missing proofs.

Proof of Corollary **??**. For any two random variables, if $X \perp\!\!\!\perp Y$ we have: $H(Y|X) = H(Y)$. This can be generalized to conditional independence as well. We get:

$$I(O;T|E,R_E=1,C) = H(O|E,R_E=1,C) - H(O|T,E,R_E=1,C) = H(O|E,C) - H(O|T,E,C) = I(O;T|E,C)$$

□

Proof of Corollary **??**. We have:

$$I(E_i;E_j|R_{E_i}=1,R_{E_j}=1) = H(E_i|R_{E_i}=1,R_{E_j}=1) - H(E_i|E_j,R_{E_i}=1,R_{E_j}=1) = H(E_i) - H(E_i|E_j) = I(E_i;E_j)$$

□

In what follows, to ease the exposition, we assume that there is no WHERE clause in the query, i.e., $C = \emptyset$. Our results also hold for cases where C is not empty.

Proof of Theorem 4.1. Recall that by definition of the algorithm, we assume that $\boldsymbol{E_{k-1}}$, i.e., the set of $k-1$ attributes, has already been obtained, and thus $\boldsymbol{E_{k-1}}, O$, and T are fixed when selecting the k-th attribute. The goal is to select the optimal k-th attribute to be added, E_k, from $\mathcal{D} \setminus \boldsymbol{E_{k-1}}$.

By the definition of conditional mutual information, we have:

$$I(O;T|\boldsymbol{E_{k-1}},E_k) = I(O;T|\boldsymbol{E_k}) = H(O;\boldsymbol{E_k}) + H(T;\boldsymbol{E_k}) - H(O;T;\boldsymbol{E_k}) - H(\boldsymbol{E_k})$$

We use the following definition of [57] for the attributes $E_1, \ldots, E_k$: $J(\boldsymbol{E_k}) = J(E_1, \ldots, E_k)$ where:

$$J(\boldsymbol{E_k}) = \sum \ldots \sum Pr(E_1, \ldots, E_k) \frac{Pr(E_1, \ldots, E_k)}{Pr(E_1) \cdot \ldots \cdot Pr(E_k)}$$

Similarly, we have:

$$J(O, T, \boldsymbol{E_k}) = \sum \ldots \sum Pr(E_1, \ldots, E_k, O, T) \frac{Pr(E_1, \ldots E_k, O, T)}{Pr(E_1) \ldots Pr(E_k) Pr(O) Pr(T)}$$

$$J(X, \boldsymbol{E_k}) = \sum \ldots \sum Pr(E_1, \ldots . E_k, X) \frac{Pr(E_1, \ldots . E_k, O)}{Pr(E_1) \cdot \ldots Pr(E_k) Pr(X)}$$

We can derive:

$H(O; \boldsymbol{E_k}) + H(T; \boldsymbol{E_k}) - H(O; T; \boldsymbol{E_k}) - H(\boldsymbol{E_k}) =$

$$H(O) + \sum_{i=1}^{k} H(E_i) - J(O, \boldsymbol{E_k}) + H(T) + \sum_{i=1}^{k} H(E_i) - J(T, \boldsymbol{E_k})$$

$$- H(O) - H(T) - \sum_{i=1}^{k} H(E_i) + J(O, T, \boldsymbol{E_k}) - \sum_{i=1}^{k} H(E_i) + J(\boldsymbol{E_k}) =$$

$$J(O, T, \boldsymbol{E_k}) + J(\boldsymbol{E_k}) - J(O, \boldsymbol{E_k}) - J(T, \boldsymbol{E_k})$$

Thus we consider the following expression:

$$J(O, T, \boldsymbol{E_k}) + J(\boldsymbol{E_k}) - J(O, \boldsymbol{E_k}) - J(T, \boldsymbol{E_k}) \quad (6)$$

We argue that (6) is minimized when the k-th attribute minimizes the Min-CIM and Min-Redundancy criteria.

As stated in [57], the maximum of $J(O, \boldsymbol{E_k})$ is attained when all variables are maximally dependent. When $O, \boldsymbol{E_{k-1}}$ are fixed, this indicates that the attribute E_k should have the maximal dependency to O. In this case, we get that $J(O, T, \boldsymbol{E_k}) = J(T, \boldsymbol{E_k})$. Note that when the dependency of O or T in E_k increases, the conditional mutual information $I(O; T|\boldsymbol{E_k})$ decreases. This is the Min-CIM criterion.

Moreover, as noted in [57], the minimum of $J(\boldsymbol{E_k})$ is attained when the attributes $E_1, \ldots, E_k$ are independent of each other. As all the attributes $E_1, \ldots, E_{k-1}$ are fixed at this point, this pair-wise independence condition means that the mutual information between the attribute E_k and any other attribute E_i is minimized. This is the Min-Redundancy criterion.

Thus, we get that the overall expression in (6) is minimized (i.e., $J(O, \boldsymbol{E_k})$ is maximized, $J(O, \boldsymbol{E_k}, T) = J(T, \boldsymbol{E_k})$, and $J(\boldsymbol{E_k})$ is minimized) when we are minimizing the Min-CIM and Min-Redundancy criteria.

□

Proof of Lemma **??**. First, since $(O \perp\!\!\!\perp E_{k+1}|\boldsymbol{E_k})$ we have: $I(O, E_{k+1}|\boldsymbol{E_k}) = 0$. We get:

$I(O; T|\boldsymbol{E_k}) - I(O; T|\boldsymbol{E_k}, E_{k+1}) =$

$H(O|\boldsymbol{E_k}) - H(O|T, \boldsymbol{E_k}) - H(O|\boldsymbol{E_k}, E_{k+1}) + H(O|T, \boldsymbol{E_k}, E_{k+1})$

Since $H(O|\boldsymbol{E_k}) - H(O|E_{k+1}, \boldsymbol{E_k}) = H(O|\boldsymbol{E_k}) - H(O|\boldsymbol{E_k}) = 0$, we get:

$I(O; T|\boldsymbol{E_k}) - I(O; T|\boldsymbol{E_k}, E_{k+1}) =$

$$H(O|T, \boldsymbol{E_k}, E_{k+1}) - H(O|T, \boldsymbol{E_k}) \leq 0$$

For the last inequality we used the fact that for every three ransom variables X, Y, X: $H(X|Y) \leq H(X|Y, Z)$, since adding more conditions can only reduce the uncertainty of X.

We get that the numerator of the responsibility score of E_{k_1} is ≤ 0, and thus $Resp(E_{k+1}) \leq 0$ □

Proposition A.1. *The time complexity of the incremental MCIMR algorithm is $O(k|\mathcal{A}|)$.*

Proof. At each iteration, the MCIMR algorithm selects a new attribute to be added based on the condition defined in Equation (5). In the worst case, it examines all attributes in $\mathcal{A}$. Since it stops after at most k iterations, we get that the time complexity is $O(k|\mathcal{A}|)$. □

We next prove that logical dependencies can lead to a misleading conclusion that we found a confounding attribute.

Lemma A.2. *If for an attribute E we have: $FD : E \Rightarrow T$ then we get $I(O; T|E, C)$=0.*

Proof. If for an attribute E we have: $FD : E \Rightarrow T$ then we have $H(T|E) \approx 0$. We get: $I(O; T|E, C) = H(O|E, C) - H(O|T, E, C)$. But since T and E are dependent, we get: $H(O|T, E, C) \approx H(O|E, C)$ and thus $I(O; T|E, C)$=0. □

The lemma also holds for the case that the attribute E logically depends on the outcome O.

Relevance Test: Given a candidate attribute E, if $(O \perp\!\!\!\perp E|C)$ and $(O \perp\!\!\!\perp E|C, T)$ we get that $H(O|E, C) = H(O|C)$ and $H(O|T, E, C) = H(O|T, C)$. Thus:

$I(O; T|E, C) = H(O|E, C) - H(O|T, E, C) = H(O|C) - H(O|T, C) = I(O; T|C)$

That means that the individual explanation power of E is low, and thus it can be dropped as we assume E cannot be a part of the optimal explanation.

B EXPERIMENTS

Explanation quality. Next, we provide references supporting the explanations generated by MESA. These in-domain findings serve as "domain-expert" explanations.

SO Q1: It was shown in [5] that there is a correlation between the developers salary and countries' economies. In https://www.daxx.com/blog/development-trends/it-salaries-software-developer-trends, it was also shown that the countries with the highest salary for developers are countries with a relatively high HDI (e.g., the US, Switzerland, Denmark).

SO Q2 + Q3: It was mentioned in https://content.techgig.com/career-advice/what-is-the-average-salary-of-software-engineers-in-different-countries/articleshow/91121900.cms that countries that have a scarcity of software graduates tend to offer higher salaries than countries like India which produce hundred of thousands developers every year. This suggests that besides the economy of a country (resp., continent), the population size is also a factor that affects the average salary of developers.

Flights Q1: It was stated in [2] that weather is one of the top reasons for flights delay in the US.

Flights Q2-Q4: It was mentioned in https://www.bts.gov/topics/airlines-and-airports/understanding-reporting-causes-flight-delays-and-cancellations that besides weather conditions, main causes for flights delay in the US are heavy traffic volume, and air traffic control. Those two factors are highly correlated with population size. In bigger and more dense area, the air traffic increases.

Flights Q5: It was mentioned in https://www.bts.gov/topics/airlines-and-airports/understanding-reporting-causes-flight-delays-and-cancellations that a main cause of the delay of flights in the US is the airline's control (e.g., maintenance or crew problems).

Covid Q1: It was shown that there is a correlation between countries' economies and Covid-19 death rate [36, 69].

Covid Q2-Q3: It was stated in [**? ?**] that population density impact on COVID-19 mortality rate.

Forbes Q1: It was shown in https://www.theguardian.com/world/2019/sep/15/hollywoods-gender-pay-gap-revealed-male-stars-earn-1m-more-per-film-than-women that there is a gender pay gap for actors in Hollywood. Thus, it make sense that gender is a factor affecting the average salary of actors. It was also stated in https://www.gobankingrates.com/money/jobs/how-much-do-actors-make/ that actors get paid according to their experience, which is reflected in their net worth.

Forbes Q2: It was mentioned in https://climbtheladder.com/producer-salary/ that what affects directors and producers salary is their level of experience (which is reflected in the awards and net worth attributes).

Forbes Q3: It was stated in that very often professional athletes salaries are performance-based. The performance quality is reflected in the Cups and Draft Pick attributes (for tennis, basketball and football athletes, which are the majority of athletes in the Forbes dataset).

We next present additional experiments.

Impact of pruning. We next examine how useful were our pruning techniques. **Offline Pruning**. We found that our two offline pruning optimizations to be highly useful: On average, we dropped 41%, 59%, 45%, and 73% of the extracted attributes, in the SO, Flights, Covid-19, and Forbes dataset, resp. **Online Pruning**. At query time, we filter the extracted attributes using the logical dependency and the low relevance techniques. Not surprisingly, as most irrelevant attributes were already dropped in the offline phase, we dropped many fewer attributes at this phase. On average, we dropped 14%, 6%, 11% and 3% of the remaining attributes, in SO, Flights, Covid-19, and Forbes, resp.

Entity linker. Many of the missing values were caused by an unsuccessful matching of values from the table to their entities in the KG. For example, in SO, for some developers, their origin country is `Russian Federation`. However, the corresponding entity in DBpedia is called `Russia`. We thus failed to extract the properties of this country. In other cases, the values that appear in the tables were ambiguous, and thus we failed to match them to DBpedia entities. For example, in Forbes, one of the athletes is called `Ronaldo`. SpaCy entity linker could not decide whether to link this value to the entity `Ronaldo Luís Nazário de Lima` (Brazilian footballer) or to `Cristiano Ronaldo` (Portuguese footballer).

Multi-Hops. We examine the effect of extracting attributes following more than one hop in the KG. We report that in the vast majority of cases, MESA's explanations were unaffected. Namely, almost all attributes extracted from 2 or more hops were found to be irrelevant (and were pruned). In some cases, we found at most one more attribute that was included in the explanations. For example, in Forbes Q_1, an attribute representing the average budget of the films played by actors (attribute extracted from 2-hops) was included in the explanation. In all cases, no attributes from 3 or more hops was considered to be relevant. Further, since the number of candidate attributes was increased (in 145%, on average), running times were increased (by up to 15 seconds). This indicates that most of the relevant information can be found in the first hop. Future research will predict which paths in the KG may lead to relevant attributes.