Mining Explanations from Knowledge Graphs

Anonymous Author(s)
mail
Affiliation

ABSTRACT

When analyzing large datasets, analysts are often interested in the explanations for surprising or unexpected results that were produced by their queries. Previous approaches to results explanations have focused on generating explanations from the data accessed by the query. However, in many real-life scenarios, the explanations are not solely contained in the input table(s). In this work we are interested in generating explanations in terms of a set of *confounding variables that explain away spurious correlations* observed in query results. We present a system called MESA that automatically mines candidate attributes from a Knowledge Graph (KG) that can be seen as columns missing from the input table. It then employs an efficient algorithm to detect a bounded-size subset of attributes (from the input table and a KG) that explain away the unexpected correlations. 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 efficiency of MESA and its ability to handle large input datasets and a KG containing millions of tuples and facts.

ACM Reference Format:
Anonymous Author(s). 2023. Mining Explanations from Knowledge Graphs. In *Proceedings of the 2023 International Conference on Management of Data (SIGMOD '23), June 18–23, 2023, Seattle, WA, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/XXXXXX.XXXXXX

1 INTRODUCTION

When analyzing large datasets, users are often interested in explanations for surprising observations. Namely, they would like to find what might lie behind the surprising or unexpected results that were produced by their queries. Query results are often hard to interpret, especially for aggregate query answers [53]. Further, good explanations might be found outside the narrow query results that the user observes and the database being used [33]. Thus, there is a need to develop automated solutions that can explain query results to data analysts in a meaningful way, which goes beyond just the data accessed by the user query.

In this work, we focus on aggregate SQL queries (select-from-where-groupby) that are aggregating an *outcome variable* based on some group of interest indicated by a grouping or *exposure variable*. A major challenge that hinders the interpretation of such queries is *confounding bias* that can lead to spurious association between the exposure and an outcome variables and hence perplexing conclusions. Confounding bias is a systematic error due to the uneven or unbalanced distribution of a third variable, known as the *confounding variable* in the competing groups [49]. In this work we are interested in generating explanations in terms of a set of confounding variables (i.e., attributes) that **explain away the unexpected and spurious correlations observed in query results.** To illustrate, consider the following example.

Example 1.1. *Ann is a data analyst in the WHO organization who aims to collect insights and understanding of the coronavirus pandemic for improved policymaking. She examines Covid-Data [1], a dataset containing information describing different Covid-19-related facts in multiple cities worldwide (as of May 2020). It consists of the number of deaths-/recovered-/active-/new-cases and the number of deaths-/recovered-/active-/new- per-100-cases in each city. Ann evaluates the following query over this dataset:*

```
Q = 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 attribute is Country *and the outcome is* Deaths_per_100_cases*. Ann observes a big difference in the death rate among countries, and she is interested in finding a set of confounding variables (i.e., explanation) that explain away the relationship between* Country *and* Deaths_per_100_cases*. She uses statistical methods to see whether the observed correlation can be explained using attributes from Covid-data. She learns that the number of confirmed cases is correlated with* Deaths_per_100_cases*. However, this attribute alone is not enough to explain away the correlation. For example, she sees that while Germany had the fifth-most confirmed coronavirus cases in the world, it had only a fraction of the death toll seen in other countries.*[1] *She understands that other factors (that are not in this data) affect the relationship between death rate and country. For example, it was shown that as a country's success (defined by multiple variables including GDP*[2] *and HDI*[3]*) grows, the deaths rate decreases [29, 61]. However, such economic characteristics of countries are not available in the dataset and thus are not considered in the analysis. But such features are available on the web and could be mined from knowledge graphs.*

We propose to mine query results explanations from a Knowledge Graph (KG). KGs are an emerging type of knowledge representation that has gained much attention in recent years [10, 17, 18, 63]. KGs typically contain a very large amount of data. The sheer breadth of coverage that makes knowledge graphs potentially valuable is

SIGMOD '23, June 18–23, 2023, Seattle, WA, USA

ACM ISBN 978-1-4503-8343-1/21/06…$15.00
https://doi.org/10.1145/XXXXXX.XXXXXX

[1] As reported in https://tinyurl.com/2p9y8xz6.
[2] Gross domestic product (GDP) is the monetary value of all goods and services made within a country during a specific period.
[3] The Human Development Index (HDI) is a statistic composite index of life expectancy, education, and per capita income indicators, which is used to rank countries.

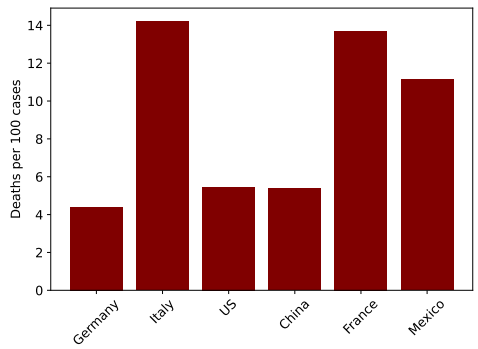

Figure 1: Visualization of the results of the query Q.

also what makes the need to automate the process of mining relevant confounding variables. There are multiple general-purpose (e.g., Wikidata [9], DBpedia [5], Yago [50]) or domain-specific (e.g., [44, 54]) KGs that act as central storage for data extracted from multiple sources (e.g., Wikipedia). We argue that such valuable data could be utilized for explaining away unexpected correlations observed in user queries.

To this end, we present an efficient algorithm that finds a bounded-size subset of attributes (mined from a KG and the dataset) that explain away spurious correlations observed in user queries. Those attributes can assist analysts in interpreting and understating the results. This algorithm is embodied in a system called MESA (for **M**ining **E**xplanation**S** from knowledge gr**A**phs), a system that automatically mines candidate attributes form a KG.

Example 1.2. *Ann uses MESA to search for an explanation for her query. MESA mines all available features about countries that appear in Covid-Data from DBpedia [5]. She finds that, besides the number of confirmed cases, the countries' HDI and GDP are two factors that have a great effect on death rate [29, 61]. She is pleased because she found a plausible real-world explanation for her query results. She learns that the number of confirmed cases, HDI, and GDP are uncontrolled confounding attributes. She sees that the death rate in countries with a similar number of confirmed cases, HDI, and GDP is also similar.*

Previous work provides explanations on how a query result was derived by analyzing the query provenance [39, 40, 43]. Those methods generate tuple-level explanations and thus inapplicable for unearthing correlations between attributes. Another type of explanation is a set of patterns that are shared by one (group of) tuple in the results but not by another (group of) tuple [22, 33, 51, 52, 62]. However, those works as well do not account for correlations among attributes. The authors of [53] presented a system that provides explanations based on causal analysis, measured by correlation among attributes. However, this system only considers the input dataset, and it cannot efficiently handle a large amount of candidate confounding attributes. We share with [33] the motivation for considering explanations that are not solely drawn from the input table. They presented a system that generates insightful explanations based on contextual information mined from tables related to the table accessed by the query. However, their explanations are a set of patterns that are unevenly distributed among groups in the query results. Thus, this system is inapplicable for explaining away correlations observed in queries. (See discussion in Section 6).

Challenges. There are several challenges that need to be overcome to facilitate an efficient system producing high-quality explanations. The first challenge is how to generate high-quality explanations for query results that are interpretable and useful (challenge **C1**). The second challenge is how to extract data from a KG. This required a principled way of handling missing data (challenge **C2**). Last, the search space of possible attribute subsets is intractable. Further, the search for the optimal attribute set involves estimating high dimensional partial correlations. It has been shown that it is hard to get an accurate estimation for the multivariate partial correlation [48] (challenge **C3**).

Given a query Q we denote by T and O the exposure and outcome attributes in Q, resp. To address **C1**, we present the Correlation-Explanation problem that seeks a subset of attributes (extracted from a KG or the input table), referred to as *the explanation for* Q, that explain away the correlation between T and O. Namely, when conditioning on these attributes, T and O become independent. We note that a "good" explanation is *concise*, as people prefer the simplest possible explanation [35, 45], and it is *minimal* in size [33, 52]. We thus formalize Correlation-Explanation as the problem of finding a minimal-size set of attributes that minimizes the partial correlation between T and O. Further, MESA enables analysts to learn the individual responsibility of selected attributes, and to automatically identify unexplained data subgroups correspond to refinements of the query in which a different explanation is required and may require for them. (Section 2).

To address **C2**, we present a method for knowledge extraction. Given an input table $\mathcal{T}$ and a KG, we extract attributes from the KG that can be linked to $\mathcal{T}$. Namely, the extracted attributes can be seen as columns missing from $\mathcal{T}$. This process utilizes off-the-shelf Named Entity Disambiguation (NED) tools [46] to match any non-numerical value in $\mathcal{T}$ to a unique entity in KG. However, the extracted attributes may contain missing values. Restricting the analysis only to tuples with no missing values may induce *selection bias*. Inverse probability weighting (IPW) is a commonly used method to correct this bias [55]. We provide sufficient conditions to detect selection bias and explain how IPW can be employed in our setting (Section 3).

To address **C3**, we propose the MCIMR algorithm. This algorithm avoids estimating high dimensional conditional mutual information (a common measure for partial correlation) and does not require iterating over all possible attribute subsets to find the optimal explanation. It selects attributes based on minimal-conditional-mutual-information and minimal-redundancy criteria. MCIMR is based on the first stage of a well-studied feature selection algorithm called MRMR [48]. We show how the first stage of the MRMR algorithm can be adjusted to our setting, yielding a polynomial-time 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 heuristic pruning techniques to speed up computation time. (Section 4).

We conducted an experimental study based on four real-life datasets that evaluate the quality and efficiency of MESA in multiple scenarios. We ran a user study consisting of 12 subjects to evaluate the quality of explanations produced by MESA compared with five approaches. We show that the explanations generated by MESA are

almost as good as those of a naive approach that iterates over all attribute subsets, and are much better than those of the competitors. We then study the effect of different parameters on the performance. Our experimental results indicate the effectiveness of our algorithm in finding explanations for queries evaluated on datasets containing more than 5M tuples in less than 10s (Section 5)

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

2 MODEL AND PROBLEM FORMULATION

2.1 Data Model and Assumptions

We operate on a dataset consisting of a single relational table database $\mathcal{T}$. 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 $\boldsymbol{A}\subseteq\mathcal{A}$. In this work we focus on simple single-block SQL queries with a single aggregate function [33, 53], as shown in Listing 1. We restrict the queries to group-by-average queries. We do not consider more complex queries (e.g., nested sub-queries), and we also do not consider aggregate operators other than average.

Listing 1: A group-by-average query.

```
Q = SELECT T, avg(O)
FROM D
WHERE C
GROUP BY T
```

We denote by O the outcome attribute of the query Q and by T the exposure attribute used for grouping. To simplify the exposition, we assume a single grouping attribute. However, our results can be naturally generalized for cases with multiple grouping attributes. We call the condition C the context for the query Q. We interpret the query as follows: given the context, we want to explain the difference among $avg(O)$ for each $T{=}t_i$, where $t_i{\in}Dom(T)$.

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

Example 2.1. *Stack Overflow (SO) dataset contains information about people who code around the world, such as their age, gender, income, and country. Consider the following query, denoted as Q_{so}:*

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

The query outcome O is Salary, its exposure T is Country and the context C is Continent = Europe. We aim to explain the difference among the average salary of developers from each country in Europe. While some attributes from the dataset may partially explain this difference (e.g., Gender, DevType), other important attributes that can cast light on this difference cannot be found in the SO dataset, but could be found in external sources.

We make the following assumption, standard in statistics. The database $\mathcal{T}$ is a uniform sample from a large population (e.g., all developers, all countries in the worlds, etc.), obtained according to some unknown distribution $Pr(A)$. Then, the query Q represents an empirical estimate $E[O|T = t_i, C]$, for each $t_i{\in}Dom(T)$. We expect the reader is familiar with the notions of entropy $H(X)$, joint and conditional entropy $H(Y;X), H(X|Y)$, and conditional mutual information $I(X;Y|Z)$ associated to the probability distribution Pr.

External Knowledge. The system has access to a Knowledge Graph (KG), denoted as $\mathcal{G}$. From this KG, we extract a set of attributes $\mathcal{E}$ representing additional properties of entities from $\mathcal{T}$. All attributes in $\mathcal{E}$ are not attributes in $\mathcal{A}$ (the attributes of the table $\mathcal{T}$). Continuing with our running example, $\mathcal{E}$ could be a set of properties of the countries extracted from some KG (e.g., DBpedia [5], Yago [50]), such as the density, population size, HDI, etc. See Section 3.1 for more details about the extraction process. We can potentially join $\mathcal{E}$ and $\mathcal{T}$ together, by linking values from $\mathcal{T}$ with their corresponding entities in $\mathcal{G}$ that were used to extract the attributes in $\mathcal{E}$. However, $\mathcal{E}$ may contain many attributes, most of them are most likely irrelevant for the user interested in explaining the results of a query Q. Our goal is therefore to identify a bounded-size subset of attributes $\boldsymbol{E}$ from $\mathcal{E}\cup\mathcal{A}$ that would *explain the relationship between the outcome and exposure attributes* of Q.

Example 2.2. *Let $\mathcal{E}$ denote the set of all attributes extracted from a KG for all the countries that appear in SO. $\mathcal{E}$ contains many attributes, such as the country's Leader Name, Population Size, GDP, etc. (those are real examples of attributes extracted from DBpedia [5]). Some attributes can potentially explain the differences in salary among countries (e.g., GDP, HDI), while others are irrelevant for Q_{so} (e.g., Length of coastline, Year snow Inch).*

2.2 Problem Formulation

Given a query Q, we assume the user observed an unexpected correlation between the exposure T and the outcome O attributes she would like to investigate. In this work, we assume that 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 may preclude finding a true effect; it may lead to an inaccurate estimate (underestimate or overestimate) of the true association between T and O. Therefore, our goal is to discover the confounding variables, to control their impact on the observed correlation between T and O.

Let $\mathcal{D}$ denote $\mathcal{E}\cup\mathcal{A}\setminus\{O,T\}$, referred to as the candidate attributes. We denote by $\boldsymbol{E_k}\subseteq\mathcal{D}$ a k-size set of attributes, and by $E_i{\in}\mathcal{D}$ a single attribute. We assume that $\mathcal{D}$ contains confounding attributes that affect both T and O. We aim to find a set of attributes from $\mathcal{D}$ that control the correlation between O and T. Namely, when conditioning on this attribute set, the correlation between O and T is diminished. We call such a set *the explanation of the query results*.

Example 2.3. *It is very likely that countries' economic features (such as GDP, Gini Coefficient[4], HDI, HDI Rank) affect developers' salaries. To unearth the unexpected association between Country and Salary, one must measure the correlation while controlling such attributes. This will allow users to understand which factors affect the differences in developers' salaries in different countries. Intuitively, we expect the average developers' salaries to be similar in countries with similar economic characteristics. Our goal is therefore to a find*

[4]The Gini index is a measure of statistical dispersion intended to represent the income inequality within a nation or a social group.

a set of attributes in which when conditioning on them, there is no correlation between Country *and* Salary.

We note the following properties of a "good" explanation: **Explainability** —to what extent the solution explains the correlation between T and O; **Conciseness** —people prefer the simplest possible explanation for a phenomenon, if that explanation is equal to other possible explanations [35, 45]. Previous work has pointed out on the importance of a minimal-size explanation [52]; **No redundancy** —each attribute in the explanation should be uncorrelated with other attributes in the explanation [33, 52]. We thus formalize our problem as the problem of finding a minimal-size set of attributes $\boldsymbol{E}\subseteq\mathcal{D}$ that explain away the correlation between T and O. Explaining away the observed correlation addresses explainability. Conciseness is addressed by the minimal cardinality requirement. No redundancy is derived from the conciseness requirement in our setting. Two highly correlated attributes are not likely to appear in a minimal-size set of attributes that, when conditioned on it cause T and O to become independent.

Ideally, we would like to find a minimal-size set of attributes $\boldsymbol{E}\subseteq\mathcal{D}$ s.t: $(O\perp\!\!\!\perp T|\boldsymbol{E},C)$. However, in practice, we may not find these perfect explanations, hence we are looking for a minimal-size set of attributes that *minimize 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 approach to measure the partial correlation is multiple linear regression, which is sensitive only to linear relationship. Here we use *conditional mutual information*, a simple measure of the mutual dependence between two variables, given the value of a third. Note that $(O\perp\!\!\!\perp T|\boldsymbol{E},C)$ holds iff $I(O;T|\boldsymbol{E},C)$=0. Thus, we formalize the Correlation-Explanation problem as follows:

Definition 2.1 (**Correlation-Explanation**). *Given a set of candidate attributes $\mathcal{D}$ and a query Q, find a minimal-size set of attributes $\boldsymbol{E}\subseteq\mathcal{D}$ that minimizes $I(O;T|\boldsymbol{E},C)$.*

Example 2.4. *Among other attributes, we extracted from the KG the* Gini coefficient *(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, conditioning on Gini coefficient, the correlation between* Country *and* Salary *decreases. That is, in countries with a similar Gini coefficient, 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[5], 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 gender of the developers also cannot not explain the correlation in Q_{so}.*

To assist analysts in interpreting and understating the results, we enable users to learn the individual *responsibility* of each selected attribute. Given an explanation $\boldsymbol{E}\subseteq\mathcal{D}$, we rank the attributes in $\boldsymbol{E}$ in terms of their responsibilities as follows:

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

Definition 2.2 (Degree of responsibility). *Given a query Q and set of attributes $\boldsymbol{E}$, the degree of responsibility of the 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 degree of responsibility of an attribute $E_i\in\boldsymbol{E}$ 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 degree of responsibility of E_i is positive if E_i contributes to the explanation. Thus, a negative responsibility score indicates that adding E_i only harms the explanation (it happens since E_i has a negative interaction information with O and T). The higher the responsibility score of an attribute, the greater is its individual explanation power.

Example 2.5. *Recall that E_1= Gini coefficient, E_2= Density, E_3= HDI, and E_4= Gender. Let $\boldsymbol{E}$=$\{E_1,E_2\}$. According to our data we have: $I(O;T|C,E_2)$=1.51. We get: $Resp(E_1)$=0.53, and $Resp(E_2)$=0.46. Here, the sum of the responsibility scores is < 1, since $I(O;T|C,\boldsymbol{E})\neq$0. Now consider the attribute Hobby (E_5), indicating 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. Attributes with low responsibility scores, that is, their individual explanation power is low, are of no interest to the user. Further, XOR-like explanations (in which the explanation power of each individual attribute is low, but their combination makes a good explanation) are very rare. Of course, such multivariate explanations are technically possible, but we did not find any such useful explanations in our experiments. Even if they do exist, they are hard to understand and, therefore, may not be comprehensible to a user. Since finding such XOR-like explanations pose a series of further technical challenges, in this work, we assume that the optimal explanation does not contain multivariate associations among attributes, which are individually unimportant but become important in the context of others. A similar assumption is often made in feature selection algorithms [14, 59, 60], where they assume that the optimal feature set does not contain multivariate associations among variables, which are individually irrelevant to the target class but become relevant in the presence of others.

3 ATTRIBUTES EXTRACTION

We describe how the candidate set of attributes $\mathcal{E}$ is extracted from the KG $\mathcal{G}$. We then present a novel method to handle missing values.

3.1 Extracting the Candidate Attributes

The first step is to map values that appear in the table $\mathcal{T}$ to their corresponding unique entities in the KG $\mathcal{G}$. This task is often referred to as the Named Entity Disambiguation (NED) problem [46]. We can use any off-the-shelf NED algorithm (e.g., [46, 64]) to match any non-numerical value in $\mathcal{T}$ to an entity in $\mathcal{G}$. At the next step, given an entity identified in $\mathcal{T}$, we extract all of its available properties from the KG and store them in a dictionary. We then organize all the extracted dictionaries into a table, setting a null value to all properties whose values were missing for a given entity. This

China

Property	Object
HDI	0.76
GDP Rank	2
Leader	X. Jinping

US

Property	Object
HDI	0.92
GDP Rank	1
Gini	48.5
Leader	J. Biden

India

Property	Object
HDI Rank	131
HDI	0.64
GDP Rank	6
Gini	35.7
Leader	R.N Kovind

$\mathcal{E}$

Country	HDI Rank	HDI	GDP Rank	Gini	Leader
China	-	0.76	2	-	X. Jinping
US	-	0.92	1	48.5	J. Biden
India	131	0.64	6	35.7	R.N. Kovind

Table 1: Example dictionaries extracted for three entities (countries) and the corresponding attribute set $\mathcal{E}$.

process is equivalent to building the *universal relation* [23, 38] out of all of the entity specific relations that were derived from the KG.

Example 3.1. *The set of entities extracted from the SO dataset consists of all countries names that appear in SO. Table 1 depicts example of three dictionaries extracted for countries. Observe that the dictionaries are of different sizes and include different attributes. The corresponding table $\mathcal{E}$ is given in the bottom part of Table 1.*

Note that one of the strengths of a KG is that most of the attributes are already reconciled. Namely, we will not have to match, e.g., different versions of HDI across different entities.

Multi-hops. To extract more attributes and potentially improve the explanations, one may "follow" links in $\mathcal{G}$. Namely, extract also all the 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. In cases where a property leads to multiple entities, one may aggregate the values using different aggregation functions, such as average or mode.

Example 3.2. *W can extract all properties of the countries' leaders, such as their age and gender. In this case, we add 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 (which are entities in $\mathcal{G}$ as well). Each ethnic group has the property Population size. One may add the property of Avg Population size of Ethnic-Group to $\mathcal{E}$ by averaging the population size of the ethnic groups.*

3.2 Handling Missing Data

As mentioned, the extracted attributes may contain missing values. The simplest approach to dealing with missing values is to restrict the analysis to complete cases. Namely, discard cases that have missing values for some attribute $E_i \in \mathcal{E}$. However, this can induce *selection bias* if the excluded tuples are systematically different from those included. For example, if the HDI values of only the countries with a very high HDI are missing, restricting the analysis only to complete cases may lead to inaccurate and misleading explanations.

Existing Approaches. Handling missing data is an enduring problem for many systems [20]. We do not aim to make a novel intellectual contribution in this area, but rather to choose and adapt an existing method for our problem. One common approach is to impute missing values. Previous work showed that data imputation is unlikely to cause substantial bias if few data are missing, but bias may increase as the number of missing data increases [55]. Another common approach is Multiple Imputations (MI) [47]. While MI could be useful in supervised learning as long as it leads to models with an acceptable level of accuracy, one must use it with care when generating explanations. This is because inaccurate imputation can lead to highly misleading explanations. Also, MI makes a missing-at-random assumption [20], 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 [55]. In IPW, we restrict the attention only to complete cases, but more weight is given to some complete cases than others. We next explain how to employ this approach in our setting.

For simplicity of presentation, we assume that the table $\mathcal{T}$ and the attributes in $\mathcal{E}$ 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 $E \in \mathcal{E}$, let R_E denote a selection attribute that indicates whether the values of E for the i-th tuple in the results of Q is missing. Namely $R_E[i]=1$ 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 an attribute $E \in \mathcal{E}$ (e.g., $P(O|E)$, $P(E)$) is recoverable if: $P(X)=P(X|R_E=1)$. If both E, E' contain missing values, then $P(E|E')$ is recoverable if: $P(E|E')=P(E|E', R_E=1, R_{E'}=1)$.

$P(O|E,C)$ is recoverable if the completeness of a case is independent of O given E and the context C. $P(O|T,E,C)$ is recoverable if the completeness of a case is independent of O given E, T, and C. Namely, the complete cases are a representative sub-sample of the original sample, and each complete case is a random sample from the population of individuals with the same E and T values.

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

$P(E)$ and $P(E|E')$ are recoverable if the completeness of a case is independent of E, and remains independent of E given E'. In this case, we get that $I(E;E')$ is recoverable.

Corollary 3.2. *For $E_i, E_j \in \mathcal{E}$, 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 $I(E_i;E_j|R_{E_i}=1,R_{E_j}=1)=I(E_i;E_j)$.*

Weights Definition. 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 the inverse of the probability that X is a complete case. The weight $W(X)$ of an event X is defined as: $P(X)=P(X|R_E=1)W(X)$. We get: $W(X)=P(R_E=1)/P(R_E=1|X)$. However, since E contains missing values, $P(X)$ is unknown, and thus we can not compute $W(X)$. We thus estimate $P(X)$. Commonly, a logistic regression model is fitted [27, 30]. Data available for this are the values of the attributes in $\mathcal{A}$. We therefore employ a logistic regression (at pre-processing) to predict missing values for E to estimate $P(X)$. We note that although, as in MI, we predict missing values, in the IPW approach, we only use those predicted values for weights computation and not for the entire analysis.

4 ALGORITHMS

In this section we present the MCIMR algorithm for the Correlation-Explanation problem, and our proposed optimizations that aim to reduce its execution time. We then present a procedure to identify unexplained subgroups, correspond to refinements of the input query Q in which a different explanation is required for them.

4.1 The MCIMR Algorithm

The Correlation-Explanation problem is closely related to the Feature Selection (FS) problem (see Section 6). Next, we present our algorithm, named the Minimal Conditional mutual Information Minimal Redundancy (MCIMR) algorithm for the Correlation-Explanation problem. This algorithm is based on the first stage of a well-studied FS algorithm, called the MRMR algorithm [48]. The MRMR algorithm [48] is based on the Max Relevance Min Redundancy criteria. The main difference in MCIMR is that instead of the Max-Relevance criterion, we use a Min-Conditional-Mutual-Information criterion, adjusting the objective function to a minimum. We note that what makes this work different from ours is the details of the correctness proof, which is critical for obtaining the optimal output guarantee for our case.

We show how the first stage of MRMR can be adjusted for our setting, yielding a polynomial-time algorithm that finds the optimal k-size explanation where k is given. We then define a stopping criterion, allowing our algorithm to stop when no further improvement is found. We start by assuming that k, the size of the optimal explanation, is known. We, later on, remove this assumption.

When k equals 1, the optimal solution to Correlation-Explanation is the attribute $E \in \mathcal{D}$ that minimizes $I(O;T|C,E)$. When $k \geq 1$, a simple incremental solution is to add one attribute from $\mathcal{D}$ 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,

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

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

Previous work has shown that it is difficult to get an accurate estimation for multivariate mutual information [48] (as in Equation (1)), as it requires materializing the joint probability of multiple attributes. Instead, our algorithm calculates only bivariate probabilities, which is much easier and more accurate. We thus incrementally select attributes based on Minimal-Conditional-mutual-Information (MCI) and Minimal-Redundancy (MR) criteria.

The idea behind MCI is to search a k-size set of attributes $\boldsymbol{E_k} \subseteq \mathcal{D}$ that satisfies Equation 2, which approximates Equation 1 with the mean value of all Conditional mutual Information (CI) values between the individual attributes in $\boldsymbol{E_k}$ and O and T:

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

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

As in the MRMR algorithm, it is likely that attributes selected according to MCI are redundant, i.e., the dependency among attributes could be large. Therefore, the following Minimal Redundancy (MR) condition is added to select mutually exclusive attributes:

$$\boldsymbol{E_k} = argmin_{\boldsymbol{E_k} \subseteq \mathcal{D}} R(\boldsymbol{E_k}) \quad (3)$$

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

Algorithm 1: The MCIMR Algorithm.

input : A number k, a set of attributes $\mathcal{D}$, the outcome, treatment attributes O and T, and the context C
output : An explanation $\boldsymbol{E}$.

```
MCIMR(k, D, O, T, C):
E ← ∅.
for i ∈ [1, k] do
    E_i ← NextBestAtt (O, T, C, E, D)
    if O ⫫ E_i | E                    // The responsibility test for E_i
     then
        return E
    E ← E ⋃ {E_i}
return E
NextBestAtt (O, T, C, E, D):
E* ←None, v ← ∞
foreach E ∈ D \ E do
    /* Weights are added if selection bias was detected      */
    v_1 ← I(O; T|C, E)                          // Min CI computation
    v_2 ← 0
    foreach E' ∈ E do
        /* Weights are added if selection bias was detected  */
        v_2 ← v_2 + I(E; E')                    // Min redundancy computation
    if v_1 + v_2/|E| < v then
        E* ← E, v ← v_1 + v_2/|E|
return E*
```

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

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

We present an incremental algorithm that finds the optimal k-size attribute set defined by Equation 4. It is defined as follows. In the k-th iteration we already have the set $\boldsymbol{E_{k-1}}$ with $k-1$ attributes. The goal is to select the k-th attribute to be added. This is done by selecting the attribute that minimizes the following condition:

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

We can prove that the combination of the MCI and MR criteria is equivalent to Equation 1.

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

Stopping Criteria. The MCIMR algorithm assumes that the size of the explanation k is given. However, given two consecutive solutions of sizes k and k+1, we do not know which one provides a better explanation. Namely, 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 in Section 2.2, we assume that attributes in which their marginal explanation power is small are of no interest to the user. We thus stop the algorithm after the first iteration in which the responsibility score 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 attributes selected so far $\boldsymbol{E_k}$, this test verifies if the responsibility score 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 [53].

The full MCIMR algorithm is depicted in Algorithm 1. First, the algorithm 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 the selected attribute. If the responsibility of this attribute is $\approx$0, the algorithm terminates and returns the solution obtained until this point (lines 5-7). Otherwise, the algorithm terminates after k iterations (line 9).

Given the set of attributes 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), as well as their average pair-wise mutual information with all attributes selected in the previous steps (lines 16-18).

For simplicity, we omitted the parts dedicated to handling missing data from presentation. In our implementation, before executing lines 14 and 18, we check whether or not weights are needed to be added and adjust the computation accordingly.

The time complexity of this algorithm is $O(k|\mathcal{D}|)$. Nevertheless, note that the size of $\mathcal{D}$ is potentially very large because of the KG.

4.2 Optimizations

We propose several simple optimizations that could reduce the size of $\mathcal{D}$ and thereby reduce execution times. These optimizations may cause the MCMIR algorithm to overlook some important attributes. However, our experiments show that the solutions are almost unaffected in practice while running times significantly improve. 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 and can be discarded [33, 53]. **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 CountryCode, wikiID, that have high entropy and (almost) a unique value for each tuple (as was done in [53]).

Online pruning. **Logical Dependencies**: Logical dependencies (such as keys or functional 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. If for an attribute E we have: $FD: E \Rightarrow T$ then we get: $H(T|E) \approx 0$. We have: $I(O;T|E,C) = H(O|E,C) - H(O|T,E,C)$. But since T and E are dependent, we get: $H(O|T,E,C) \approx H(O|E,C)$ and thus $I(O;T|E,C)=0$. We thus discard all attributes E s.t. $H(T|E)=\epsilon$ and $H(E|T)=\epsilon$ for $\epsilon \approx 0$ (resp., for O) as was done in [53]. These tests correspond to approximate functional dependencies, such as CountryCode $\Rightarrow$ Country. **Low Relevance**: As mentioned, we assume 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. Specifically, 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.

Remarks. We conclude with two remarks. First, another simple yet effective optimization is to cache pair-wise mutual information values. We note that pair-wise mutual information calculations (especially for computing the MR criteria) are shared among many queries. We thus cache these values to avoid redundant computations. Similarly, we also cache weights to reduce execution time.

Second, we note that a possible optimization to be applied to improve running times is to cluster together attributes that are highly correlated, such as HDI and HDI Rank. This will reduce the redundancy among the candidate attributes (as was done in [33]). However, according to our experiments, this optimization was not useful because of the following reasons: (1) It could only be done after the query arrives, namely after we are done filtering. Since the clustering process took longer than running our algorithm on all attributes, we conclude this is not useful. (2) We found that attributes clustered together were not necessarily semantically related. Thus, it is unclear which attribute should represent each cluster.

4.3 Identifying Unexplained Subgroups

In Section 4.1 we have presented our algorithm for finding optimal explanations. While the explanations found by the algorithm are optimal considering the whole data, they may be insufficient for some parts in the data as we demonstrate next.

Example 4.1. *Consider the following query:*

```
Q = SELECT Country, avg(Salary)
FROM SO
GROUP BY Country
```

The explanation found by MESA is $\boldsymbol{E}$=*{HDI, Gini}. As mentioned, the HDI of all countries in Europe is similar. Thus, for of all countries in Europe, it is likely that* $\boldsymbol{E}$ *is not a satisfactory explanation.*

To this end, we present a procedure to identify unexplained data subgroups. Intuitively, each subgroup corresponds to refinements of the query Q in which a different explanation is required and may require further exploration. For simplicity, numerical attribute values are assumed to be binned into a fixed number of bins. Data groups are defined by a set of attribute-value assignments and correspond to refinement 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. $C' \subset C$. The goal is then to find the largest data groups s.t. $\boldsymbol{E}$ can not serve as their explanation. More formally, we are inserted in the top-k data groups (in terms of their size in the data), each correspond to a context C' (a refinement of C), s.t. $I(O;T|C',\boldsymbol{E}) > \tau$ for some threshold τ. Intuitively, τ can be set based on the initial value $I(O;T|C,\boldsymbol{E})$.

Example 4.2. *Continuing with Example 4.1, we refine the query Q by adding a WHERE clause selecting only countries in Europe (*C' = {Continent = Europe}*). Let* Q_{EU} *denote this refinement query of Q. Indeed, we get:* $I(O;T|C',\boldsymbol{E})$=2.13. *As mentioned in Example* 2.4, *the optimal explanation for* Q_{EU} *is* {Gini, Density}.

A naive algorithm for this problem would traverse over all possible contexts C', check if $I(O;T|C',\boldsymbol{E}) > \tau$ for each corresponding

Algorithm 2: Top-k unexplained data groups.

input : A number k, a set of attributes $\mathcal{D}$, 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 $\mathcal{R} \leftarrow \emptyset$
2 $MaxHeap \leftarrow$ GenChildren(C)
3 **while** $|\mathcal{R}| < k$ *or* $MaxHeap.isEmpty()$ **do**
4 $C' \leftarrow MaxHeap.exatractMax()$
5 **if** $I(O;T|C',\boldsymbol{E}) > \tau$ **then**
6 update($\mathcal{R}, C'$)
7 **else**
8 **for** $C'' \in$ GenChildren(C') **do**
9 $MaxHeap.insert(C'')$
10 **return** $\mathcal{R}$

refinement query, and choose the largest data groups for which $\boldsymbol{E}$ is not a satisfactory explanation. We next propose a more efficient algorithm for this problem, exploiting the notion of pattern graph traversal presented in [11]. Intuitively, the set of all possible context refinements can be represented as a graph (similar to the pattern graph presented in [11]), where nodes correspond to refinements (set of conditions) and there is an edge between C and C' if C' can be obtained from C by adding a single value assignment to the condition. In this case, we say that C (C') is a parent (child) of C' (C). As shown in [11], the graph can be traversed in a top-down fashion, while generating each node at most once.

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 the context 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 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' in the graph 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).

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? $Q2$: What is the efficiency of the proposed algorithm and the optimization techniques? $Q3$: How useful are our proposed extensions?

5.1 Experimental Setup

Our code was implemented in Python 3.7 and is available at [8]. We used DBPedia [5] for attributes extraction, the SpaCy Entity Linker [7] for named entity disambiguation, and the Pyitlib Python library [6] for information-theoretic computations. The experiments were executed on a PC with a 4.8GHz CPU, and 16GB memory.

Table 2: Examined Datasets.

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

Datasets. We examine four publicly available datasets, as depicted in Table 2. **(1) SO**: Stack Overfow's (SO) annual developer survey is the largest survey of people who code around the world [4]. It has more than 47K records containing information about the developers' such as their age, gender, ethnicity, income, and country. **(2) Covid-19**: This dataset [1] includes information about the number of confirmed, death, and recovered cases in 2020 across the globe. It contains information about 188 countries. **(3) Flights Delay**: This dataset [2] contains transportation statistics of over 5.8M domestic flights operated by large air carriers in the USA in 2015. **(4) Forbes**: This dataset [3] contains annual earning information of 1647 celebrities since 2005. The data was gathered from the "Celebrity 100" lists of Forbes magazine. It contains the celebrities' annual pay, and category (e.g., Actors, Producers).

Attribute Extraction. Using the entity linker, for each dataset, we match its values (that could be linked to entities) to their corresponding DBPedia entities. We then extracted all properties from the KG for each entity. The attributes in which their values were linked to entities and the number of extracted attributes in each dataset are given in Table 2. By default, we follow 1-hop in the KG.

Baseline Algorithms. We compare MESA against the following baselines: **(1) Brute-Force**: The optimal solution according to problem definition 2.1. This algorithm implements an exhaustive search over all subsets of attributes. To make it feasible, we run it after employing the pruning optimizations described in Section 4.2. **(2) Top-K**: This naive algorithm ranks the candidate attributes according to their individual explanation power (i.e., their conditional mutual information with T and O). The lower the score, the higher the rank is. **(3) Linear Regression** (LR): This baseline employs the Ordinary Least Squares (OLS) method to estimate the coefficients of a linear regression that describes the relationship between the outcome and the candidate attributes. The explanations are defined as the top k attributes with the highest coefficients (and their p value is $< .05$). **(4) HypDB** [53]: This is a state-of-the-art system for bias detection in queries. HypDB[6] uses causal analysis to generate explanations to query results. The explanations are defined as the top-k attributes with the highest responsibility scores, as defined in [53]. *This baseline serves as a representative example for a causal-analysis-based approach.* **(5) MESA$^-$**: Last, to examine how pruning affects the explanation, we examine the explanation generated by MESA without the pruning optimizations.

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

5.2 Quality Evaluation (Q1)

We present a user study consisting of multiple queries and their generated explanations produced by different algorithms. The goal of

[6] We use the implementation available at https://github.com/CoreyCole/HypDB.

Table 3: 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 Sqmi, Year Snow Inch, 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 Coefficient, Population Census, Confirmed cases**	**Gini Rank, Density, Confirmed cases**	**Gini Coefficient, Population Census, Confirmed cases**	Gini Rank, Gini Coefficient, 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 4: Average explanation scores according to the subjects (the higher the better).

Baseline	Average Score	Average Variance
Brute-Force	4.2	0.9
MESA-	4.1	1.3
MESA	3.9	1.1
HypDB	3.3	1.4
Top-K	2.5	1.5
LR	1.9	1.3

this user study is twofold: we aim to validate our problem definition and evaluate the explanation quality of each method.

We consider 14 queries (in which we have manually picked and their explanations are likely to be found in a KG), as shown in Table 3. For each query, we recruited 12 subjects on Amazon Mechanical Turk[7]. Subjects were asked to rank each explanation (shown together with its corresponding query) on a scale of 1–5, where 1 indicates that the explanation does not make sense and 5 indicates that the explanation is highly convincing.

As mentioned in [53], the worst-case time complexity of HypDB is exponential in the size of the number of candidate attributes ($|\mathcal{D}|$). We run it over all attributes in $\mathcal{D}$ (after pruning) and report that it never terminates within 30 minutes. 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 our pruning optimizations, we omitted candidate attributes uniformly at random, ensuring that $|\mathcal{D}| \leq 50$. We only report the results of Brute-Force for the small Covid-19 and Forbes datasets, as it was infeasible to compute the explanations for the larger datasets. We do not randomly drop attributes for computational efficiency here because Brute-Force is intended to be an optimal benchmark against which the others are judged. The explanations generated by different methods are given in Table 3, and the average scores given by the subjects for each algorithm are depicted in Table 4.

First, the subjects found the explanations generated by Brute-Force, MESA$^-$, and MESA to be the most convincing. *This positive result supports our mathematical definition of what constitutes a good explanation (Definition 2.1).* In all cases where the results of Brute-Force and MESA are different, it happens because MESA drops attributes with insignificant responsibility scores (according to the responsibility test). For example, in Forbes Q_1, MESA dropped Age, as its responsibility score is ≈ 0. The low difference between the results of MESA$^-$ and MESA indicates that the pruning has little effect on the quality of explanations. Namely, *MESA is able to execute efficiently without compromising on explanation quality.*

The next best competitor is HypDB. This is not surprising as HypDB finds causal relationships among attributes, estimated by mutual information. However, as mentioned, the main disadvantage of HypDB is its ability to scale for large number of confounding attributes. In cases where HypDB generated explanations that were considered not convincing by the subjects, it was mainly because the most important attributes were dropped (since we sampled only a fraction of the attributes to enable feasible execution times). *This demonstrates the limitation of causal-analysis-based solutions in handling large search spaces.* Not surprisingly, the explanations generated by Top-K and LR were considered to be less convincing. For Top-K, this demonstrates that "the k best explanations are not the best k explanations" [48]. This is substantially because it

[7] Amazon Mechanical Turk: https://www.mturk.com/

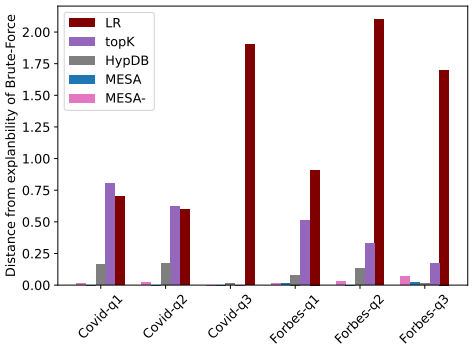

Figure 2: Distance from explainability scores of Brute-Force.

ignores redundancy among selected attributes. For example, in Flights Q_1, the explanation consists of 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 in which their p-value was lower than 0.05. In cases where it succeeded in generating explanations, the subjects found them to be less convincing than MESA. The reason is that LR focuses only on finding linear correlations among attributes.

Explainability scores. Let $\boldsymbol{E}$ denote the explanation found by an algorithm. We call $I(O;T|\boldsymbol{E})$ the explainability score. An Explainability score equal to 0 means that $\boldsymbol{E}$ perfectly explains away the correlation between O and T. We study the explainability scores of each algorithm. The explainability scores of Brute-Force serve as ground truth. In some cases, the explanations generated by all algorithms cannot fully explain away the correlations. For example, in Flights Q_2, the explainability score of MESA is 0.25. This means that other factors that affect flight delays may not exist in the KG (e.g., airline delays may be due to labor problems). Note that most of the explanations of all methods consist of attributes extracted from the KG (and not by attributes from the datasets). This validates our intuition that *explanations cannot be found in the data in many real-life scenarios but could be mined from KGs*. 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.

Recall that the responsibility scores can be used to rank the individual explanation power of selected attributes. This may help users to understand the explanations better. For example, in Forbes Q_3, the responsibility scores of Cups and Draft Pick are 0.57 and 0.42, resp. These high scores are because most of the athletes in the Forbes dataset are tennis and basketball players.

Missing Data. 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 the entities used for data extraction were celebrities' names from different categories (actors, athletes, etc.), and DBpedia uses different attributes to describe a person from each category. For example, for athletes, facts regarding their height and weight are typically present. However, these properties can rarely be found on actors' DBpedia pages. In Covid-19, SO, Flights, and Forbes, the percentage of attributes with selection bias (tested according to the conditions defined in Section 3.2) 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.*

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).

The percentage of unsuccessfully entity matches in each dataset are 12.2, 13.4, 8.7, and 9.5 in Covid-19, SO, Flights, and Forbes, resp. We note that any off-the-shelf entity linker could be used.

Impact of pruning. We next examine how useful were the pruning techniques (mentioned in Section 4.2). **Offline Pruning**. At the offline phase, we filter the extracted attributes using the simple filtering and the high-entropy techniques. We found those two simple 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, 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.

5.3 Efficiency Evaluation (Q2)

To examine the contribution of our optimizations, we report the running times of the following methods: **No Pruning** —the MCIMR algorithm without pruning; **Offline Pruning** —MCIMR only with offline pruning optimizations; **MCIMR** —MCIMR with all pruning optimizations. 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 5.2. In all cases, the execution time of MCIMR was less than 10 seconds, a reasonable response time for an interactive system. In what follows, 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. We study the effect of the number of candidate attributes on performance. For this experiment, we omitted from consideration attributes from $\mathcal{D}$ uniformly at random. The results are depicted in Figure 3. In all dataset, we exhibit a (near) linear growth in running times as a function of the size of $\mathcal{D}$. *Observe that the execution times of No-Pruning are significantly higher than those of Offline Pruning and MCIMR, indicating the usefulness of the offline pruning techniques.* The difference in times across the datasets is due to the size of the dataset. Estimating mutual information on large datasets (e.g., Flights, SO) takes longer than on small datasets (e.g., Forbes). Also, note that in the small Forbes dataset, Offline Pruning is faster than MCIMR. This is because, in the small Forbes

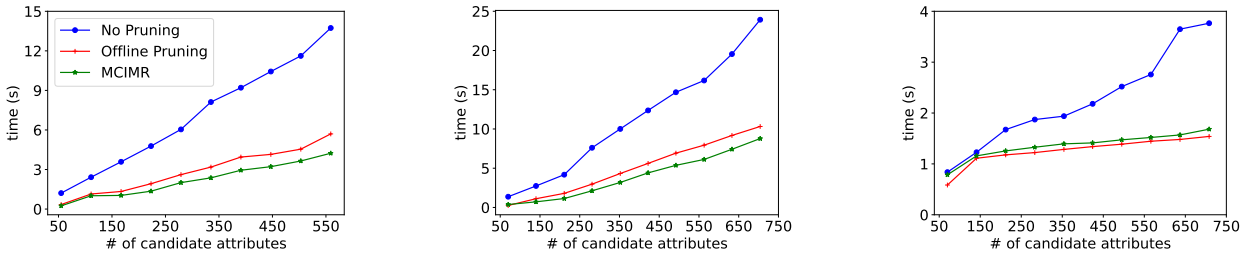

Figure 3: Running times as a function of the number of candidate attributes.

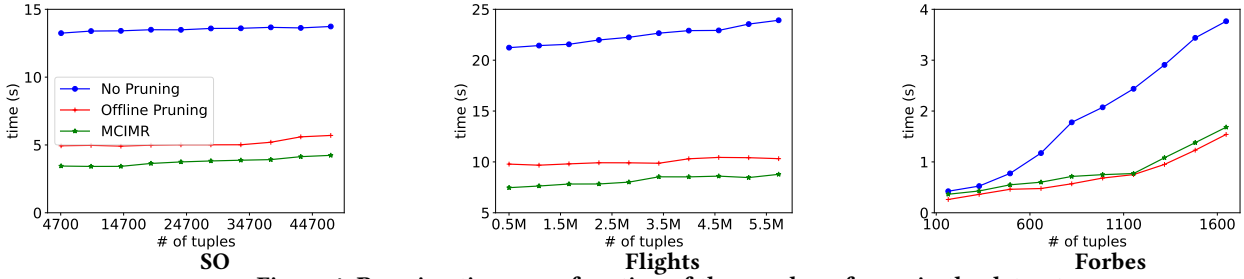

Figure 4: Running times as a function of the number of rows in the dataset.

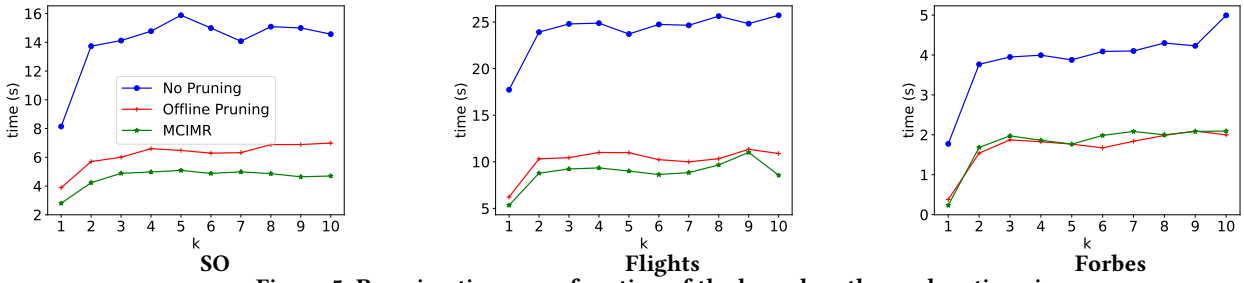

Figure 5: Running times as a function of the bound on the explanation size.

dataset, pruning took longer than running MCIMR, suggesting that in small datasets, online pruning is not necessary.

Dataset Size. We vary the number of tuples in the datasets, by removing tuples from the datasets uniformly at random. The results are depicted in Figure 4. Observe that in SO and Flights, we see that the dataset size has a little effect on running times. This is because the queries considered over those datasets are group-by queries. Thus, when randomly omitting tuples from the datasets, the number of considered groups is almost unchanged. On the other hand, since in Forbes the examined queries compared among tuples from the dataset (without a GROUP-BY statement), we exhibit a (near) linear growth in running times.

Explanation size. We vary the bound k on the explanation size. Recall that given a bound k, MCIMR returns an explanation of size up to k. It may return an explanation of size $l<k$ if the responsibility of the l+1 attribute is close to 0. The results are shown in Figure 5. In all cases, the size of the explanations was no bigger than 3. Thus, as can be seen, for all methods, k has almost no effect on running times, as the algorithms terminate after no more than 4 iterations.

5.4 Extensions (Q3)

In this part we examine the effect of extracting attributes from the KG using more than one hop, and demonstrate the effectiveness of our unexplained groups identification procedure.

Multi-Hops. We examine the effect of extracting attributes following more than one hop. We report that in the vast majority of cases, the explanations generated by MESA 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 (an aggregated 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 for the explanations.

Unexplained Subgroups. We demonstrate the effectiveness of the Top-K unexplained groups algorithm by focusing on SO Q_1, setting

Table 5: 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

$\tau > 0.2$. The top-5 largest unexplained groups for this query are given in Table 5. Observe that economy-related attributes (such as GDP, HDI, Gini Coefficient) of selected data groups are internally consistent (e.g., as mentioned, the HDI of countries in Europe is similar). Thus, it makes sense that the explanation generated for SO Q_1 ({HDI, Gini}) will not be a satisfactory explanation for these data groups. Indeed, as can be seen in Table 3, the explanation found by MESA for the top-1 unexplained data group (SO Q_3) is different from the explanation found for all countries.

We ran this algorithm over the 14 queries depicted in Table 5. The average execution time is 4.4*s*. This demonstrates the ability of this algorithm to efficiently identify subgroups (corresponding to query refinements) that are likely to be of interest to the user.

6 RELATED WORK

We discuss multiple lines of work that are relevant to ours.

Results Explanations. There is a wealth of work on query results explanation. Methods explaining why data is missing or mistakenly included in a query result have been studied in [13, 16, 31, 58]. Explanations for interesting or unexpected tuples in the results have been presented in [12, 42]. Those two lines of work are orthogonal to our work, as we aim to explain away unexpected correlations observed in queries. Another line of work provides explanations on how a query result was derived by analyzing the query provenance and pointing out database facts that significantly affect the results [39, 40, 43]. Those methods are designed to generate tuple-level explanations and not attribute-level explanations that are required for unearthing correlations between attributes. Another type of explanation for query results is a set of patterns that are shared by one (group of) tuple in the results but not by another (group of) tuple [22, 33, 51, 52, 62]. However, those works as well do not account for correlations among attributes. The authors of [53] presented HypDB, a system that provides explanations based on causal analysis, measured by the correlations among attributes. However, as mentioned in our experiments, HypDB only considers attributes from the input table, and it cannot efficiently handle a large amount of candidate confounding attributes.

We share with [33] the motivation for considering explanations that are not solely drawn from the input table. The authors of [33] presented CajaDE, a system that generates explanations of query results based on information mined from tables related to the table (accessed by the query). However, related tables often do not exist. Further, their explanations are a set of patterns that are unevenly distributed among groups in the query results. Such explanations are independent of the outcome. Thus, even if CajaDE is given the attributes mined from a KG, it cannot generate explanations that explain away the correlations between the exposure and outcome. Therefore, [33] is orthogonal to our work.

Feature Selection. The Correlation-Explanation problem is closely related to the well-studied Feature Selection (FS) problem [15, 26, 34], the problem of selecting a subset of the most relevant attributes for use in model construction. FS methods aim to select a concise and diverse set of attributes relevant to a target attribute [15]. The goal of FS is twofold: (1) *Simplification of models*: select as few features as possible. This makes models easier to interpret, reduces training times, and helps to avoid the curse of dimensionality [37]. (2) *Model accuracy*: minimize the classification error. If the underlying model is not given, The goal is to maximize the dependency between selected features and the target class [48]. We note the model simplification requirement corresponds to the conciseness and no-redundancy requirements of a "good" explanation, and model accuracy corresponds to explainability.

Closest to our project, there is a line of work using information-theoretical-based methods for FS [34]. Their main goal is to maximize the relevance of the selected features and minimize redundancy. Such methods are typically independent of the underlying learning algorithm. Algorithms in this family [21, 24, 32, 36, 41] define different criteria to measure the importance of features. Namely, to maximize feature relevance and minimize their redundancy. The relevance of a feature is typically measured by its correlation with the target attribute. Our proposed MCIMR algorithm is based on the first stage of a commonly used FS algorithm in this family, called MRMR [48]. The MRMR algorithm has been widely used in different domains, such as gene classification [19] and emotion recognition from electrodermal activity [56].

Confounding Bias. Confounding bias, referred to as a "mixing of effects", occurs when an analyst tries to determine the effect of an exposure on an outcome, but unintentionally measures the effect of another factor (referred to as the confounding variable) on the outcome. This results in a distortion of the actual association between the outcome and exposure [28, 57]. Overlooking demographic and clinical factors as potential confounding variables can bias clinical studies' results and lead to erroneous conclusions [25]. Identifying confounding bias is especially important in etiologic and medical studies, as the separation of the effects of confounding variables from the effect of the exposure is a key prerequisite for validly estimating the exposure effect. We share with [53] the motivation for identifying bias in SQL queries for improved decision making. However, we mine confounding attributes from KGs, and, as we demonstrated in our experiments, our approach is more scalable.

7 CONCLUSION AND FUTURE WORK

This paper presented the Correlation-Explanation problem, whose goal is to identify uncontrolled confounding attributes that explain away spurious correlations observed in SQL query results. We developed an efficient algorithm that finds the optimal subset of confounding attributes. This algorithm is embodied in a system called MESA, which adapts the IPW technique for handling missing data and enables users to identify unexplained data groups.

While we discussed simple SQL aggregate queries, an extension of our work can be made for more general queries (e.g., nested sub-queries). In the future, we plan to extend our work by mining explanations from text documents. Another interesting direction for future work is to automatically identify which links in the KG are relevant to the explanation and worthy to follow.

REFERENCES

[1] 2020. COVID-19 Dataset. https://www.kaggle.com/imdevskp/corona-virus-report?select=usa_county_wise.csv.

[2] 2020. Flights Delay Dataset. https://www.kaggle.com/usdot/flight-delays?select=flights.csv.

[3] 2021. Forbes Dataset. https://www.kaggle.com/datasets/slayomer/forbes-celebrity-100-since-2005.

[4] 2021. Stack Overfow developer survey. https://insights.stackoverflow.com/survey.

[5] 2022. DBPedia. https://www.dbpedia.org/.

[6] 2022. Pyitlib library. https://pypi.org/project/pyitlib/.

[7] 2022. spaCy Entity Linker. https://spacy.io/api/entitylinker.

[8] 2022. Technical Report. https://github.com/ResultsExplanations/ExplanationsFromKG.

[9] 2022. Wikidata. https://www.wikidata.org/wiki/Wikidata:Main_Page.

[10] 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.

[11] 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

[12] 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.

[13] Nicole Bidoit, Melanie Herschel, and Katerina Tzompanaki. 2014. Query-based why-not provenance with nedexplain. In *Extending database technology (EDBT)*.

[14] Laura E Brown and Ioannis Tsamardinos. 2008. Markov blanket-based variable selection in feature space.

[15] Girish Chandrashekar and Ferat Sahin. 2014. A survey on feature selection methods. *Computers & Electrical Engineering* 40, 1 (2014), 16–28.

[16] Adriane Chapman and HV Jagadish. 2009. Why not?. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*. 523–534.

[17] Xiaojun Chen, Shengbin Jia, and Yang Xiang. 2020. A review: Knowledge reasoning over knowledge graph. *Expert Systems with Applications* 141 (2020), 112948.

[18] 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.

[19] Chris Ding and Hanchuan Peng. 2005. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology* 3, 02 (2005), 185–205.

[20] Bradley Efron. 1994. Missing data, imputation, and the bootstrap. *J. Amer. Statist. Assoc.* 89, 426 (1994), 463–475.

[21] 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.

[22] 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.

[23] 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.

[24] François Fleuret. 2004. Fast binary feature selection with conditional mutual information. *Journal of Machine learning research* 5, 9 (2004).

[25] Sander Greenland and Raymond Neutra. 1980. Control of confounding in the assessment of medical technology. *International journal of epidemiology* 9, 4 (1980), 361–367.

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

[27] David Hinkley. 1985. Transformation diagnostics for linear models. *Biometrika* 72, 3 (1985), 487–496.

[28] KJ Jager, C Zoccali, A Macleod, and FW Dekker. 2008. Confounding: what it is and how to deal with it. *Kidney international* 73, 3 (2008), 256–260.

[29] 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.

[30] 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.

[31] Seokki Lee, Bertram Ludäscher, and Boris Glavic. 2020. Approximate summaries for why and why-not provenance (extended version). *arXiv preprint arXiv:2002.00084* (2020).

[32] David D Lewis. 1992. Feature selection and feature extraction for text categorization. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

[33] 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.

[34] 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.

[35] Jonathan B Lim and Daniel M Oppenheimer. 2020. Explanatory preferences for complexity matching. *PloS one* 15, 4 (2020), e0230929.

[36] 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.

[37] Huan Liu. 2010. *Feature Selection*. Springer US, 402–406.

[38] David Maier, Jeffrey D Ullman, and Moshe Y Vardi. 1984. On the foundations of the universal relation model. *ACM Transactions on Database Systems (TODS)* 9, 2 (1984), 283–308.

[39] 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).

[40] 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).

[41] 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.

[42] 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.

[43] 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.

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

[45] Michael Pacer and Tania Lombrozo. 2017. Ockham's Razor Cuts to the Root: Simplicity in Causal Explanation. *Journal of Experimental Psychology* 146, 12 (2017), 1761.

[46] 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.

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

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54] 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.

[55] 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.

[56] Jainendra Shukla, Miguel Barreda-Angeles, Joan Oliver, GC Nandi, and Domenec Puig. 2019. Feature extraction and selection for emotion recognition from electrodermal activity. *IEEE Transactions on Affective Computing* 12, 4 (2019), 857–869.

[57] Andrea C Skelly, Joseph R Dettori, and Erika D Brodt. 2012. Assessing bias: the importance of considering confounding. *Evidence-based spine-care journal* 3, 01 (2012), 9–12.

[58] 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.
[59] Ioannis Tsamardinos and Constantin F Aliferis. 2003. Towards principled feature selection: Relevancy, filters and wrappers. In *International Workshop on Artificial Intelligence and Statistics*. PMLR, 300–307.
[60] 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.
[61] 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.
[62] Eugene Wu and Samuel Madden. 2013. Scorpion: Explaining away outliers in aggregate queries. (2013).
[63] 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.
[64] 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 3.1. 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 3.2. 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 [48] 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 [48], 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 [48], 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 4.2. 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:

$$I(O;T|\boldsymbol{E_k}) - I(O;T|\boldsymbol{E_k},E_{k+1}) = H(O|T,\boldsymbol{E_k},E_{k+1}) - H(O|T,\boldsymbol{E_k}) \le 0$$

For the last inequality we used the fact that for every three ransom variables X,Y,X: $H(X|Y) \le H(X|Y,Z)$, 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 $Resp(E_{k+1}) \le 0$ □