ZaliQL: Causal Inference from Observational Data at Scale

Babak Salimi Corey Cole Dan R. K. Ports Dan Suciu
Department of Computer Science & Engineering
University of Washington
{bsalimi, drkp, suciu}@cs.washington.edu, coreylc@uw.edu

ABSTRACT

Causal inference from observational data is a subject of active research and development in statistics and computer science. Many statistical software packages have been developed for this purpose. However, these toolkits do not scale to large datasets. We propose and demonstrate ZaliQL: a SQL-based framework for drawing causal inference from observational data. ZaliQL supports the state-of-the-art methods for causal inference and runs at scale within PostgreSQL database system. In addition, we built a visual interface to wrap around ZaliQL. In our demonstration, we will use this GUI to show a live investigation of the causal effect of different weather conditions on flight delays.

1. INTRODUCTION

Randomized experiments (A/B testing) remain the gold standard for causal inference; however, they do pose a number of problems. Namely, controlled experiments are not feasible for ethical, economical, or practical reasons in a number of disciplines [9]. Observational studies can be used to draw causal inference without controlled experiments [9, 11, 7].

Computational, physical, and social scientists all increasingly want to perform causal inference on big observational data, e.g., data from social networks and biological networks. Unfortunately, the current software for processing observational data in terms of causal inference cannot scale. R, Stata, SAS, and SPSS all have packages [3, 5]; however, they are designed to be used only with single table data, making them cumbersome and often ineffective with large datasets. For example, we found performing CEM on a dataset with 5M entries takes up to an hour using Stata, R, or SAS. This is obviously not an effective practice for researchers.

Additionally, causal analysis is part of a larger pipeline that includes data acquisition, cleaning, and integration. For large datasets, these tasks are better handled by a relational database engine which provides most of the functionality needed for these tasks while also scaling up to large datasets.

Proceedings of the VLDB Endowment, Vol. 10, No. 12

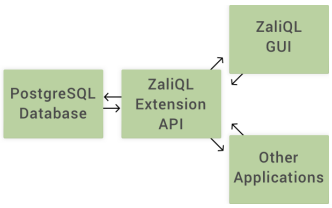

Figure 1: ZaliQL architecture

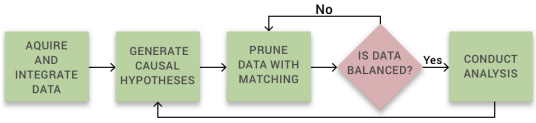

Figure 2: Causal analysis workflow

In this demonstration, we propose ZaliQL,[1] a SQL-based framework for drawing causal inference. ZaliQL takes the initial step towards scalable causal inference by modeling it as a data management problem. We show that causal inference can be approached from this perspective and that doing so is key for scalable and robust causal analysis. ZaliQL supports state-of-the-art methods for causal inference and runs at scale within a database engine.

2. SYSTEM ARCHITECTURE

The overall architecture of ZaliQL can be seen in Fig. 1. The API is a set of functions that support causal inference on data stored in a PostgreSQL DBMS. The API will be packaged as a PostgreSQL extension. The ZaliQL API is modeled after the MatchIt and CEM toolkits [3, 5] and includes methods for drawing causal inference from relational data. A web GUI is also included for demonstration and exploration purposes and is illustrated in Fig. 3.

3. DEMONSTRATION DETAILS

As the illustration in Fig. 2 shows, modern causal analysis is an iterative process. An analyst must acquire and integrate data from a myriad of sources, generate a hypothesis, pre-process the data through a matching method, and finally conduct the causal analysis. This linear process often must be repeated with new matching methods, new hypotheses, or even new datasets [4]. Note that our system does not address the statistical validity of performing multiple hypothesis testing. We will demonstrate ZaliQL by providing

[1] The prefix Zali refers to al-Ghzali (1058-1111), a medieval Persian philosopher. It is known that David Hume (1711-1776), a Scottish philosopher, who gave the first explicit definition of causation in terms of counterfactuals, was heavily influenced by al-Ghzali's conception of causality.

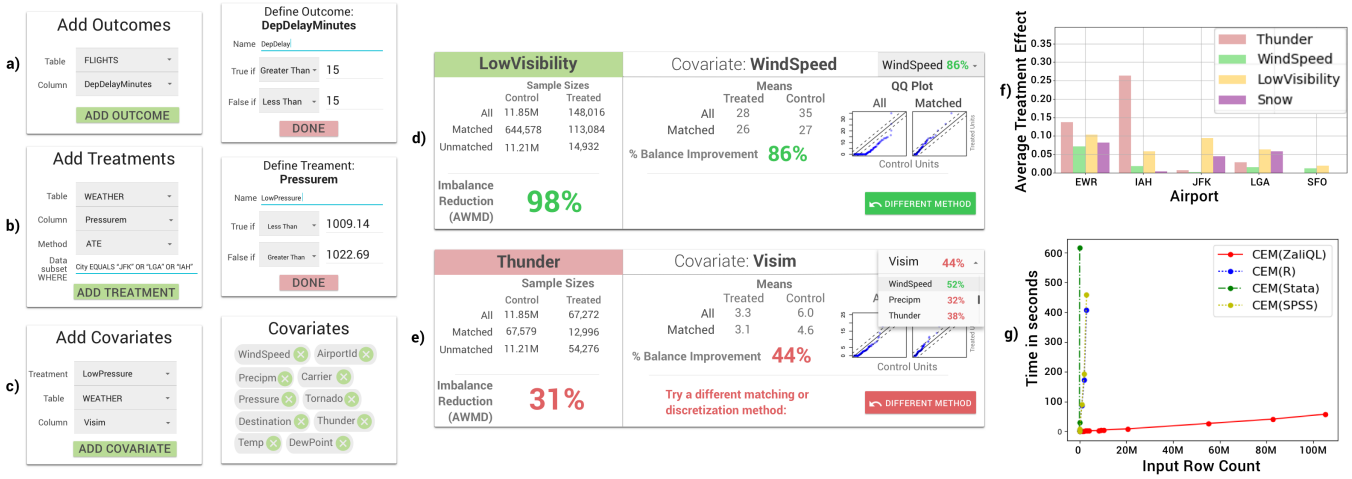

Figure 3: Demonstration screenshot described in Section 3

walkthroughs using causal investigations on integrated flight and weather datasets.

Data: The analysis will be conducted on a spatio-temporal join of the following datasets: (a) *Flight data (105M entries)* collected by the U.S. Department of Transportation. This dataset contains records of over 90% of U.S. domestic flights of major airlines between 1988 and the present. It includes the following variables: Date, AirportID, CarrierID, and DepDelay (departure delay); (b) *Weather data (40M entries):* collected using Weather Underground API.[2] It contains historical weather data of U.S. airports and includes the following variables: Code (airport ID), Date, Time, Visim (visibility in km), Tempm (temperature in Celsius), Wspdm (wind speed in kph), Pressurem (pressure in mBar), Precipm (precipitation in mm), Snow (binary), and Thunder (binary).

Data exploration: Our demonstration will start by exploring the effect of different weather features on flight departure delay. As an example, we will show that 11% of flights were delayed when pressure was low; however, only 0.4% of flights were delayed when pressure was high. This suggests that pressure is inversely correlated with flight delay. However, after grouping by different variables such as airport, airline, and other weather features, flight delay frequency declined from 11% to nearly 0%. [3] Thus, we are forced to reconsider if low pressure is really a causative factor for weather delays.

Causal questions: The following causal questions have been defined: Q1: Does low air pressure cause flight departure delays? Q2: Which weather features are major causes of departure delays? Q3: Do the findings to the previous questions differ between major airports? These questions will be answered by ZaliQL using the following steps:

- Specifying DepDelay as our outcome of interest (effect) as shown in Fig. 3a.
- Specifying a set of binary treatments (causes) that could affect DepDelay as shown in Fig. 3b. Particularly, the following binary treatments will be created: LowVisibility (1 if Visim < 1); HeavySnow (1 if Prcipm > 0.3 and Snow $= 1$); HighWindSpeed (1 if Wspdm > 40; 0 if Wspdm < 20); Thunder; Low-Pressure (1 if Pressurem < 1009.14; 0 if Pressurem > 1022.69).
- Specifying a relevant subset of data for analysis. We will select five U.S. airports with frequent weather-related delays. Specifically, San Francisco (SFO), John F. Kennedy (JFK), Newark Liberty (EWR), George Bush-Houston (IAH), and LaGuardia (LGA).

Computing ATE: The primary objective of causal inference is to quantify the causal effect of a binary treatment on an outcome. It is quite common to compare an outcome (DepDelay) between the *treated* and *control* groups. That is, to compare subjects (flights) that receive a treatment (LowPressure $= 1$) with subjects that do not (LowPressure $= 0$). A common measure to do this is that of average treatment effect (ATE) which is computed, for example, for Q1, as follows:

$$\mathbb{E}[\text{DepDelay}|\text{LowPressure} = 1] - \mathbb{E}[\text{DepDelay}|\text{LowPressure} = 0]$$

In this example, $\mathbb{E}[\text{DepDelay}|\text{LowPressure} = x]$, $x = (0, 1)$ is computed by taking the emperical average of DepDelay where LowPressure is x. We show that for LowPressure ATE is significantly large, which suggests pressure affects DepDelay. However, it is known from prior analysis that LowPressure alone does not serve as the sole causative factor for departure delay. This observation thus raises the question: Where is this difference coming from?

Confounding variables: The difference is a product of confounding variables, which make it difficult to establish a causal link between a treatment and outcome. This is often the case in real world analysis as the myriad of variables involved greatly complicate understanding a phenomenon. In this example, we will show that the observed positive effect of LowPressure was actually the result of the confounding influence of other factors including low visibility, snow, and thunder. Specifically, we will show that Low Pressure is associated with unsettled weather conditions such as LowVisibility (see Fig. 4). The average of LowVisibility in the treated group is much less than that of the opposite group. Thus, it is unclear that the observed DepDelay difference between the groups is caused by LowPressure or LowVisibility.

[2] `https://www.wunderground.com`

[3] This phenomenon is known as Simpson's paradox and arises frequently in observational studies [8].

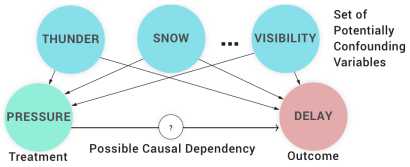

Figure 4: Confounding influence

Adjusting for confounding variables: For the sake of drawing valid conclusions to the causal questions, we must adjust for these confounding variables. Conceptually, adjusting for one confounding variable is easy. It involves partitioning the data into similar groups with similar confounding influence measures. Next, the ATE is computed using the weighted average of the effects *in each group*. However, real world causal inference is much more complicated as it involves many confounding variables. Fig. 4 shows some of the many confounding variables in this example. In this case, the groups become overly specific and begin to lack enough units to create a meaningful calculation of the ATE.

Sophisticated techniques are required to adjust for a large set of confounding variables. ZaliQL implements two primary approaches that are commonly found in statistics: *coarsened exact matching (CEM)* and *propensity score matching (PSM)* [4, 10]. The goal of matching is to prune data so that the remaining *matched* data have greater *balance* between the treated and control groups. In other words, the empirical distributions of the confounding variables in the treated and control groups are similar after matching. Once the groups have more or less achieved a sense of balance, the observed difference of the outcome between the two groups can be attributed to the treatment. In our demonstration, we will select a set of covariates deemed to confound the treatments and outcome (Fig. 3c) and we will select a matching method and adjust its tuning parameters.

Checking balance: This next step requires verifying that the matching process improved the covariates balance. Specifically, we will compare the distribution for each covariate between the treated and control group on the matched data for all treatments. ZaliQL provides both numerical and visual summaries for this step of the process including quartile-quartile and mean difference plots. This can be seen in Fig. 3d and Fig. 3e.

Answers to our causal questions: Finally, we are able to answer the causal questions created in the first step of our process. For Q1, we will show that LowPressure has no significant causal effect on DepDelay. For Q2, we will identify that other treatments have significant causal effects on DepDelay (Fig. 3f). Regarding Q3, we will report that the major causes of flight delay at the airports included in the study are actually different. Two sample z-tests will be used to validate the statistical significance of the results. We will show that the obtained results are validated by FAA reports.

Scalability: The last part of this demonstration will allow us to show the scalability by letting users interactively run queries of this large dataset. This process would take existing toolkits hours to complete; however, as seen in Fig. 3g, ZaliQL can process 100M entries in less than a few minutes, whereas other systems such as R, SPSS, or SAS, take up to an hour just to process 5M entries.[4]

[4]We note that the developers of statistical software packages for CEM have identified ZaliQL as a more scalable approach: see `http://gking.harvard.edu/cem`.

Unit	Covariates X	Treatment assignment T	Treatment outcome $Y(1)$	Control outcome $Y(0)$	Causal Effect $Y(1)-Y(0)$
1	X_1	T_1	$Y_1(1)$	$Y_1(0)$	$Y_1(1)-Y_1(0)$
2	X_2	T_2	$Y_2(1)$	$Y_2(0)$	$Y_2(1)-Y_2(0)$
...	...	...	...	...	...
N	X_N	T_N	$Y_N(1)$	$Y_N(0)$	$Y_N(1)-Y_N(0)$

Figure 5: The Potential Outcome Framework

4. INTERNALS

4.1 Basic Formalism

The basic causal model in statistics is called the *potential outcome framework* (POF) [11]. In this model, we are given a table with N rows called *units* indexed by $i = 1 \ldots N$ (see Table 5). The binary attribute T denotes *treatment assignment*. X is a vector of background characteristics, (e.g., airport, airline, weather ...) of each unit, called *covariates*, unaffected by treatment; and the two attributes $Y(0), Y(1)$ represent *potential outcomes*: $Y(1)$ is the outcome of the unit if it is exposed to the treatment and $Y(0)$ is the outcome when it is exposed to the control.

The *treatment effect* caused by the treatment T_i for the ith unit is defined as $Y_i(1) - Y_i(0)$. The goal of causal analysis is to compute the *average treatment effect (ATE)*:

$$ATE = \mathbb{E}[Y(1) - Y(0)] = \mathbb{E}[Y(1)] - \mathbb{E}[Y(0)].$$

The so-called *fundamental problem of causal inference* is that for each unit we only know either $Y(1)$ or $Y(0)$ but not both. For example, each individual flight has either LowPressure=1 or LowPressure=0, so only one of DepDelay(1) or DepDelay(0) is available for each row. Thus, further assumptions are needed for estimating ATE.

The strongest is the *independence assumption*: the treatment mechanism is independent of the potential outcomes, i.e., $(Y(1), Y(0)) \perp\!\!\!\perp T$. This might hold in a properly constructed randomized trial. Then, $\mathbb{E}[Y(1)] = \mathbb{E}[Y(1)|T = 1]$ and similarly $\mathbb{E}[Y(0)] = \mathbb{E}[Y(0)|T = 0]$ and so:

$$ATE = \mathbb{E}[Y(1)|T = 1] - \mathbb{E}[Y(0)|T = 0].$$

However, ZaliQL is designed for drawing causal inference from *observational data*, where independence fails in general. Here, statistical literature makes two standard, weaker assumptions called *Strong Ignorability*: for all x:
(1) $Y(0), Y(1) \perp\!\!\!\perp T|X = x$ (unconfoundedness) and
(2) $0 < \Pr(T = 1|X = x) < 1$ (overlap) [10].

If strong ignorability holds, one can estimate ATE by taking the average difference for each group with a particular value of X i.e., $\mathbb{E}_x[\mathbb{E}[Y(1) - Y(0)|X = x]]$. In practice, however, a direct application of this method is impossible, because the data is typically very sparse: for any value $X = x$ we either have no data values at all, or very few such values, which means that estimating $\mathbb{E}[Y(T)|T = 1 \text{ or } 0, X = x]$ as an average from the database leads to large sampling error. A solution adopted in statistics is *matching* [10].

4.2 Matching Methods

This process involves theoretically matching each treated unit to one of multiple control units with similar values of the covariate attributes [4]. Closeness is defined using some distance function between the covariate values of two units. Given a table, $R(T, X_1 \ldots, X_n, Y)$, ZaliQL offers the following two matching methods.

```
CREATE VIEW PSM_Matched
AS WITH potential_matches AS
  (SELECT treated.ID AS tID, control.ID AS cID,
          abs(Treated.PS-Control.PS)  AS distance
   FROM R AS control, R AS treated
   WHERE control.T=0 AND treated.T=1
     AND abs(Treated.PS-Control.PS) < caliper)),
           ordered_potential_matches AS
  (SELECT *, ROW_NUMBER() over (ORDER BY distance) AS order
   FROM potential_matches)
SELECT *
FROM ordered_potential_matches AS rp
WHERE NOT EXISTS
    (SELECT *
     FROM ordered_potential_matches AS z
     WHERE z.order < opm.order AND z.cID=opm.cID)
  AND (SELECT count(*)
     FROM ordered_potential_matches AS opm
     WHERE z.order < opm.order AND z.tID=opm.tID) ≤ k;
```

Figure 6: Propensity score matching

```
CREATE VIEW CEM_Matched AS
WITH subclasses AS
  (SELECT *,
          max(ID) OVER w subclass, max(T) AS min_treatment,
          min(T)AS max_treatment
   FROM R^c
   Group by 𝒳
   Having min_treatment!=max_treatment)
SELECT *
FROM subclasses,R^c
WHERE subclasses.𝒳=R^c.𝒳
```

Figure 7: Coarsened Exact Matching

Propensity score matching: The most common method is $k:1$ nearest neighbor matching based on *propensity score matching (PSM)* [10]. Propensity score is the probability of a unit being assigned to a treatment given a set of covariates (i.e. $P(x) = P(T = 1|X = x)$). This method selects the k best control matches for each individual in the treatment group which are closer than a pre-specified *caliper*. ZaliQL estimates propensity score using logistic regression.

The basic SQL statement to perform PSM is depicted in Fig. 6. In this solution, nearest control units are identified by means of an anti-join. In other words, all potential matches, and their distances, are identified by joining the treated with the control units that are closer on propensity score than the caliper. Then, this set is sorted in ascending order of distances. Additionally, the order of each row in the sorted set is identified using the window function `ROW_NUMBER`. Finally, k closest controls are selected as the matched units. Note that ZaliQL generates more efficient SQL statements using recent developments in spatial-databases (see e.g., [6]).

Coarsened exact matching (CEM): In this method, the vector of covariates X is coarsened according to a set of user-defined cut points or with a discretization algorithm. All units with similar coarsened covariate values are placed in unique groups. Then, groups with at least one treated and one control unit are retained while all others are discarded from data. We let $\mathcal{X}$ be the coarsened version of X and R^c be a coarsened version of R. As depicted in Fig. 7, CEM is essentially a GROUP-BY-HAVING query.

We observed that CEM is an "Iceberg query" in the sense that it is a GROUP-BY-HAVING query where the output size is typically much smaller than the input (i.e. the tip of an iceberg). Iceberg queries have been studied extensively in databases and data mining [1, 2]. ZaliQL leverages these techniques and the prior research on them in order to more efficiently compute CEM for several treatments simultaneously, giving it a strong advantage over other softwares.

Analysis after matching: After a balanced and matched subset of data is extracted, ATE can be computed. ZaliQL supports a wide range of statistical tests such as z-test, t-test, egression-based tests , and Chi-square test to compute the statistical significant of the treatment group.

5. CONCLUSIONS

In this demonstration, we will introduce ZaliQL, a tool for performing causal inference on large relational data within a DBMS. ZaliQL makes the first step towards truly scalable causal inference by modeling it as a data management problem. ZaliQL implements a wide range of methods for causal inference developed in statistics with existing techniques in data management. This provides scalable evaluation of several causal hypothesis on relational data. Overall, this demonstration will serve as both an introduction to causal inference and an illustration of the vast benefits ZaliQL possesses over traditional statistical analysis packages.

6. ACKNOWLEDGEMENTS

This work is supported in part by the National Science Foundation through NSF grants IIS-1614738 and University of Washington's CSE Postdoc Research Award.

7. REFERENCES

[1] M. Fang, N. Shivakumar, H. Garcia-Molina, R. Motwani, and J. D. Ullman. Computing iceberg queries efficiently. In *VLDB*, 1998.

[2] L. Findlater and H. J. Hamilton. Iceberg-cube algorithms: An empirical evaluation on synthetic and real data. *Intelligent Data Analysis*, 7(2):77–97, 2003.

[3] D. E. Ho, K. Imai, G. King, and E. A. Stuart. Matching as nonparametric preprocessing for reducing model dependence in parametric causal inference. *Political analysis*, 15(3):199–236, 2007.

[4] S. M. Iacus, G. King, and G. Porro. Causal inference without balance checking: Coarsened exact matching. *Political analysis*, page mpr013, 2011.

[5] S. M. Iacus, G. King, G. Porro, et al. Cem: software for coarsened exact matching. *Journal of Statistical Software*, 30(9):1–27, 2009.

[6] R. O. Obe and L. S. Hsu. *PostGIS in action*. Manning Publications Co., 2015.

[7] J. Pearl. *Causality: models, reasoning, and inference*. Cambridge University Press, 2000.

[8] J. Pearl. Comment: understanding simpsons paradox. *The American Statistician*, 68(1):8–13, 2014.

[9] P. R. Rosenbaum. *Observational studies*. Springer, 2002.

[10] P. R. Rosenbaum and D. B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):pp. 41–55, 1983.

[11] D. B. Rubin. Causal inference using potential outcomes. *Journal of the American Statistical Association*, 100(469):322–331, 2005.