ZaliQL: Causal Inference from Observational Data at Scale

Babak Salimi Corey Cole Dan Port 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

To this day, *randomized experiments* (A/B testing) remain the gold standard for causal inference. However, in many domains, controlled experiments are not feasible for ethical, economical or practical reasons [9]. *Observational studies* can be used to draw causal inference without controlled experiments [9, 11]. Computational, physical, and social scientists all increasingly want to perform causal inference on big data from observational studies: social networks, biological networks, sensor networks and more. Unfortunately, current software for processing observational data for causal inference does not scale. R, Stata, SAS, and SPSS have packages such as *MatchIt* and *CEM*[4, 6], but they are designed for single-table data and are cumbersome 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.

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, and also scales up to large datasets.

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

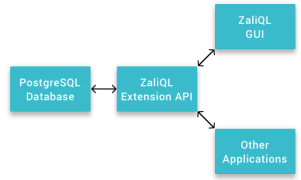

Figure 1: ZaliQL architecture

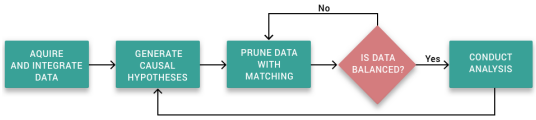

Figure 2: Causal analysis workflow

In this demonstration, we propose ZaliQL,[1] a SQL-based framework for drawing causal inference within the DBMS. ZaliQL takes the first step towards truly scalable causal inference by modeling it as a data management problem. We demonstrate that causal inference can be approached from this perspective, and that doing so is key to scalability and robustness. 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 is shown 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 [4, 6] and includes methods for drawing causal inference from relational data. For demonstration and exploration purposes, ZaliQL also includes a web GUI (see Fig. 3).

3. DEMONSTRATION DETAILS

Modern causal analysis is an iterative process, as illustrated in Fig 2. An analyst acquires and integrates data from multiple sources, generates a hypothesis, pre-processes the data with a matching method (explained below), and then finally conducts the causal analysis. Most often this process needs to be repeated with a new matching method, a new hypothesis or new datasets [5]. We demonstrate ZaliQL by showing several causal investigations on integrated flight and weather datasets.

[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.

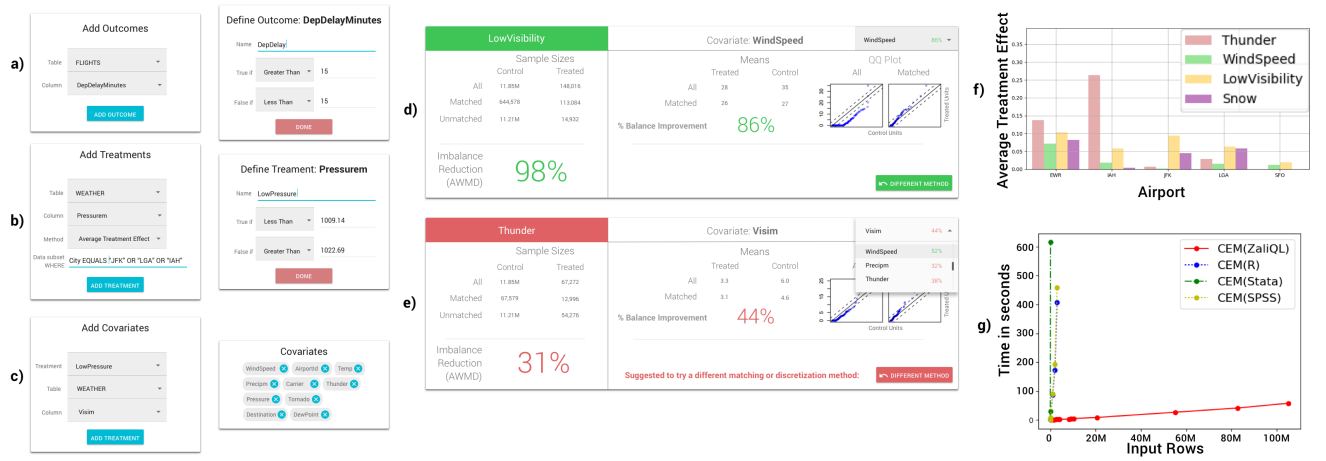

Figure 3: Demonstration screenshot described in Section 3

Data: The analysis will be conducted on a spatio-temporal join of the following two datasets: (a) *Flight dataset (105M entries)*: collected by the US Department of Transportation.[2] It contains records of more than 90% of US domestic flights of major airlines from 1988 to the present. It includes attributes such as FlightDate, OriginAirportID, CarrierID, CRSDepTime (scheduled departure time), and DepDelay (departure delay). (b) *Weather dataset (40M entries):* collected using the Weather Underground API.[3] It contains historical weather data for related to flights. It includes attributes such as Code (Airport ID), Date, Time, Visim (visibility in km), Tempm (Temperature in C°) Wspdm (wind speed in kph), Pressurem (Pressure in mBar), Precipm (Precipitation in mm), Snow (binary), Thunder (binary).

Data Exploration: Our demonstration starts by exploring the effect of different weather features on flight departure delay. As an example, we show that 11% of flights were delayed when pressure was low, but only 0.4% of flights delayed when pressure was high. This suggests that pressure is *inversely correlated* with flight delay (as pressure goes down, flights tend to be more delayed). However, after grouping by variables like airport, airline and other weather features, flight delay frequency declined from 11% to almost 0%.[4] In light of this, does low pressure really affect flight delays?

Causal questions: In this step, we define the following *causal questions*: Q1: Does low air pressure causes flight departure delays? Q2: Which weather features are major causes of departure delays? Q3: Do the findings to the previous question differ between major airports in the states? We answer these questions using ZaliQL by

- specifying DepDelay as our outcome of interest (effect) as shown in Fig. 3(a).
- specifying a set of binary treatments (causes) that might affect DepDelay, as shown in Fig. 3(b). In particular, the following binary treatments will be created: LowVisibility (1 if Visim< 1); HeavySnow (1 if Precipm> 0.3 and Snow= 1); HighWindSpeed (1 if Wspdm> 40; 0 if Wspdm< 20); Thunder; LowPressure (1 if Pressurem< 1009.14; 0 if Pressurem> 1022.69).
- specifying a subset of data that is relevant to the analysis. We select five US airports with high rate of weather-related delay, namely, San Francisco (SFO), John F. Kennedy (JFK), Newark Liberty (EWR), George Bush (IAH), and LaGuardia Airport (LGA).

Computing ATE: In causal inference, the objective is usually to quantify the causal effect of a binary treatment on an outcome. A common measure of effect is *average treatment effect* (ATE). ATE is defined as the difference in expected values of the treated and untreated units.[5] For example for Q1, ZaliQL computes ATE as

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

where, $\mathbb{E}[\text{DepDelay}|\text{LowPressure} = x]$, $x = (0, 1)$ is computed by taking the emetical average of DepDelay where LowPressure is x. It is a common measure to compare an outcome (DepDelay) in the *treated group*, those subjects (flights) that receive a treatment (LowPressure=1), with the *control group* (LowPressure=0). We show that for LowPressure, ATE is 10 minutes. This is relatively large and suggests pressure affects DepDelay. However, it is known that LowPressure alone does not cause departure delay. This observation raises the question: where is this difference coming from?

Confounding variables: The difference is a product of *confounding variables* that make it difficult to establish a causal link between a treatment and outcome. We will show that the observed positive effect of LowPressure was actually the result of the confounding influence of other factors such as low visibility, snow and thunder. Specifically, we show that, LowPressure is highly associated with unsettled weather conditions such as LowVisibility (Fig. 4). For instance the average of LowVisibility in the treated group is much less than the opposite group. Thus, it is unclear that the observed DepDelay difference between the groups is caused by LowPressure or LowVisibility.

[2] `http://www.transtats.bts.gov/`

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

[4] This phenomenon is known as Simpson's paradox and arises frequently in observational studies [8]. We demonstrate that the issue could be avoided by conducting a careful causal analysis.

[5] In causality literature, subjects of a study – e.g., patients in a clinical trial – are called units. In our case, each flight is considered a unit.

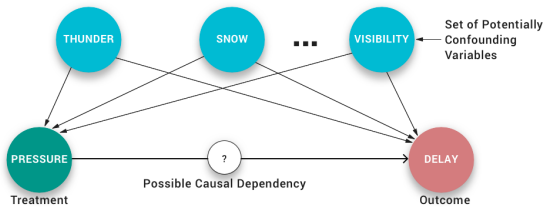

Figure 4: Confounding influence

Adjusting for confounding variables: In order to draw valid causal conclusions, we need to adjust for confounding influences. Conceptually, adjusting for one confounding variable is easy. First, we partition the data into groups with similar confounding influence measures. Then, we compute the ATE as the weighted average of the effects *in each group*. However, real-world causal inference requires many confounding variables. For example, LowPressure has more than ten confounding variables (some of them are shown in Fig. 4). In such a case, the groups often become too specific, lacking enough units in each group to enable a meaningful estimation of the ATE.

Sophisticated techniques are required to adjust for a large set of confounding variables. ZaliQL implements two dominant approaches used in social science and statistics namely, *coarsened exact matching (CEM)* and *propensity score matching(PSM)* [5, 10]. The goal of matching is to prune data so that the remaining *matched* data have better *balance* between the treated and control groups. That is, the empirical distributions of the confounding variables in the treated and control groups are similar after matching. Once the two groups are balanced, any observed difference of the outcome between the two groups can be attributed to the treatment. In our demo, (1) for each treatment, we select a set of covariates deemed to confound the treatment and DepDelay (Figure 3(c)) and (2) we select a matching method and adjust its tuning parameters.

Checking Balance: In this step, we check whether the matching process successfully improved the covariates' balance. In particular, (1) we compare the distribution for each covariate between the treated and control group on the matched data (this will be done for all treatments). For this task, ZaliQL provides numerical and visual summaries such as mean difference and quantile-quantile plots: Fig. 3(d,e).

Answers to our causal questions: In this step, we answer the causal questions created at the very first step of the demonstration using ZaliQL. For Q1, we show that LowPressure has no significant causal effect on DepDelay. For Q2, we identify that other treatments have significant causal effect on DepDelay (Figure 3(f)). For Q3, we report the major causes of flight delay at the airports under the study are actually different. We validate the statistical significance of our findings with t-tests. We show that the obtained results are in accordance with FAA reports.

Scalability: Finally, we demonstrate scalability by letting users interactively run queries on this large dataset– which would take the existing toolkits hours. As seen in Figure 3(g), ZaliQL can process 100M entries in less than a few minutes, whereas existing systems like R, SPSS or SAS, take up to an hour to process just 5M entries. [6]

[6]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* (FPCI) 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*, which states that the treatment mechanism is independent of the potential outcomes, i.e., $(Y(1), Y(0)) \perp\!\!\!\perp T$. Then, it holds that $\mathbb{E}[Y(1)] = \mathbb{E}[Y(1)|T = 1]$ and similarly $\mathbb{E}[Y(0)] = \mathbb{E}[Y(0)|T = 0]$ and we have:

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

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

If strong ignorability holds, one can estimate ATE by taking the average difference for each group with $X = x$ 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

The idea is to match each treated unit to one or multiple control units with "close" values of the covariate attributes X (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:

Propensity score matching: The most common method is $k:1$ nearest neighbor matching based on *propensity score matching* (PSM) [10]. A propensity score is the probability

```
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

of a unit being assigned to a particular 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, that are closer than a pre-specified *caliper*, in a greedy manner. 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. Specifically, 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 into ascending order of distances. In addition, 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: ZaliQL generates more efficient SQL statements by leveraging recent developments in the context of spatial-databases (see e.g., [7]).

Coarsened exact matching (CEM): In this method the vector of covariates X is coarsened according to a set of user-defined cutpoints or any automatic discretization algorithm. All units with similar coarsened covariate values are placed in unique groups. All groups with at least one treated and one control unit are retained and the rest of units are discarded. 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. ZaliQL also supports several supervised and unsupervised methods for coarsening proposed in [1].

We observed that CEM is an *Iceberg query*. These are GROUP-BY-HAVING queries in which the output size is typically much smaller than the input (the tip of an iceberg). Iceberg queries have been studied extensively in databases and data mining (see e.g., [2, 3]). ZaliQL leverages these techniques to efficiently compute CEM for several treatments simultaneously.

Analysis after matching: After a balanced matched subset of data is extracted, ATE can be computed. ZaliQL supports a wide range of statistical tests such as z-test, t-test, linear regression t-test, and Chi-Square test to compute the statistical significance of the treatment effect.

5. CONCLUSIONS

In this demonstration, we 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 hypotheses on relational data.

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] J. Dougherty, R. Kohavi, M. Sahami, et al. Supervised and unsupervised discretization of continuous features. In *Machine learning: proceedings of the twelfth international conference*, volume 12, pages 194–202, 1995.

[2] M. Fang, N. Shivakumar, H. Garcia-Molina, R. Motwani, and J. D. Ullman. Computing iceberg queries efficiently. In *Internaational Conference on Very Large Databases (VLDB'98), New York, August 1998*. Stanford InfoLab, 1999.

[3] 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.

[4] 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.

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

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

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

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

[9] P. R. Rosenbaum. Observational studies. In *Observational Studies*, pages 1–17. 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.