Cohort Analysis with Ease

Zhongle Xie
zhongle@comp.nus.edu.sg
National University of Singapore

Qingchao Cai
caiqc@comp.nus.edu.sg
National University of Singapore

Fei He
f_ho@outlook.com
National University of Singapore

Gene Yan Ooi
gene@shentilium.com
Shentilium Private Limited

Weilong Huang
weilong@shentilium.com
Shentilium Private Limited

Beng Chin Ooi
ooibc@comp.nus.edu.sg
National University of Singapore

ABSTRACT

The tremendous volume of user behavior records generated in various domains provides data analysts new opportunities to mine valuable insights into user behavior. Cohort analysis, which aims to find user behavioral trends hidden in time series, is one of the most commonly used techniques. Since traditional database systems suffer from both operability and efficiency when processing cohort analysis queries, we proposed COHANA[4], a query processing system specialized for cohort analysis. In order to make COHANA easy-to-use, we present a comprehensive and powerful tool in this demo, covering the major use cases in cohort analysis with intuitive and accessible operations. Analysts can easily adapt COHANA to their own use with provided visualizations which can help verify their analysis assumptions and inconspicuous trends hidden in user behavior data.

CCS CONCEPTS

• **Information systems → Database query processing**;

KEYWORDS

Cohort Analysis; Cohana; Query Processing

ACM Reference Format:
Zhongle Xie, Qingchao Cai, Fei He, Gene Yan Ooi, Weilong Huang, and Beng Chin Ooi. 2018. Cohort Analysis with Ease. In *Proceedings of 2018 International Conference on Management of Data (SIGMOD'18).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3183713.3193540

1 INTRODUCTION

In an era where most user activities are electronically recorded, the data analysts can gain insights into user behavior by mining the large amounts of accumulated data. *Cohort analysis*, originating from social science[3], fits well into the analysis of user behavior by collectively exploring the impact of two factors, namely *social change* and *age*, which are considered to be the major source affecting user behavior. Typically, cohort analysis groups users into different cohorts based on how they perform certain activities of interest, and measures how the behavior of each cohort evolves with time thereafter. A real-world application of cohort analysis in health care is described below.

Example: *A hospital wants to know the side effects of a new medicine A on patients cohorted by different physical ages who are diagnosed with disease B. The patient is monitored after taking the medicine at least 2 times by observing abnormal values in daily-conducted lab-test C.*

However, cohort analysis brings in two challenges that standard analytic techniques, such as SQL query, cannot easily overcome. First, it is difficult and time consuming to compose correct SQL queries for cohort analysis, especially for those without a strong database background. Second, due to the complex GROUP BY and JOIN operators involved in the query, the efficiency of leveraging traditional database for cohort analysis is extremely low [4].

We have recently proposed COHANA [2, 4] to tackle the second challenge. This work encompasses the design of cohort operators and implements an efficient cohort query processing engine. It defines cohort query in a more concise and accurate standard than its SQL equivalent, and the specially designed storage manager and query executor in COHANA grants performance superiority against traditional database systems.

However, the complexity of building the cohort query, which is another challenge for data analysts using traditional SQL tools, is still unaddressed by COHANA. According to our experience of industrial deployment in collaboration with multiple organizations, the major complexity lies in understanding the various terms and options that are involved in the queries submitted to COHANA.

To address this issue, we propose a web tool that runs on top of COHANA to provide with an intuitive and practical cohort analysis service. It lets analysts determine the parameters of cohort analysis by selecting options described in natural language on the web page, instead of writing queries required by COHANA. This tool is especially helpful for those who lack the knowledge of composing the cohort analysis in pre-defined terminologies. Furthermore, the result of the cohort query is visualized, which plays a key role for analysts to evaluate their queries and generate business reports. Generally, the ultimate goal of the tool is to help analysts gain insights into the data swiftly, accurately, and intuitively.

In this paper, we give an overview of our system by introducing extended key concepts in cohort analysis along with the system architecture first. We then demonstrate the entire analysis process in the context of the running example, including data preparation, cohort selection, and result visualization. Although COHANA is a general tool for cohort analysis, we focus on the medical area in the rest of the paper for better illustration and explanation.

SIGMOD'18, June 10–15, 2018, Houston, TX, USA

ACM ISBN 978-1-4503-4703-7/18/06…$15.00
https://doi.org/10.1145/3183713.3193540

2 SYSTEM OVERVIEW

In this section, we start with three key definitions presented in all cohort queries, and then depict the overall system architecture.

2.1 Query Definition

In cohort analysis, the behavior a user performed is called "event", which is usually recorded as a string column in database. There are four key concepts presented in the COHANA paper [4], i.e., **User Birth**, **Cohort**, **Age** and **Metric**. However, to define the cohort query more powerfully, they are further extended as following:

User Birth: A user is selected into the cohort query result if and only if this user performs a given sequence of events. This selection is called "user birth" and the given sequence is called "birth events" accordingly. It is worth noting that a sequence of events, instead of a sole event, is defined to make the selection more precise and comprehensive. For the running example, the birth events are defined as taking medicine A twice.

Cohort: Cohort refers to a group of users sharing certain common characteristics when born. In other words, a user is born right after completing the birth events and selected into a cohort accordingly. For the running example, the patients who perform the birth events are cohorted by one of their common characteristics, namely the physical age.

Age: The age of a selected user is defined as the number of time units passed since its birth. Two types of time units are supported in the query. The first one is calendar time unit, such as day/week/-month. The other age type is defined by the occurrence of a certain event. For example, we can use the admission to define the age of patients such that a patient is at age 1 before the first admission since born, and he will be at age 2 if unfortunately he is sick again and gets re-admitted by the hospital.

Metric: Metric refers to the result of user-defined calculation function, denoted as metric function and performed on the generated cohorts within a same age. Users can define metric function as simple aggregations like counting the number of occurrences of a particular event, or as complicated validation like fitting a probability distribution. For example, we can check the retention rate of patients' re-admission to the hospital by counting patients' admission records every month (i.e., one unit of age) after being discharged at the first time (i.e., the birth event).

2.2 System Architecture

The architecture of the system is depicted in Figure 1. The input is the data in the green rectangle. There are two components of the system, i.e., the COHANA engine and the web interface, which are respectively surrounded by a blue and an orange rectangle.

The input of the system is a file containing all the user records in CSV format. Besides user behaviors and the associated timestamps, the file can also include user details and other information. The data records of a single user are clustered together in chronological order. This sorting alignment for the records is a ubiquitous format for data analysis, and it aids the COHANA engine in the detection of the tuples containing birth activities of users.

The COHANA engine [4] consists of a parser, a catalog, a storage manager and a query executor, wherein the last two components

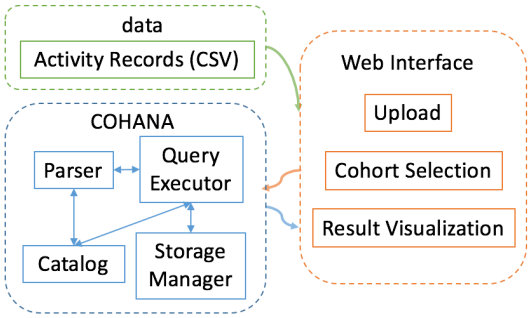

Figure 1: System Architecture

are the cores to support efficient cohort analysis queries. The storage manager organizes the dataset into a hierarchical layout such that the dataset is first horizontally split into equal-size chunks. These chunks are then stored in a columnar manner. To save storage space without sacrificing query performance, each chunk column is further compressed such that the compressed values can be read directly without decompression. The query executor generates a logical query plan where a birth selection operator, which finds qualified users and allocates them into cohorts, is pushed into bottom layer for early execution and minimizing the number of user behavior records for processing. Due to the optimized storage layout and natively implemented cohort operators, COHANA improves the performance of cohort analysis by a factor of orders of magnitudes compared to traditional methods.

The interactive web interface for COHANA is implemented via Django[1] framework with Gentelella[2]. We employ ECharts 2.0[3] to support chart manipulations in our system. The web interface is responsible for all frontend interactions such as uploading datasets, constructing queries, and visualizing results. It is also in charge of backend data transfer and interactions with COHANA engine.

To conduct the cohort analysis, one needs to first import a user behavior dataset by specifying its URI in the frontend. The backend then retrieves the dataset and extracts the needed column information for COHANA engine to compress the dataset. After compression, this dataset is ready for cohort analysis. To that end, one needs to decompose an analysis task into four components: user selection, cohort definition, age specification and metric calculation. This decomposition is done easily with simple yet clear options provided in the frontend, as we shall see later. Next, the backend converts the options selected into the required query format, and passes the resultant cohort query to COHANA engine for execution. Finally, the processed result returned from the engine is parsed by the backend, and then visualized, in the frontend, into various tables and charts, to help analysts gain insights of the data.

We add several new functionalities to COHANA system to further make the query result much more readable and traceable. For instance, one can name the cohort, or the set of cohorts, generated by the query, for further usage. This is helpful, for example, in finding the effective target patients of a new medicine. A practical use case can be: the doctor first discovers the patients for which a medicine of interest is effective, and then groups those patients

[1] https://www.djangoproject.com
[2] https://github.com/puikinsh/gentelella
[3] http://echarts.baidu.com/echarts2/index-en.html

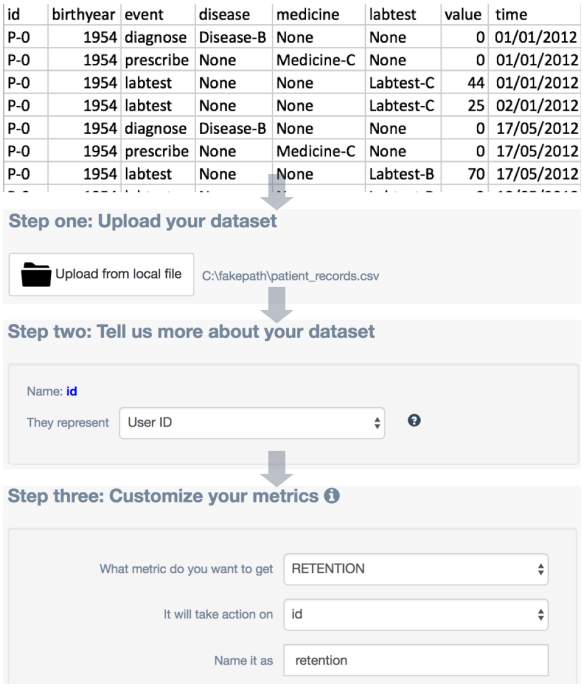

Figure 2: Data Upload

into different cohorts based on their diseases. Then, the doctor can look into one specific disease and further profile the patients in this disease cohort in terms of age, gender, diagnosing histories, etc.

3 DEMONSTRATION OUTLINE

In this section, we elaborate data preparation, query selection, and result visualization with the medical example in Section 1 to explain cohort analysis process using our system.[4] This example is quite representative in the medical context. Although the data we use for this demonstration is not real due to data confidentiality issues, we follow the schema of the real health care records and the distribution of the original dataset. It is notable that while the example resides in the medical domain, such analysis is common in other domains and can likewise be elegantly handled by our system.

3.1 Data Preparation

There are eight columns in the CSV dataset, i.e., *id, birthyear, event, disease, medicine, labtest, value, and time. id* is a unique identifier for each patient. *Event* contains the real affairs ("diagnose", "prescribe" and "labtest") the patients experiences. *Disease* contains a particular code for the illness diagnosed when a patient is admitted if the *event* is "diagnose". The *medicine* column indicates the prescription issued to the patient. The *labtest* refers to the type of tests patients take and the *value* presents integral test result.

An example in the dataset is shown in Figure 2. The patient with id P-0 was admitted to the hospital on 1st January 2012, diagnosed with Disease-B, prescribed Medicine-C, and scored 44 on Labtest-C. On the next day, he scored 25 on the same lab test. It should be noted that the redundancy in the records, which is common for real medical scenarios, has little impact on storage consumption and

[4]Demo Video: https://youtu.be/r28_jBD9qKg

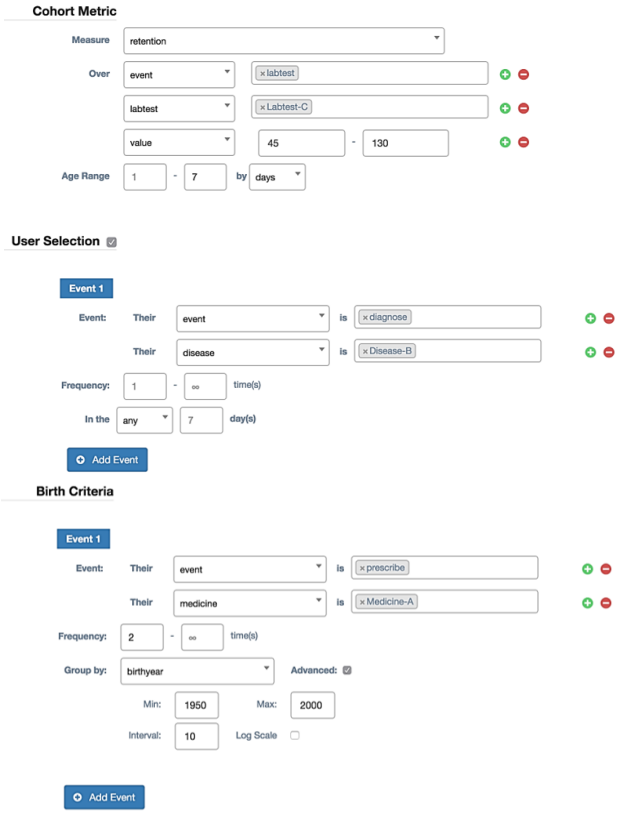

Figure 3: Cohort Selection

query efficiency due to customized storage layout enforced by the storage manager of COHANA.

The first step to conduct cohort analysis is to upload the CSV file and specify the schema of the data and the metrics to be evaluated, as shown in Figure 2. Here, we choose RETENTION as our metric to reflect the number of qualified patients in each age since their birth. The web interface allows for assigning multiple metrics, such as choosing COUNT on the labtest column if the number of abnormal values is needed, or using SUM for the total occurrences of abnormal values. After COHANA engine receives the dataset and its specifications, the web interface navigates to the cohort analysis page which provides an intuitive interface to specify analysis tasks.

3.2 Conducting Cohort Analysis

According to the running example, a decomposition of the natural language can be formed: 1) only patients diagnosed with disease B are needed for the analysis; 2) the birth events for patients are taking medicine A twice; 3) the time unit of age is one day; and 4) the metrics are collected by counting patients with abnormal values in lab-test C. Following the decomposition, the options provided on the web page can be chosen easily, as shown below.

As depicted in Figure 3, in "Cohort Metric" panel, we measure the *retention* defined in last step over patients who experience event "labtest" with a "Labtest-C" score in the range of (45, 130]. For each patient, the measured period, namely the range of the age, is the following 7 days after taking medicine A twice. This range defines the time period over which the patient behavior is measured. A

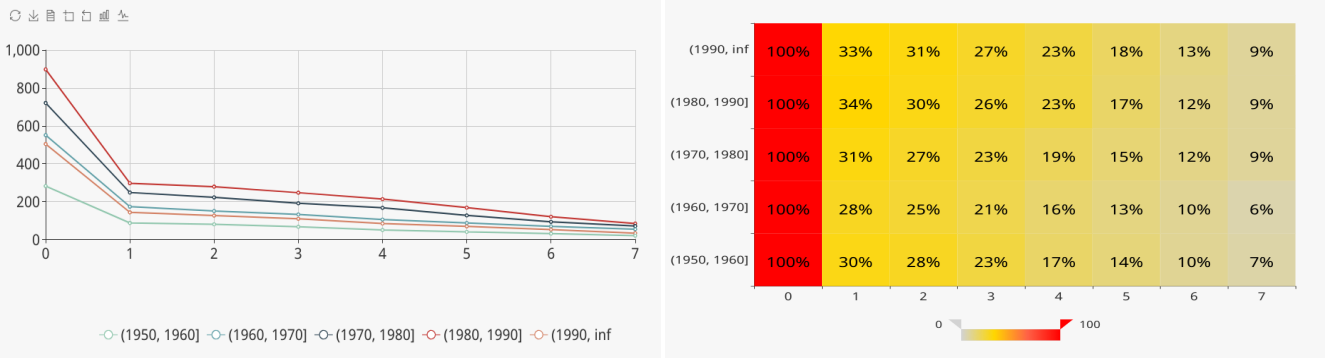

Figure 4: Result Visualization

small range can be chosen for short-term effects while a large one can be used for long-term effects.

The next selection is to decide the patients to be included in the analysis, which is specified in the "User Selection" panel. By default, all patients will be included. In Figure 3, those who experienced event "diagnose" with diagnosing result "Disease-B" within any week are specified as the target patients. More constraints on patient selection can be cast by ticking the + button on the "Event" line.

For the "Birth Criteria" panel, birth event "prescribe" and the corresponding medicine "Medicine-A" are specified. In addition, a qualified patient selected into a cohort should take the medicine at least twice since the frequency of this birth event is specified to be larger than or equal to 2. If multiple birth events are assigned by ticking the + button, a patient will be selected into cohorts only when all the birth events are met. In the last part of this panel, it is indicated that the qualified patients should be grouped by their "birthyear" in the range of *1950 to 2000* with a scale of 10 years.

3.3 Result Visualization

Within seconds of submitting the cohort analysis task, the analysis result is returned and visualized in various ways in frontend to help doctors better understand it. Here we choose a line chart and a heat map, as shown in Figure 4. In the line chart, the x-axis represents the *age* of the patients since their birth, i.e., the number of days after taking medicine A twice, and the y-axis represents the aggregation result, i.e., the number of patients with abnormal values detected for Labtest-C. The value of y-axis on age 0 is the total number of patients selected into the cohort. Each line stands for a cohort of patients that were physically born in a same decade. This line chart, answering the cohort query numerically, not only illustrates the trend of patient behavior along the time axis, but also offers a view of the difference in the behavior of different cohort patients.

The heat map is presented along age and cohort group dimensions. A cell (i, j) in the map represents the proportion of patients in cohort i that are detected with an abnormal Labtest-C value at age j. Having cells be shaded with different color depths in accordance to its metric value gives spontaneous expression on the evolvement of patient behavior in terms of Labtest-C result, and indicates deep insight into patient behavior between different cohorts.

The two charts explain the result of the submitted cohort query in an absolute and relative manner respectively. We can observe that younger patients are actively exhibiting side effects, as suggested by the high values in group (1980, 1990] in the line chart, while elder patients take longer to get accustomed to the medicine.

Additionally, with the help of ECharts, we provide a wide variety of operations to manipulate the chart, such as zooming in and changing the chart type, as shown in the toolbar above the line chart. These functions enable immediate responses to visualization exploration, which is helpful for further data excavation.

4 RELATED WORK AND CONCLUSION

Recently, due to the increasing volume of web user behavior data, cohort analysis has been introduced to mine unusual user behavior and improve user retention as proposed in [1, 5, 6]. However, these products directly follow the single birth event specification, restricting the diversity and representativeness of the cohorts generated. Moreover, they all employ simple SQL-based approaches, which are much slower than COHANA.

This demonstration shows a powerful and comprehensive tool on top of COHANA for cohort analysis while keeping operations as simple and intuitive as possible. The extended cohort concepts cover a broad range of practical data analysis needed in many domains concisely. Benefiting from the high efficiency of the COHANA engine, analysts without any backgrounds can conduct reporting or verify their ideas in minimal time with a user-friendly interface. Finally, the visualization of query results provides deep understanding and insights into user behavior.

ACKNOWLEDGMENT

This research was in part supported by the National Research Foundation, Prime Ministers Office, Singapore, under its Competitive Research Programme (CRP Award No. NRFCRP8-2011-08). Zhongle Xie's work was partially supported by the National Research Foundation Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme (E2S2-SP2 project).

REFERENCES

[1] Amplitude. 2017. Amplitude Official Site. https://amplitude.com.
[2] Qingchao Cai, Zhongle Xie, Meihui Zhang, Gang Chen, H.V. Jagadish, and Beng Chin Ooi. 2018. Efficient Temporal Dependence Discovery in Time Series Data. *PVLDB*.
[3] Norval D Glenn. 2005. *Cohort analysis*. Vol. 5. Sage.
[4] Dawei Jiang, Qingchao Cai, Gang Chen, H.V. Jagadish, Beng Chin Ooi, Kian-Lee Tan, and Anthony K.H. Tung. 2016. Cohort query processing. *PVLDB* 10, 1–12.
[5] Mixpanel. 2017. The definition of retention. https://mixpanel.com/retention/.
[6] RJMetrics. 2017. RJMetrics Official Site. https://rjmetrics.com/.