

Leveraging Human Learning in Interactive Data Exploration

Sanad Saha
Oregon State University
Corvallis, Oregon

Arash Termehchy
Oregon State University
Corvallis, Oregon

Leilani Battle
University of Maryland
College Park, Maryland

sahasa@oregonstate.edu termehca@oregonstate.edu leilani@cs.umd.edu

ABSTRACT

As most users cannot precisely express their information needs over databases, Database Management Systems (DBMS) are not able to satisfy their information needs effectively. Recent study shows that users leverage reinforcement learning methods to better express their information needs in the form of queries over the course of their interaction with the DBMS[10]. Such interaction between a user and DBMS can be naturally modeled as a game with identical interest between two rational agents whose goal is to establish a common language for representing information needs in the form of queries. Also, recent approaches for characterizing low-level user interactions in visual analytic tasks[5, 1] inspired us to gain semantic understanding of user interactions with the Data Exploration System (DES) and changes in user strategy at different stages of exploration. Based on such a framework, we build a DES for complex data exploration tasks that adapts user learning during different stages of exploration accordingly and help users to find their desired patterns easily and effectively.

PVLDB Reference Format:

. . PVLDB, (): xxxx-yyyy, .
DOI:

1. INTRODUCTION

Structured and semi-structured data sources, such as graph and relational databases, store a great deal of high-quality information, which can be used to extract interesting and useful insights. However, users often do *not* have any clear idea about the precise characteristics of the patterns that they want to find. Also, they may *not* know the structure and content of the database well. Therefore, they cannot precisely express the queries that return their desired patterns. They usually have to explore the dataset to understand the properties of entities and relationships in the data and formulate the queries that retrieve their desired pattern. For instance, a user may be interested in finding similarities

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.

Proceedings of the VLDB Endowment, Vol. , No.

ISSN 2150-8097.

DOI:

between political events in certain regions. Since the information need is quite vague, the user has to explore the data and submit queries to get a general picture of political events, submit more focused queries about some specific events, and repeat these steps until she can formulate the query that effectively finds the interesting entities and relationships shared between these events. Users may have to go through several states of exploration learn about the data in various degrees of generality to find their desired patterns. This process is very challenging and time-consuming over large datasets and users may give up the exploration without getting their desired results. To make the matters worse, most users do *not* know formal query languages, e.g., SQL and prefer to convey their intents in easy-to-use but ambiguous forms, such as keyword or natural language queries, which are hard for a database management system to understand and satisfy.

As both the user and Data Exploration System (DES) have the same goal of returning the users' desire information, the user and DES would like to gradually improve their understandings of each other and reach a common rapport of communication over the course of various interactions and exploration tasks.

Let us consider a scenario where a novice analyst of Delta Airlines is analyzing airlines performance dataset from Bureau of Transportation Statistics¹ to write a report on service expansion opportunities in the US airports. Now the analyst has a vague idea about the entities which have direct correlation with performance evaluation of an airline and lacks understanding of the data at hand. So, she is unable to express her information need in a single query. Thus, she may learn more about the structure and content of the database as she submits queries and observes the returned results. Leveraging experiences from past queries, she will learn to plan her exploration to know about the overall structure of the data quickly and focus on exploiting the data for answers rather than exploration.

The DES may learn more about how the user expresses her intents by leveraging user's feedback on the returned results, e.g., her clicks on desired answers, types of data manipulation operations (aggregation, sort, group by, etc.) performed on the attributes. The DES may also learn how the user explores the data and in what stages she looks for a general picture of the database and when she looks for a more specific and focused pattern(s). For example, in the aforementioned scenario, if we find out the user is using the

¹<https://www.transtats.bts.gov/>

same set of entities and just swapping the value of 'airport', it gives us the semantic meaning that the user has found out a way to test the hypothesis and checking possible expandability of every airport in the database; thus exploiting data.

Ideally, the user and DES should establish as quickly as possible this common understanding and plan of collaboration in which the DES accurately understands the intent behind all or most user's queries and can guide her to the parts of the data that may interest her the most. In each interaction, the user and DES receive certain reward according to how much the returned results are useful for the user. The user receives her reward by consuming the relevant information and the DES becomes aware of its reward by observing the user's feedback on the returned results and whether she has leveraged the information provided by the DES to formulate her next query. Hence, this interaction is a collaboration between two rational agents, which is naturally modeled as a game of identical interests.

Current interactive data exploration systems generally ignore the impact of user learning in exploration and assume that users do *not* modify their methods of exploring data. For example, they may assume that the user may follow the same set of states and stages of exploring data with fixed probabilities or the user select queries to express an intent according to a fixed probability distribution. However, it is known that a learning algorithm that is designed for a static setting may not predict accurate results in a dynamic environment. Thus, interactive data exploration systems that assume users to follow a fixed strategy in expressing their intents may *not* accurately learn and adapt to the changes in the user's strategy in exploring data.

It is time to develop a data exploration system that effectively interacts with intelligent humans! Our recent research using real-world keyword query workloads indicate that to express specific and focused intents, users select the keyword queries proportional to their past successes and this learning behavior is accurately modeled by a well-known algorithm widely used in economics to model human learning [10]. We have proposed a learning algorithm for DS that adapts to this learning behavior and outperforms keyword query answering systems. Our work has been selected as one of the best papers in SIGMOD 2018. Building on the success of this work, we plan to build an interactive data exploration system for complex and exploratory information needs that understands and adapts to the changes of the users' strategies in data exploration. We also extend on the effect of user learning during different phases of data exploration. From studies conducted on visual analytic systems we find out the users behave differently during different phases of data exploration [7, 5]. We build a **four** state user learning model that gives us semantic understanding of users' mental state while exploration and adaptive user learning strategy based on those states.

Towards this end, this project will contribute: 1) a model of users' learning in complex and exploratory information needs based on empirical user studies; 2) a reinforcement learning algorithm for the DES that answers users' queries based on the users' current states of explorations and adapts to user learning behavior; 3) efficient implementations of the aforementioned algorithm over large-scale databases; and 4) an experimental validation of the efficiency and effectiveness

of the algorithm on large real-world databases.

2. RELATED WORKS

Traditional Database Management System requires the users to have knowledge about the structure, content on the database which causes hindrances in data exploration tasks; especially, when the user does not know where to look for the desired information precisely. We see different approaches to build such data exploration systems. Using user feedback on sample examples from the database to perceive user interest and steer the user toward interesting data area is such an approach [3, 2, 8]. But such systems rely highly on user feedback and requires the user to label a large number of data for better accuracy. [10] took a different approach of viewing the problem of data exploration as a game between two agents: user and database; where the user uses reinforcement learning methods to build a mutual language of interaction with the database.

We see the characterization of low-level user actions into high-level blocks to create better visualization systems. But these high-level blocks are task dependent. ForeCache [1] uses such modeling to assist users with efficient and accurate exploration in tiles data. In [4], high-level states help to recommend better visualizations to the user and [5] defines pattern for insight provenance. Interactive visualization systems like VizDeck [6] uses statistical properties of the data for visualization recommendation; which inspires us with ways to find correlations among similar attributes.

3. FRAMEWORK

In this section, we lay down some foundations for our system and provide formal definition of the problem from a game theoretic approach.

Types of users: Depending on the users intent, we can separate the users of a data exploration system into two categories. One category contains users without any particular intent; just using the database to find something interesting. (e.g. people working with medical data to find out interesting facts). Another one has users who have precise intent but they don't know how to achieve it; due to their lack of knowledge on the database. Our data exploration system is geared towards the second category of users.

Formal representation of the problem from game theory perspective: Our first goal in this research task is to study formally the long-term interaction of a user or group of users and a DES over various exploration tasks. To simplify our discussion, we assume a database is a relation with a finite set of tuples (records) and each information need (intent) is a subset of these tuples. Our models extends to other types of databases, such as graph databases. In a simple scenario of user and DES interaction, we formulate this as a game played between two players: the *user* and the *DES*. At each (discrete) interaction time $t = 0, 1, \dots$, the user randomly chooses an intent $i(t) \in I$ where I is a finite set of intents. The intent expressed in each step is essentially a query formulated in a formal query language, such as SQL. As the user does *not* know the formal query language and the structure and content of the database well, she uses a natural language query $j(t) \in Q$, where Q is the set of finitely many natural language queries, to convey her

intent. Finally, the DES interprets the query $j(t)$ as an intent $\hat{i}(t)$ and returns its results.

Therefore, the strategy set of the user (at time t) is a stochastic mapping from I to Q , $U(t)$, and the DES uses a strategy $D(t)$ to interpret the intent of the user, which is a stochastic mapping from Q to I . We show $U_{ij}(t)$ as the probability that the user uses query j to express intent i and $D_{jk}(t)$ as the probability that the DES interprets intent k by observing query j at time t . For every $i(t) \in I$, we have $\sum_{j \in Q} U_{ij}(t) = 1$ and for every $j(t) \in Q$, $\sum_{k \in I} D_{jk}(t) = 1$. After each interaction, the user provides feedback on the returned intent. Using a probabilistic strategy helps the DES not remain biased towards the intents that receive positive feedback early in the interaction and provides opportunity to solicit feedback on other potentially relevant intents and accurately understand users' intents. We seek to study the behavior of the pair $(U(t), D(t))$ over various exploration tasks, i.e., as t goes to infinity. It is natural to assume that the payoff (utility) of a user using a strategy $U(t)$ and the DES using a strategy $D(t)$ are aligned and it is equal to the number of intents that are effectively answered by the DES. Formally, this common utility function is equal to: $u(U(t), D(t)) = \sum_{i \in I} \sum_{j \in Q} \sum_{k \in I} U_{ij}(t) D_{jk}(t) \mu(i, k)$, where $\mu(i, k)$ is the *satisfaction function* that measures how much the returned results of $k(t)$ satisfy the original intent $i(t)$. One may use well-known effectiveness metrics, such as precision@k, which is the number of relevant answers in the returned top-k results to measure this value. The DES may compute this value using user feedback.

Modeling user behavior through states: Researchers have recognized that users go through various states during each exploration task [4]. For example, users may start by scanning different fields in the data, i.e., *scanning* phase, to understand more about the structure of the data [4]. Then, they may check the content of the data by *checking* different queries and finally they will *focus* on their region of interest. The desired intent and reward structure of the user may depend on the state of exploration. For example, the user may prefer a more diverse set of results in her earlier interactions in the exploration task as these results will provide her with more information about the general structure of the information. On the other hand, she may like to see more focused results toward the end of her data exploration task. Thus, the returned results for a query should not depend only on the query but also to on the state of the exploration. Thus, we use a *different satisfaction function* for the payoff in each state of the exploration. For example, the satisfaction functions in earlier states of the interaction. e.g. scanning, may consider both the relevance and the diversity of the returned results. Thus, the user and DES should have different strategies per each state in the exploration.

User and DES state representation using POMDP: We denote the strategies of the user and DES and payoff in state s as U_s , D_s , and u_s , respectively. The DES should follow a Markov Decision Process whose states reflect the state of exploration that the user follows in each task. Since the DES returns results based on its observation of user's states, the user may also follow Markov Decision Process. More precisely, since the DES (user) cannot observe the current state of the user (DES), we model the DES and user process in a single exploration task using a Partially Observ-

able Markov Decision Process (POMDP)[9]. At each state, the DES and user would like to maximize the expectation of the discounted reward $E(\sum_{0 < t} \gamma^t u_s(U_s(t), D_s(t)))$ where s is the state of the exploration at time t and $0 \leq \gamma \leq 1$ is a discount factor that reflects how much users care about immediate payoff. Contrary to two-player games, in our system the two agents: user and DES will cooperate with each other to maximize profit by sharing decision states. POMDP has two more parameters than MDP: Belief (β) and Observation (Ω) defined over s .

We have shown that in a non-exploratory query answering, users both *exploit* their past experience and *explore* novel alternatives when formulating their queries and updating their strategies [10]. They exploit their knowledge of their past interactions with the DES and construct queries by picking keywords that have successfully expressed their intents in the past. They *explore* using keywords and queries that have *not* been sufficiently used in the past to find queries that express their intents more effectively. We leverage our work to study the user learning behavior in exploratory interaction.

4. METHODOLOGY

In this section we briefly describe the phases which we would execute in order to build a Data Exploration System (DES) that can leverage human intelligence. Then we would test its empirical effectiveness and efficiency on real life data exploration tasks.

4.1 Building User Exploratory Learning Model

Using an empirical user study, we plan to determine the states, users' follow in data exploration tasks. We plan to characterize low-level user interactions into high-level states with semantic meaning. Based on [4], we mapped the set of low-level user interactions into four high-level states: Scan, Flip, Swap, and Drill-Down. These high-level states give us the semantic understanding of user-actions; thus user decision process in exploratory systems. During Scan state, the user only concentrates on exploration; using low-level interactions to explore different parts of the database for details. In Flip and Swap state the user does both exploration and exploitation, while in Drill-Down the focus is completely on exploitation; user repeatedly filters down along orthogonal dimensions of the dataset to focus on subset of previously acquired data. We want to properly model the low-level interactions in order to discern users' need of exploration vs. exploitation. Also, we study the algorithms using which users update their transition probabilities in their decision process and modify their querying strategy in these states.

Now, we will informally define a users data exploration task with our aforementioned states. Consider a user, who is exploring the weather² dataset to find an answer to the question: 'What weather prediction would you make for February 14th, 2019 in Seattle and Why?'. Now, the user does not have any prior knowledge of the necessary entities to make the final decision. As a result, she will have to interact with the database to understand imperative information and make a decision. This exploration task can be decomposed to our four high-level states. We say the user is in Scan state when she is searching the database for relevant attributes that have impact on weather prediction

²<https://www.ncdc.noaa.gov/gHCN-daily-description>

(e.g. precipitation, rain, fog, date, year, state, etc.). After selecting the necessary attributes, the user might explore the weather condition in the month of February for preceding years (setting year values to 2018, 2017,...); which we address as Flip state. To confirm the hypothesis the user will try to find correlations between different dimensions which we refer as Swap state (e.g. finding correlations between precipitation and rain, fog and snow, etc.). In the Drill-Down state, the user will focus on specific dates and the city of Seattle. Though the user performs data comparison operations on both Flip and Swap states, in Flip state the user keeps the data object fixed whereas in Swap state the user works with different data objects. These states will provide DES with necessary information about exploration and exploitation needs of the user. Besides, users tend to learn at different rates in different stages. As a result, to effectively help users, the DES should first detect the current states of the user and return answers based on its strategy in the current state of the user. We use POMDP to model the states of the user.

4.2 Finding User Intent for Different States

Given the aforementioned decision processes for the user, we design an algorithm for the DES to estimate the current state of the user exploration and update its POMDP.

The DES should also update its strategy in each state based on the user feedback to learn to answer queries more effectively over time. We have proposed the following algorithm in [10] for understanding intents.

Reinforcement Learning Algorithm for Query Interpretation:

1. Let R be an arbitrary $n \times m$ positive matrix (i.e. $R_{ji} > 0$ for all $i \in I, j \in Q$)
2. For $t = 0, 1, \dots$ do
 - (a) Let $D_{ji}(t) := \frac{R_{ji}}{\sum_{i=1}^m R_{ji}}$.
 - (b) For the query $j(t)$ of the user, let $\hat{i}(t)$ be i with probability $D_{j(t)i}(t)$.
 - (c) Update the reward matrix R by

$$R_{j(t)\hat{i}(t)} := R_{j(t)\hat{i}(t)} + \mu(\hat{i}(t), \hat{i}(t)).$$

An interesting feature of the algorithm is that it provably improves the utility of players over time [10]. This result provides a convincing starting point to study reinforcement learning on user and DES interactions in data exploration. By changing the reinforcement values, we can maintain the trade-off between exploration and exploitation, according to user state requirement. For example, in Scan state, lowering the reinforcement value can make the DES system more interested in exploration. Since in each state, we use a different satisfaction function, we plan to extend the aforementioned algorithm for each state.

4.3 DES Learning Algorithm

In this phase, we plan to extend the aforementioned learning algorithm for different states of the DES POMDP. Which is a challenging task given that we extended POMDP for our DES. For example, the Observation (Ω) parameter of POMDP makes observations about *exploration vs. exploitation*. Our previous discussion on users learning model motivates us to expand Ω for different states to properly handle user queries. When our POMDP states that the user is in Scan state, based on the Ω value we can recommend more interesting attributes relevant to the user based on the Belief (β) DES learned from the user. We can find relevant

attributes using correlation matrices such as Kurtosis, Coefficient of Variation, Periodicity[6]. We also plan to study the convergence properties of the proposed DES learning.

4.4 Effect of User Learning

Given the aforementioned decision processes for the user and DES and the user learning model in Task 4.1, we want to define the asymptotic behavior of the user and DES interactions. In particular, how does $E(\sum_{0 < t} \gamma^t u_s(U_s(t), D_s(t)))$ evolve as $t \rightarrow \infty$?

Since the set of possible queries and intents are very large, the DES cannot efficiently maintain the reward matrix for our proposed algorithm over a large database. Hence the DES should quickly compute the probabilities and sample k most likely intents to keep the user engaged.

4.5 Efficient Implementation of the Query Interpretation Algorithm

Efficiently implementing the reinforcement learning algorithm over a large database is a difficult task. So, we will model queries and tuples using a smaller set of features, e.g., n-grams. After the user reinforces some tuples for a query, we reinforce their common features. The DES learning and query answering algorithm randomly samples k intents according to their probabilities due to its exploitative nature. We extend current sampling techniques over databases to efficiently return intents.

5. REFERENCES

- [1] L. Battle, R. Chang, and M. Stonebraker. Dynamic prefetching of data tiles for interactive visualization. In *Proceedings of the 2016 International Conference on Management of Data*, pages 1363–1375. ACM, 2016.
- [2] A. Bonifati, R. Ciucanu, and S. Staworko. Learning join queries from user examples. *TODS*, 40(4), 2015.
- [3] K. Dimitriadou, O. Papaemmanouil, and Y. Diao. Explore-by-example: An automatic query steering framework for interactive data exploration. In *SIGMOD*, 2014.
- [4] D. Gotz and Z. Wen. Behavior-driven visualization recommendation. In *Intelligent User Interfaces*, 2009.
- [5] D. Gotz and M. X. Zhou. Characterizing users’ visual analytic activity for insight provenance. *Information Visualization*, 8(1):42–55, 2009.
- [6] A. Key, B. Howe, D. Perry, and C. Aragon. Vizdeck: self-organizing dashboards for visual analytics. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 681–684. ACM, 2012.
- [7] H. Lam, M. Tory, and T. Munzner. Bridging from goals to tasks with design study analysis reports. *IEEE transactions on visualization and computer graphics*, 24(1):435–445, 2018.
- [8] H. Li, C.-Y. Chan, and D. Maier. Query from examples: An iterative, data-driven approach to query construction. *PVLDB*, 8(13), 2015.
- [9] J. Luo, S. Zhang, and H. Yang. Win-win search: Dual-agent stochastic game in session search. In *SIGIR*, 2014.
- [10] B. McCamish, V. Ghadakchi, A. Termehchy, B. Touri, and L. Huang. The data interaction game. In *SIGMOD*, 2018.