

ShiftScope: Adapting Visualization Recommendations to Users' Dynamic Data Focus

Sanad Saha*
Oregon State University
sahasa@oregonstate.edu

Leilani Battle
University of Washington
leibatt@cs.washington.edu

Nischal Aryal*
Oregon State University
aryaln@oregonstate.edu

Arash Termehchy
Oregon State University
termehca@oregonstate.edu

ABSTRACT

Visualization Recommendation Systems help users discover important insights during data exploration. These systems should understand users' exploration behaviors and goals to suggest relevant visualizations. However, users' mental models constantly evolve as they learn more about their data or as their personal or organizational goals change, leading to shifts in their data focus. Current systems do not adapt to these changes; therefore, they may inevitably suggest irrelevant visualizations over time. Thus, we introduce ShiftScope, an interactive system that recommends personalized visualizations while adapting to users' conceptualization of data. ShiftScope utilizes a dual-agent reinforcement learning framework, where one agent adapts to evolution in data focus and collaborates with the other agent to recommend the best visualizations to satisfy users' current and future exploration needs.

KEYWORDS

Exploratory Data Analysis, User Modeling, Reinforcement Learning, Personalized Visualization Recommendation

ACM Reference Format:

Sanad Saha, Nischal Aryal, Leilani Battle, and Arash Termehchy. 2018. ShiftScope: Adapting Visualization Recommendations to Users' Dynamic Data Focus. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 4 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 INTRODUCTION

As datasets continue to grow, data-driven discovery becomes paramount for solving pressing societal and business challenges. This interactive process involves users exploring new datasets to uncover crucial and interesting insights [1, 8]. It is particularly challenging for users performing open-ended exploration tasks with vague exploration goals that give them minimal guidance on their objectives

*Both authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference acronym 'XX, June 03–05, 2018, Woodstock, NY
© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/18/06
<https://doi.org/XXXXXXX.XXXXXXX>

and how to achieve them. Thus, users do the hard lifting of generating hypotheses, analyzing the data, and deciding how to proceed with the exploration problem (Figure 1). As users gain insights from their data interactions/queries over time, what they want to learn changes, dictating a shift in users' data focus.

Visualization recommendation (VisRec) tools alleviate the insight discovery process for users by suggesting and personalizing visualizations suited to their intent or task [1, 6]. The predominant methods for inferring user intent involve building a taxonomy of intents, i.e., tasks, and predicting which of these predefined tasks the user will switch to next. Then, visualization tools provide recommendations for what to explore next. However, task taxonomies are too brittle to account for spontaneous tasks that may appear during data exploration. Additionally, current VisRec systems fail to account how the user's understanding of the data, and thus, their data focus evolves over time, causing the underlying models and task taxonomies to quickly become outdated.

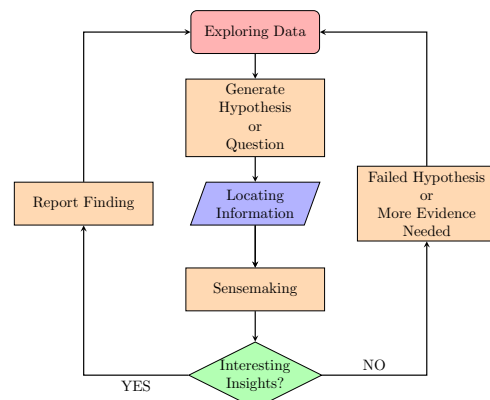


Figure 1: Example of an exploration approach

During exploration, users may learn about the data and change their exploration strategies to be more effective. For instance, users may revise their prior beliefs [15], shift their focus to different data attributes [10], and change the queries they submit to the system [9]. Similarly, users may deepen their understanding of the data through visualization recommendations at different steps of exploration (Figure 1), bringing the user to a new goal or insight [4]. The following scenario illustrates how users *shift their data focus*, i.e., change their data selection and transformation preferences, based on what they learn over time.

Scenario: Alice and Bob are exploring the Birdstrikes dataset [16] containing records of wildlife strike incidents with aircrafts. Alice

is exploring opportunities to improve aviation safety and reduce the losses incurred by airlines. Bob is researching the current safety of wildlife behavior and migration patterns. Both explore the same visualizations for their first three interactions at time t , $t+1$ and $t+2$ (see Figure 2). Interaction t reveals the number of strikes per month, $t+1$ displays the total strikes at different times of the day (e.g., Day, Dusk, Night), and $t+2$ depicts total strikes in different phases of flight (e.g., Climb, Approach).

However, Alice and Bob start to deviate at time $t+3$. Alice analyzes the Total **Cost** incurred for strikes happening at different phases of flight whereas Bob investigates the number of strikes involving different **Sizes** of wildlife (e.g., Small, Medium, Large).

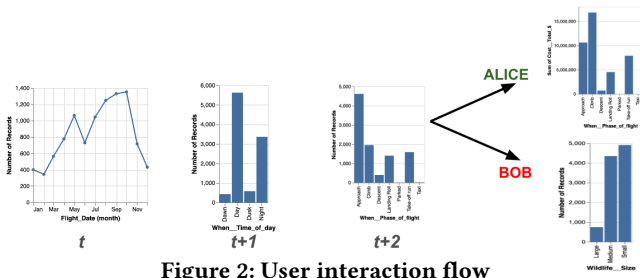


Figure 2: User interaction flow

Shifts in Data Focus: Alice spends more time on interactions $t+2$ and $t+3$ and learns: ‘Although the number of incidents during Approach is higher, incidents during the Climb phase are more costly to the airlines’. Thus, she will focus more on the attribute **Cost**, rather than ‘Number of records’ in the upcoming phases.

In contrast, Bob spends more time on interaction t . He relates the high number of birdstrikes in September and October with birds’ migratory patterns. He is not interested in the costs of airlines but rather wants to know which species are more at risk. So, he prioritizes visualizations involving attributes such as ‘wildlife_species’ and ‘wildlife_size’ (e.g., interaction $t+3$ (Bob)).

Challenges: Current techniques assume we can pre-train models to learn all relevant tasks and behavioral patterns offline. For example, rule-based [12] or ML-based [6] VisRec systems either statistically measure or utilize learned models to find the best possible visualizations given users’ current state. However, data exploration is inherently dynamic. So, these models will fail during tasks they have never seen before, which is quite common during data exploration. So, relying on static methods that recommend visualizations based on common or popular features may not accommodate sudden shifts in users’ data focus and aid users’ long-term objectives.

Our approach: Rather than trying to model every aspect of the user’s learning process, which is out of scope for this paper, we focus on modeling a key aspect of the user’s cognition: when they *shift their data focus*. In this way, we move beyond rigid task prediction towards understanding *when* and *how* the user shifts their focus to different parts of the data.

So, in this work, (1) we model users’ attribute preferences as indicators of their data focus during exploration. Additionally, existing VisRec systems typically allow users to choose data attributes to guide their exploration, generating various visualizations based on these choices.[3, 17]. ShiftScope employs a state-of-the-art reinforcement learning (RL) agent to model users’ attribute preferences.

Pretrained on past users’ exploration sessions, this user model agent can adapt and simulate a particular user’s future data focus, leveraging that user’s feedback and interface operations.

(2) We represent visual-data analysis using a dual-agent Markov decision process framework and implement it in ShiftScope. The user model agent lets the RL-based recommender agent know about potential shifts in users’ data focus. In addition to this information, the recommender agent uses explicit user feedback on past recommendations to recommend six visualizations. Both agents collaborate so that one can accurately model the data shifts, and the other can utilize that to generate visualizations. ShiftScope displays the data focus simulations for the user as visualization recommendations.

2 SYSTEM DESCRIPTION

2.1 Technical Background

2.1.1 Exploratory Data Analysis (EDA). EDA initiates when the user loads a dataset into ShiftScope. It recommends some visualizations to avoid a cold start, i.e., starting the exploration without any prior goal or knowledge. These recommendations are designed to encapsulate interesting starting points derived from other users’ interactions. It helps the user to start the EDA process as in Figure 1.

In ShiftScope, users can either select a recommended visualization or construct a new one for analysis. This chosen/created visualization plays a pivotal role in shaping both the user’s subsequent data focus as it acts as the basis of ShiftScope’s future recommendations.

2.1.2 Markov Decision Process (MDP). MDP is a framework for modeling decision-making, where an agent learns to achieve its goals through repeated interactions with an environment (interface). In each interaction, the agent takes action based on its current state (visualization), and the environment responds with a reward (feedback). The agent uses these rewards to update its policy and make better decisions in the future. In MDP, the agent’s objective is to find an optimal **policy** (π), which is a function that maps each state to an action that maximizes the expected future reward.

2.2 Dual-Agent Architecture

2.2.1 Formalized MDP. Our dual-agent MDP can be represented using a tuple $\langle \mathcal{U}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where \mathcal{U} is a discrete set of states shared by our two agents (U). User-agent ($U = *$) learns evolution in the user’s data focus, and the recommender agent ($U = !$) generates recommendations conditioned on this evolution. \mathcal{A} represents the set of actions available to each agent. $\mathcal{P} : (\mathcal{U} \times \mathcal{A} \times \mathcal{U}) \rightarrow [0, 1]^{|U|}$ represents the transition matrix, which determines the next state after all agents perform actions on the current state. $\mathcal{R} : \mathcal{U} \rightarrow \mathbb{R}$ is the reward received by each agent after performing an action, and what they observe in the environment is $\mathcal{O} : \mathcal{U} \rightarrow \mathbb{R}^{|U|}$. Formalizing EDA as an MDP enables us to use state-of-the-art RL algorithms for the decision-making process of our agents.

2.2.2 Defining MDP components in ShiftScope. **States (S):** The state corresponds to the current visualization the user interacts with, which is generated using Vega Lite [14] specifications. In this work, we represent the state by converting key visualization information, such as the mark type, selected attributes, attribute types, and applied aggregations, into a one-hot vector.

User Agent’s Actions (A): User-agent’s action involves selecting

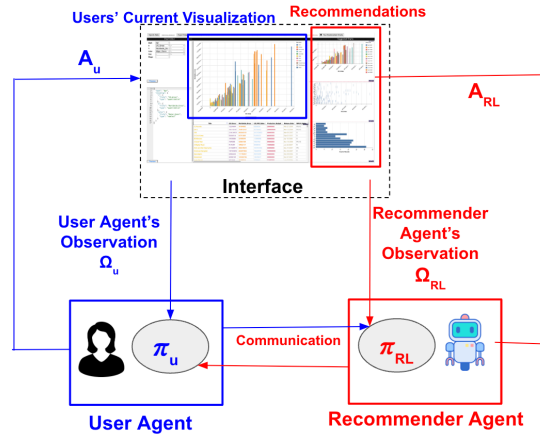


Figure 3: Dual-Agent Collaborative Framework

the attributes for the \cdot , \cdot , and $>$ fields.

User Agent's Reward r_u : The user agent gets rewarded if it can correctly predict attribute preferences, which leads the recommender agent to generate informative visualizations. However, it is challenging to quantify r_u . In EDA, it is not feasible for the user to let us know what data area they would like to focus on after every interaction. However, actions like bookmarks and SAT-clicks give explicit insight into the usefulness of particular recommendations. Moreover, this feedback is sparse, and to quickly adapt to users' needs, it would be beneficial to get feedback after every step on the recommendations. So, we take proxy measures (e.g., KL-divergence[2]) to determine the effectiveness/interestingness of the recommended attributes and generate r_u .

User Agent's Policy π_u : ShiftScope focuses on users' evolving data focus, given the current state (s, C) stochastically determines a possible set of attributes. These attributes may generate visualizations that users may find interesting to analyze in future interactions. C is updated based on the feedback from the recommender on user agent's performance in detecting data shifts and its own observations from the environment.

Recommender Agent's Actions a_{RL} : Recommending six visualizations to the user. However, on a low level, the recommender needs to predict a set of attributes for these six visualizations. Leveraging information received from r_u ensures the recommendations entail six potential directions where users' data focus may shift.

Recommender Agent's Reward r_{RL} : The recommender receives rewards (in the form of bookmarks, SAT-clicks, etc.) reflecting users' satisfaction with the recommendations. Furthermore, the recommender considers users' interactions, such as clicks and hovers over visualization items, as forms of reward. Early user feedback can be noisy, so ShiftScope employs a trained RLHF model on past exploration sessions, to generate initial visualizations, which are more likely to align with user interest in the early stages.

Recommender Agent's Policy π_{RL} : Given current state (s, C) stochastically determines the best visualizations to recommend. C is updated based on the attribute preferences suggested by the user model, user feedback, and recommender agent's observation (s, a_{RL}) . **Observation (o) :** Both the user and recommender agents monitor user interactions within the interface, utilizing observations like clicks, hovers, bookmarks, etc., to refine their respective policies.

2.3 Collaboration and Algorithm:

Without collaborating with the user agent, the recommender agent lacks the ability to generate insightful visualizations that align with the user's evolving data focus. If it fails to do so, it cannot garner positive feedback from the user, causing both agents to encounter two significant challenges: firstly, the recommender lacks the necessary guidance to create visualizations that align with the user's specific informational needs; secondly, the user agent renders ineffective, thereby undermining the recommender's capacity to produce recommendations tailored to the user's focused data areas.

In ShiftScope, both the user agent and the recommender agent is modeled using Proximal Policy Optimization (PPO), an actor-critic reinforcement learning algorithm variant. Recent studies have highlighted the advantages of PPO in modeling human learning within collaborative environments[5]. Leveraging available information from the interface and user feedback (subsection 2.2.2), the PPO-based user-agent model successfully adapts to users' data focus and lays the foundation for recommender to recommend adaptive visualizations. Additionally, PPO's popularity in systems that incorporate human feedback is well-established.

Effectively generating visual encodings from data is essential for users to comprehend information based on selected attributes. To achieve this, we employ the state-of-the-art tool Draco [11], which is embedded with visualization design knowledge. Draco aids in navigating the visualization design space, thereby enabling the identification of optimal visualizations for the selected attributes.

3 DEMONSTRATION

3.1 Dataset and Exploration Tasks

We will begin with an overview of our system, dataset, and exploration tasks. We will utilize two Voyager [16] datasets and 2 open-ended tasks to demonstrate our system.

birdstrikes: The dataset contains incident reports of aircraft (e.g., airplanes) striking wildlife (e.g., deer, birds), with contextual details (e.g., weather conditions, total struck, etc). The dataset is a redacted version of FAA wildlife airplane strike records with 10,000 records and 14 attributes (9 nominal, 1 temporal, 4 quantitative).

Exploration task: *Feel free to explore any and all aspects of the data. Use the bookmark features to save any interesting patterns, trends or other insights worth sharing with colleagues. Note the top 5 bookmarks that you found most interesting from your exploration.*

movies: The movies dataset contains 3,201 records and 15 attributes (7 nominal, 1 temporal, 8 quantitative). **Exploration task:** *What kinds of movies will be the most successful movies based on your observations of the data? Summarize the 2-3 characteristics that you believe are most important in predicting their success*

3.2 System Interface

Figure 4 presents our system interface. The top panel (A) provides data imports and exports and provides a quick view of data attributes. The data-panel (B) contains a more detailed view of the dataset. The chart-editor view (C) allows users to steer the exploration. Users can select the data areas to explore. Once users specify the data attributes, the lower panel presents the automatically generated Vega-lite specification. Users comfortable with Vega-lite can also edit this chart specification. Current-analysis view (D) renders the user-specified visualization. The suggested charts view (E)

