VisGuide: User-Oriented Recommendations for Data Event Extraction

Yu-Rong Cao[1], Xiao-Han Li[1], Jia-Yu Pan[2], and Wen-Chieh Lin[1]
[1]National Yang Ming Chiao Tung University, Hsinchu, Taiwan
[2]Google, United States

Figure 1: We propose VisGuide, a data-exploration system that guides users to extract data events and represents them as visualization trees. (1) The *Tree View* presents the generated visualization trees; (2) The *Recommendation View* shows the next chart recommendations of the user-focused chart (Chart A), of which is marked with blue border in the generated visualization tree in the Tree View. (3) The *Sheet Management Bar* supports users to create multiple visualization trees by adding a new sheet. Users can also switch among the sheets to compare the explored results.

ABSTRACT

Data exploration systems have become popular tools with which data analysts and others can explore raw data and organize their observations. However, users of such systems who are unfamiliar with their datasets face several challenges when trying to extract data events of interest to them. Those challenges include progressively discovering informative charts, organizing them into a logical order to depict a meaningful fact, and arranging one or more facts to illustrate a data event. To alleviate them, we propose VisGuide—a data exploration system that generates personalized recommendations to aid users' discovery of data events in breadth and depth by incrementally learning their data exploration preferences and recommending meaningful charts tailored to them. As well as user preferences, VisGuide's recommendations simultaneously consider sequence organization and chart presentation. We conducted two user studies to evaluate 1) the usability of VisGuide and 2) user satisfaction with its recommendation system. The results of those

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06...$15.00
https://doi.org/XXXXXXX.XXXXXXX

studies indicate that VisGuide can effectively help users create coherent and user-oriented visualization trees that represent meaningful data events.

CCS CONCEPTS

• **Human-centered Computing → Visualization; Visual Analytics**.

KEYWORDS

Visualization Sequencing, Visualization Recommendation, Visualization Trees

ACM Reference Format:
Yu-Rong Cao[1], Xiao-Han Li[1], Jia-Yu Pan[2], and Wen-Chieh Lin[1]. 2022. VisGuide: User-Oriented Recommendations for Data Event Extraction. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 17 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Data exploratory analysis is an important process in data science. It has been used in diverse areas, both practical and theoretical, to help people make sense of data and make better decisions. Successful exploration of a dataset is usually an interactive and iterative process that requires not only effective analytical algorithms and visualization tools but ideally also analysts with adequate domain knowledge about the data. The iterative nature of such exploration generally requires people to make substantial efforts before discovering meaningful and valid data events. Accordingly, there has been considerable recent interest in developing recommendation systems that lessen the demands upon analysts, in terms of both domain knowledge and analytical effort.

Visualization recommendation systems are increasingly being deployed in systems for extracting interesting data events from raw data. By adopting existing definitions from the field of data storytelling [16, 17], we define data events as a data story's subcomponents, each of which is representable by a number of chart sequences, i.e., arrangements of multiple related charts (Figure 2). Our proposed system, **VisGuide**[1], is an interactive, personalized recommendation system for data-event extraction, with data storytelling being a key potential future application.

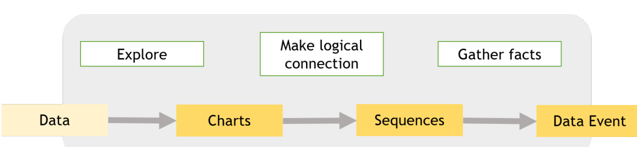

Figure 2: The process of creating data events, inspired by [17, 34].

The process of data-event extraction typically includes three steps: exploring, making logical connections between charts, and gathering facts (Figure 2). During the first step, the analyst identifies interesting patterns in the dataset and attempts to create charts to represent those findings. In the second step, after multiple charts have been created, the analyst may recognize logical connections between a few of them with overlapping topical content. At this point, a commonly used format of *chart sequences* [12, 13, 15] can be used to represent the logical connections among charts. Lastly, one or more chart sequences on the same topic can be gathered together and presented as a data event.

Anyone proceeding through the three steps of data-event extraction will face many challenges and must have the tools or systems necessary to facilitate the process. In this paper, we consider two frequently researched core issues [28] that need to be considered when creating such a system: *"How can it ensure that the exploration proceeds smoothly and continuously?"* and *"How can it satisfy its users' preferences?"*

To tackle these two core issues, we propose VisGuide, which has been designed as an interactive data exploration tool that features a user-oriented recommendation engine to guide users' data exploration by considering their preferences. The system aims to support the selection, organization, and presentation of visualizations collected during a user's exploration. Based on their interaction with VisGuide, the system progressively learns the user's preferences and provides "next chart" recommendations by taking into account (1) user preferences, (2) the statistical properties of the data in each chart, and (3) the relation between each pair of adjacent charts. We particularly focus on two common inter-chart relations, *comparison* and *drill-down* [12, 14, 18], and adopt a tree-like representation [7, 18] to visually present them. We call this representation the **visualization tree**, whose breadth expansion denotes the charts that are being compared and depth expansion lays out the charts that are connected by a drill-down exploration. We'll use "visualization tree" and "data event" interchangeably in this paper as the former is just a representation of the latter. Both of them are the output of VisGuide.

We conducted two user studies with real-world datasets to evaluate VisGuide's usability and verify its ability to make effective user-oriented recommendations. All the sampled users reported that VisGuide's expressive visualizations and its user-friendliness made it a better tool for generating insight-rich and coherent visualization trees to present data events than other tools they had used in the past. Our results also indicated that the proposed system's chart recommendations were meaningfully adapted to the participants' individual preferences and that they preferred its initial recommendations over those of a baseline system that did not provide user-oriented recommendations.

The main contributions of this paper are as follows. First, we propose VisGuide, an interactive data exploration environment that supports breadth and depth exploration by recommending comparison and drill-down charts and visually organizing the user-selected charts with a tree structure. Second, VisGuide provides user-oriented recommendations using an online-learning method that takes into account data statistics, visualization relations, and user preferences. Third, we design a comparative study that adopts *Normalized Discounted Cumulative Gain (NDCG)* [8, 21] to assess recommendation ranking quality, as well as propose the *Mean Round to Recovery (MRTR)* to measure how quickly a recommendation system can adapt to a user's preference changes. The design of our study and the proposed metrics could be applied to other future studies on evaluating dynamic recommendation systems.

[1]Open Source *https://github.com/littlehanli/VisGuide*

2 RELATED WORK

Systems for extracting data events typically contain a visual data exploration module. LisaTherefore, in this section, we begin by briefly reviewing related work in that area. Then, we survey related research on visualization recommendations and user-preference acquisition via interactive analysis.

2.1 Visual Data Exploration for Data Events

Existing data exploration tools often regard charts and sequences as basic prerequisites for forming data events (Figure 2). Conceptually, data events are very similar or identical to other constructs such as story slices [3], story pieces [3], microstructures [36], and sequences of panels [43]. In these cases, the main components of a data event are the charts, and optionally, text annotations. Here, we focus on data events consisting of chart sequences without text annotation.

Turning now to chart exploration systems, Tableau [22] and Power BI require their users to manually create visualizations (i.e., charts) and arrange them into an understandable order. After identifying a set of informative visualizations, Hullman et al. [12, 13] modeled the visualization-sequencing process as a directed graph and thereby minimized the cost of chart transitions from the audience's perspective. They also evaluated their auto-sequencing method in light of user-preference rankings.

In addition to the creation of charts and sequences, many cutting-edge studies have focused on the organization and presentation of data-exploration findings. For instance, GraphTrail [7] applied filtering (drill-down), pivoting, and cloning methods to analyze user-interaction networks by exploring nodes and edges. It used colored links to display exploration history, showing the different actions applied to subsequent charts, which can be regarded as a forerunner of the tree-layout presentation. Siming et al. [3] defined *story synthesis* as the process of constructing story content, which is essentially creating and assembling story slices. On that basis, they developed an excellent analysis tool. It is more suitable for experts who are familiar with both synthesis methods and the datasets they are using. DataShot [37], on the other hand, is more suitable for non-experts to learn interesting facts about a dataset. It pre-computes fact extraction and composes facts into a topic. A presentation is automatically generated when users select a theme.

Unlike previously proposed systems, VisGuide balances the flexibility of user authoring and the convenience of fully-automatic tools by offering its users insightful recommendations during the exploration process. Meanwhile, users can be guided to create a visualization tree representing a multi-sequence data event with personalized recommendations, which also considers their exploration history and adapts to their preference changes.

2.2 Visualization Recommendations

In the context of data-event exploration, recommendation systems aim to help users efficiently identify charts that will yield important or interesting insights about or arising from the dataset. The interestingness of visualizations can be measured in two ways: by assessing the interestingness of each chart separately, or by considering multiple charts holistically.

The first of these two measurement approaches utilizes the statistical properties of data, typically including the distributions of data subsets [23], correlations, properties of clusters, outliers, anomalies, and extreme points [33, 44]. Tools for automatically recommending visualizations that correspond to particular statistical properties of the underlying data have been proposed [4, 19, 35]. For example, Voyager2 [39] allows users to explore attributes and suitable visualizations, and then automatically recommends views related to the currently specified chart. However, none of these previous studies considered connections between pairs of adjacent charts, instead leaving the hard work of generating meaningful sequences of charts to users.

The second measurement approach analyzes the transitions between multiple charts by following the human cognitive process when inspecting a sequence of visualizations. Recommendation systems using this approach can be seen in many fully-automatic chart arrangement systems. For example, given a collection of charts, GraphScape [15] calculates the transition cost between two charts and, based on the cost, automatically orders charts into an easy-to-follow linear sequence. ChartStory [42] recommends the partitioning of a bundle of collected charts and then orders both local and global sequences into a comic layout. VisPilot [18] addresses the three challenges of exploration (safety, saliency, and succinctness) and automatically generates a set of informative and interesting visualizations to convey key insights. One focus of VisPilot is avoiding the fallacy of drill-down by considering data statistics and drill-down relations between two visualizations. Though VisGuide features a similar UI to VisPilot to show relations between charts, the tree structure in VisGuide is used to expand and rearrange on demand, while VisPilot is showing a calculated lattice.

Different from the aforementioned works, VisGuide supports a personalized recommendation mechanism that takes both data statistics and users' dynamic preferences into account to guide users to explore a dataset in breadth (comparison) and depth (drill-down), assisting users to discover data events flexibly and efficiently.

2.3 Extracting User Preferences in Interactive Data Analysis

The interestingness of a chart varies for different users according to their diverse intentions and exploration goals. To customize visualizations for users, recent studies have proposed methods based on online training. These methods build predictive models that learn a user's preferences based on the user's past interactions with the system. The basic idea is that a user first expresses their interests in a batch of charts. Based on these, models of user preference can be trained and used to recommend user-oriented charts [1]. Several studies [6, 11, 20, 27] have gauged users' interests based on classification models such as decision trees or support vector machines. Harvest [9] captures users' current behaviors as input and recommends visualizations that align with their presumed intentions. ViewSeeker [40, 41] is based on a linear regression model trained to estimate each user's ideal utility function. However, all of these approaches train their preference models based on pre-labeled information about the target user, and therefore cannot adapt when user preferences change during an exploration. When exploring new datasets with these systems, users must repeatedly engage in labeling work to build new preference models.

VisGuide, in contrast, captures its users' preferences with an online-learning approach, which can adapt the user preference model in response to mid-exploration preference changes. In addition, it incorporates a model-transfer mechanism to speed up the learning process and reduce labeling efforts when a user moves the exploration from one dataset to another.

3 DESIGN PRINCIPLES

From our pilot study and previous works [1, 33, 44], we acquired four design principles that aim to address our research problems regarding assisting the exploration process and satisfying user preferences during data event extraction. In the pilot study, we invited 10 participants and two experts in the fields of the datasets (air quality and department store transactions) to explore and create data events using VisGuide. During post-session interviews and individual brainstorming sessions with participants, they emphasized the importance of a semi-automated data exploration system and the benefits of the tree layout design, which led to the first two design principles of VisGuide (D1, D2). The third and fourth design principles (D3, D4) were adopted from the previous works on visualization sequencing. Our first user study (reported in Section 6.1.5) also verified the importance of these four design principles.

D1. Semi-automatic visual exploration guiding system. At a high level, the system should be able to intelligently direct its users to further their explorations with insight-provoking charts, but without overwhelming users. VisGuide should include a semi-automatic event-exploration system, i.e., an interactive environment that recommends small groups of meaningful charts and allows users to select those that they are most interested in working through (and perhaps beyond). One advantage of a semi-automatic system is that it can guide users to continue exploring when there are no clear exploration goals or when a bottleneck in the thought process is encountered.

D2. A contextual and expressive visualization layout for data events. Prior works on visualization sequencing and infographics [12, 13, 15, 18, 37] have indicated that human users tend to organize charts in some structural patterns to convey an observation (e.g., data event) from the data. However, arranging charts in a linear or tile layout has limitations when presenting rich relations among multiple charts. The pilot study reveals that the breadth and depth are the investigation directions that naturally emerged from data explorations. Hence, we propose that the visualization layout should adequately express the breadth (e.g., comparison) and the depth (e.g., drill-down) of an exploration. We adopt a tree layout [7, 18] to arrange the collected charts, reflecting the interrelation between these charts during an exploration.

D3. User-oriented recommendations. Each user has unique personal preferences when it comes to exploring a dataset. These preferences are manifested in their use of structural elements when building a data event. Moreover, a given user's interests may change during the course of data exploration. Therefore, the recommendations from a data event extraction system should ideally adapt at the outset to each user's individual preferences, and subsequently to variations in those preferences (Section 2.3). To achieve this, VisGuide incorporates each user's feedback interactively and implicitly, learning their personal preferred model in a progressive, online fashion (Section 5.3.2). In particular, by designing a mechanism of implicit labeling, VisGuide aims to collect enough user data to build their preference model while avoiding too many user interventions.

D4. Efficient exploration of data insights. Lastly, useful insight (Section 2.2) may only be gleaned through expressive visualizations or noticed via recommendation. To facilitate the exploration of data insights and the integration of the most relevant charts into data events, VisGuide provides visual hints about potential data insights such as extreme points and data trends (section 4.2). This would be particularly helpful for users who have not decided on a specific exploration target. Besides increasing the visibility and discoverability of data insights with effective visual hints, the data insights within a chart should also be considered by the system when ranking the candidate charts for recommendations (see chart features F1 and F2 in Section 5.3.1).

4 USER INTERFACE AND USER WORKFLOW

4.1 System Overview

VisGuide supports input datasets that are spreadsheets of multidimensional data, which are the most common type of dataset. These datasets can be modeled as a set of *measure attributes* that contain numeric values, as well as a set of *dimension attributes* that contain categorical values. In our implementation, data's quantitative attributes are classified as measures, while nominal and temporal ones are classified as dimensions.

VisGuide provides three basic types of 2D visualization: line, bar, and doughnut charts. It automatically determines the visual encoding for each chart according to a set of rules proposed in prior studies [21, 38]. For example, continuous valued data are presented using line charts, while percentage information is presented in the form of doughnut charts. Bar charts are used when the data's X channel is a nominal attribute.

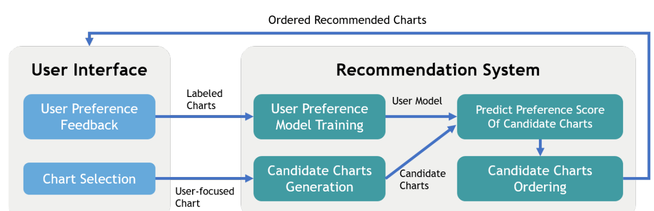

Figure 3: System overview. VisGuide consists of a user interface and a recommendation system. The user interface acquires the users' exploration intentions and preferences. The recommendation system generates and orders the candidate charts based on the predicted user preference scores, which are generated by an online-trained user preference model.

Figure 3 shows the architecture of VisGuide. It consists of two groups of components: a user interface and a recommendation system. The user interface contains a user preference feedback module and a chart selection module. The former collects the user's preferences based on their interactions with the system. These user interactions implicitly annotate charts and create a set of labeled charts that will be used for training the user preference model online. The later module allows the users to show their exploration intentions by clicking on the data points that interest them in the charts (called *user-focused charts*). This action will trigger the recommendation system to generate a set of candidate charts to be explored next based on the clicked data points. The preference score of each candidate chart is predicted using the user's preference model, and the candidate charts are then presented in descending order of preference score.

The above procedure, which consists of user interaction, chart recommendation, and online updates of the user preference model, is repeated during the exploration process. The charts selected during this process are arranged in a tree structure, with which the user can intuitively organize them into a systematic presentation that expresses the structure and patterns of a data event.

4.2 User Interface

VisGuide allows users to explore data events through simple operations such as selecting a chart from *"Recommendation View"* and organizing and tracing the exploration process in *"Tree View"*. Also, multiple visualization trees can be created by adding new sheets in the *"Sheet Management Bar"*. The main components of the user interface are described below.

Recommendation View (Figure 1, (2)) shows the recommended Drill-down and Comparison charts (Section 5.2) to explore next. The topmost chart of each of these two exploration operations is the highest-ranked chart according to VisGuide's user preference model. Users can scroll down within each section to find lower-ranked recommendations; click the "Star" button (Figure 1, (2)a) to bookmark the charts that are interesting to them, indicating their intentions of reviewing these charts later in the exploration (section 5.1); and click the "Add" button (Figure 1, (2)b) to select and insert a chart into a visualization tree (D1).

When a chart is added to a visualization tree, drill-down charts will be placed at the next tree level and linked in green, while comparison charts will be placed at the same tree level and linked in brown. This design enables users to understand and record the structure that connects the charts, especially when the visualization trees are complex (D2).

Tree View (Figure 1, (1)) shows a visualization tree generated from a user's exploration. A user can scroll to see different parts of a large tree. We provide a thumbnail image in the top-left corner to display the overall structure of a tree. Each chart has a navigation bar (Figure 4(a), A) and a visualization of the data (Figure 4(a), B). In each visualization, a dark blue line or bar represents a subset of data points according to specific filter/drill-down conditions (Figure 4(a), C) and a light blue line/bar that represents all data points right before the filtration (Figure 4(a), D). We call the dark line/bar "filtered" data and the light line/bar "overall" data. For more details about how to specify the filtered data subset from the current set of data points, please see Section 4.3. The difference between the filtered data and overall data presents the deviation information (D4). Users can click on the "Filter" button to see the full filter information for the chart. The data points with maximal and minimal values are marked with dark blue borders as point insight hints (D4) while the data point clicked by the user will be highlighted in pink.

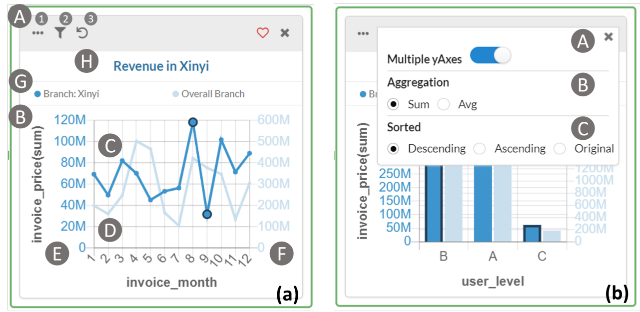

Figure 4: (a) Visual design of each chart. (b) Option panel of each chart.

The title at the top of the visualization shows the main information of the chart. The legend below the title (Figure 4(a), G) includes information about data-subset filters, in the format of an attribute followed by a value (e.g., Branch: Xinyi). Users can zoom in/out of the visualization and click the "Refresh" button to reset its scale.

Moreover, to further obtain users' preference data while not increasing the burden of user ratings, the "Heart" button is used to freely express their special interest in one or more charts. Lastly, they can click the "Close" button to delete a chart from their tree. If a non-leaf chart is deleted, its descendant charts will also be removed.

From the option panel (Figure 4(b)), users can choose whether they want to apply the second Y-axis (Figure 4(b), A), which will appear only when the magnitude difference between the filtered data and the overall data are more than two-fold (Figure 4(a), E, F). This panel can also be used to control the aggregation type (i.e., SUM, AVG) to observe different aspects of the data (Figure 4(b), B). Additionally, sorting order (i.e., descending, ascending, original) can be adjusted to identify the main contributors to such measures (Figure 4(b), C), which enhances the flexibility necessary to compare different charts.

Sheet Management Bar (Figure 1, (3)) allows the user to create multiple results for a dataset by clicking the "Add" button and switching between the sheets to compare different results.

4.3 User Workflow

We use a real user exploration session from Study 1 to demonstrate user workflow (Figure 5). This particular session explores the **transaction dataset (TR)** that consists of customer transactions at a chained department store in Taiwan in 2017. The TR dataset has 12 attributes: seven nominal, three temporal, and two quantitative. The visualization tree in Figure 1 shows the customer purchasing patterns in three different branches of this department store.

When working with a tabular dataset using VisGuide, a user initiates the exploration by selecting the X-axis and Y-axis attributes of the starting chart. In this case, the user specifies the starting chart, which lists the chain's top branches by revenue (Figure 1, A). To continue the exploration, the user can choose a data subset by specifying further filtration conditions. This is done by clicking on a specific data point on a chart. This user action of clicking on a data point will also simultaneously trigger the next chart recommendation (D1). In our running example, the user clicked

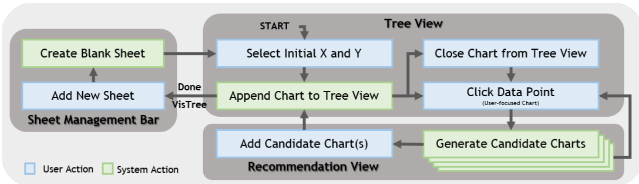

Figure 5: User Workflow: Start by selecting an initial chart and appending it into *Tree View*. Choose and click on a data point of interest from the current "user-focused" chart. The candidates to explore next will be generated in *Recommendation View*. Add the candidate charts to *Tree View* and continue the exploration.

on the bar corresponding to *X="Center Branch"* in Chart A, which corresponds to applying the filter (*X="Center Branch"*). Then a list of recommended charts is also provided (Figure 1, (2)). Please note that, even if different users click on the same data point, they will receive a different order of recommendation charts due to the different personalized recommendation models.

Continuing the exploration, the user can add a chart from the *Recommendation View* to *Tree View* by clicking the "Add" button on the top-right corner of the charts. If a *Comparison* chart is added, it will be placed below the current user-focused chart. For example, Chart B in Figure 1 shows the number of transactions in percentage, which is a *Comparison* chart of Chart A.

Based on Charts A and B in Figure 1, when considering both revenue and transactions, the user notices that the top three branches are *Center*, *Xinyi*, and *Nanjing*. To further explore these branches, the user adds *drill-down* charts (Charts C, D, and E) to *Tree View*. The newly added charts C, D, and E are placed to the right of the user-focused chart (Chart A). These charts reveal that the highest spending customer group at these three branches was different. In particular, *level-A* customers dominated the aggregate value in the *Center* branch, *level-B* customers spent the most money in the *Xinyi* branch, and the majority of customers in the *Nanjing* branch were in *level-C*, with *level-A* being the highest spenders.

The user can continue to select charts based on their interests to receive the next step recommendations and add charts from the recommended list to interactively form a data event (presented as a visualization tree). Charts F, G, and H further drill-down to show the types of merchandise purchased by top customer group at each of the three focal branches. Chart F shows that *level-A* customers at the *Center* branch mainly purchased luxury items; Chart G shows that *level-B* customers at *Xinyi* mainly purchased commodities with high unit prices, but spent less money than *level-A* customers at the *Center*; and Chart H shows that *level-C* customers at *Nanjing* mainly purchased moderately priced goods. The message in this visualization tree (a data event) could help the executives of the department stores to draw up marketing strategies that are tailored to the strengths and weaknesses of different branches and to the needs of their respective customer bases.[2]

[2]A demonstration of the interface is provided in the supplemental video. More visualization results created by the participants in our user study are also provided in the supplementary material (Appendix B).

5 METHODOLOGY OF VISGUIDE

We design VisGuide's user-oriented recommendation engine in terms of three aspects: (1) collecting user preferences; (2) presenting exploration operations; (3) extracting chart features for online training procedures.

5.1 Collecting User Preferences via Implicit Labeling

Training a user preference model requires labeled data that indicate what the user prefers. Collecting and curating such a labeled dataset typically requires human effort and can be quite labor-intensive. Moreover, building a personalized model needs labeled data from every user. To reduce the burden on the users to provide these labeled data, we design an approach to collect user preferences using a mechanism of implicit labeling.

In our design, users' chart preferences are acquired from their interactions with VisGuide. We design the preference score of a chart as a four-level numeric value, $\{0.0, 0.3, 0.6, 1.0\}$. We assume that different user interactions on a chart convey different levels of preference:

- Any chart that is neither selected nor labeled receives a preference score of 0.0.
- Each chart in the recommendation view has a "Star" button, which allows users to mark a chart as worthy of later inspection. If this button is pressed for a certain chart, that chart receives a preference score of 0.3.
- When users add a candidate chart to *Tree View*, it implies that they want to further explore this chart, and the chart, therefore, receives a preference score of 0.6.
- If the "Heart" button on a chart in the *Tree View* is clicked, we consider this a reinforcement of the user's preference on this chart, so its preference score will be increased to 1.0.

Using this design, we can acquire four levels of preference on a chart (None, Star, Add, Heart) and collect sufficient user preference data for building the preference model.

5.2 Recommendation of Candidate Charts

When recommending candidate charts for a user to explore next, VisGuide considers (a) the charts that have been selected so far, (b) the particular data point that the user has expressed interest in the current user-focused chart, and (c) the exploration directions that a human user or data analyst typically takes. In this work, the particular exploration directions that we model in the design of VisGuide are *Drill-down* and *Comparison*.

In the previous study on visualization-sequencing strategies [13], strategies are classified into two types—the *hierarchical structuring* type, which includes strategies that group subsets of charts that have attributes in common, and the *parallel structuring* type, which repeats a pattern of transitions several times in a sequence. The drill-down and comparison operations are examples of *hierarchical structuring*.

In interactive data analysis (IDA), drill-down is a common procedure whereby the behavior of data subsets is analyzed via the progressive addition of filters [7, 18, 26, 31]. Our **Drill-down** operation "zooms in" to a data subset of interest by applying a user-clicked

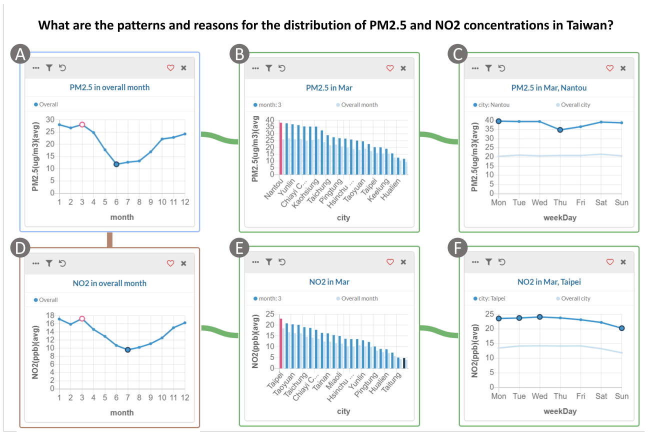

Figure 6: Case from P7: Result about the source and impact of major air contaminants in Taiwan. This event compares the concentration levels of $PM2.5$ (1st row) and NO_2 (2nd row), shows the concentration varies monthly (Charts A and B), and drills down to understand which cities are most affected by the two contaminants (Charts B and E) and presents more details by analyzing the variations on different days of week (Charts C and F).

data point as a filter condition. Its output comprises candidate charts that share the same measure attributes but differ from the current chart in their dimension attributes. For example, Chart B in Figure 6 is a drill-down chart of Chart A.

A **Comparison** operation generates candidate charts that share the same dimension attributes and filter values, but differ from the current chart in terms of measure attributes. For example, Chart D in Figure 6 is a comparison chart of Chart A.

Once VisGuide provides these two types of recommendations, users can interactively create visualization trees with *parallel structuring* by constructing sub-trees with similar patterns. For example, in Figure 6, this tree structure can be divided into two sub-trees—Chart A-B-C and Chart D-E-F—which present the impact of $PM2.5$ and NO_2, respectively.

5.3 Online Training of Recommendation Model

When a user clicks on any data point of interest in a chart and triggers the next chart recommendations, VisGuide includes the preference labels/scores (Section 5.1), which are acquired during the latest round of chart selection/exploration as training data and updates the user preference model. VisGuide's user preference model is a linear regression model of chart features. It is trained online during a user's interaction processes and transferred when the user explores a new dataset. Below, we provide details of the five types of chart features, followed by an introduction to the regression model itself.

5.3.1 Chart Features. Of the five features we use to describe any chart, the first two are statistical measures of its informativeness, which capture the "statistical interestingness" of the chart. Such interestingness does not depend on the specific domain of the dataset; it measures how much users like to see a chart with this statistical property. This supports the efficient exploration of data insights (D4). The other three features relate to a chart's contextualization within a multi-chart sequence, which depends on users' priorities. It naturally obtains their exploring habits through the interaction.

(F1) Point Insight Significance estimates the magnitude of the insights users are likely to derive from a particular point in a chart. It also represents how different an extreme or anomalous point is from other points [4, 35, 41, 44]. We adopted Tang et al.'s approach [35] to measure the significance of a point to potential user insights, computing p-value with the given null hypothesis being true. In [35], a power-law distribution is used as their dataset comes from the business domain. Let $X = \{x_1, x_2, ...x_n\}$ be the set of data points in a chart and x_{max} represent the maximal value in X. Unlike [35], we set the null hypothesis as H_0: X follows a normal distribution $N(\mu, \sigma^2)$ so that our system can be general for datasets in different domains. We determine how surprising x_{max} is by computing the p-value as $p = Pr(x > x_{max}|N(\mu, \sigma^2))$. Then, we use $1 - p$ to represent the significance of the point within this chart.

(F2) Deviation is commonly used to measure the interestingness of the charts recommended during visualization [18, 23, 41]. It computes the difference between the probability distribution of a filtered data subset and the overall dataset. VisGuide uses the Jensen–Shannon divergence (JSD) to perform this computation. Within a group of candidate charts, the ones with larger JSD values are deemed more interesting.

(F3) Granularity. A visualization tree is more understandable if the charts in it proceed either from general to specific data presentations or vice versa [12, 13, 15]. The granularity feature quantifies how closely the transitions from one chart to the next in a series adhere to one of these logical progressions.

(F4) Consistency of Generation Operations. A visualization tree is considered more consistent if the exploration operations connecting the charts in the tree are more of the same kind [12, 13, 15]. For example, a sequence of seven charts in which five are comparisons and two are drill-downs will be deemed less consistent than one with six comparisons and one drill-down. This feature, therefore, consists of the proportion of transitions in a given sequence that are between the same type of exploration operations.

(F5) Encoding Transitions. The relationship between two consecutive charts' dataset attributes is also important when selecting the next charts to recommend. VisGuide, therefore, records the transition of axis encoding between each pair of consecutive charts to capture a user's preferences regarding such transitions. For example, if a chart's X-axis attribute is "City" and its child chart's X-axis attribute is "Station", the value of the feature X-Encoding-Change will be "City2Station". The feature Y-Encoding-Change serves the same purpose with regard to Y axes. All these categorical values are transformed into one-hot encoding features.

5.3.2 Transferable User Preferences Model. A user-oriented recommendation has two aspects: (1) capturing user preference regarding the contextual relations within one dataset during each round of user interactions, and (2) bringing along the learned user preferences on one dataset to a new dataset. To this end, we adopt an online machine learning method to learn users' personalized preference models. Parameters can be reused to improve the learning performance.

A common approach in machine learning applications is transfer learning, whereby knowledge is transferred across different learning tasks by exploiting commonalities between them [2, 25]. We thus expect that any trained user-specific preference model could be reused to reduce that user's labeling effort and to speed up its learning process.

In this work, we use the stochastic gradient descent (SGD) method [41] to train a linear regression model that can capture the importance of each chart feature to different users by learning a user-oriented set of feature weights (D3). This model is then used as the utility function to predict a particular user's preference score for each candidate chart. We define the utility function of visualization as:

$$U(V) = w_0 + \sum_{i=1}^{n} w_i F_i^V, \tag{1}$$

where V denotes a visualization chart; w_i is the weight of the ith chart feature F_i^V of V; and w_0 is the intercept. The SGD algorithm updates the weights to minimize the loss function, which is the least square error with L2 regularization.

A user's preference regarding chart features may change during the exploration process. To adapt to the user's changing preferences, VisGuide puts more emphasis on the latest five rounds of the labeled charts by giving a decaying ratio of 0.9 on the preference scores of the charts labeled by the user five rounds ago. Specifically, we define the labeled data in each round as $L = \{l_1, l_2 ..., l_i, ..., l_k\}$, where l_i represents all labeled scores for charts in round i and k is the number of recommendation rounds so far. The label scores are decayed as follows:

$$D(l_i) = \begin{cases} l_i, & \text{if } i > k-5, \\ (0.9)^{k-i-4} l_i, & \text{if } i \leq k-5, \end{cases} \tag{2}$$

where $D(l_i)$ are the decayed label scores of l_i.

It would be inefficient for the system to re-learn a user-preference model from scratch every time the same user explores a new dataset. Accordingly, to both accelerate the learning process and to reduce a user's labeling efforts, VisGuide adopts a transfer design for its user-preference model. Our transfer mechanism reuses the weights of all dataset-independent features (i.e., F1, F2, F3, F4) learned from the user's previous explorations as the initial weights of the user's preference model on new datasets, while F5 is not reusable as it is sensitive to data attributes. Thus, VisGuide does not need to re-learn users' preferences and users will get a warm start to their next exploration.

6 EVALUATION

The key innovations of VisGuide can be divided into two categories: 1) its effectiveness and usability and 2) its adaptive recommendation model that facilitates the discovery of data charts matching a user's interests. To verify them, we evaluated VisGuide with two studies. Study 1 measured the overall effectiveness and usability of the VisGuide system through an analysis of the participant questionnaires and interview responses regarding their experiences. In Study 2, we compared VisGuide's recommendation performance against that of a baseline method to assess if learning users' preferences adaptively could provide better recommendation results than recommending charts without such an online machine learning mechanism.

Datasets. Two datasets were used in the studies. The first one was the **transaction dataset (TR)**, as mentioned in Section 4.3. The second one was an **air-quality dataset (AQ)**, containing historical air-quality data collected by the Environmental Protection Administration of Taiwan from 2014 to 2019. It has 10 attributes: two nominal, four temporal, and four quantitative. In Appendix B, VisGuide was also applied to analyze the **COVID-19 dataset**[3], which comprises records of daily confirmed cases, deaths, vaccinations, etc., in multiple countries from January 2020 to August 2021 with 13 attributes: two nominal, five temporal, and six quantitative.

6.1 Study 1: Effectiveness and Usability

We designed a qualitative experiment and recruited participants to assess their user experiences, as well as the visual encoding and quality of the generated visualization trees, compared to their past experiences with other visualization tools.

6.1.1 Participants. We recruited 10 undergraduate and post-graduate students (six females, average age 22.8) through recruitment messages posted on social media platforms. We financially compensated the participants with a monetary reward for their efforts. All of them had prior experience in data analysis and in using visualization tools such as Tableau, Power BI, Python, R, or Excel.

6.1.2 Procedures. First, the participants spent 20 minutes watching a tutorial video and becoming familiar with the system by using a mock dataset (YouTube from *kaggle.com*). This was followed by two exploratory sessions with the datasets AQ and TR. The presentation order of the two datasets to a participant was counterbalanced to avoid ordering bias. During each session, a participant had 20 minutes to freely explore a dataset and compile one or more visualization trees. The free exploration avoided introducing bias. The think-aloud protocol was adopted to keep track of the participants' intentions.

At the end of the exploration, each participant spent about 25 minutes filling in a questionnaire regarding the overall usability of the entire system, followed by an interview to further gauge the users' experiences of the visualization trees and VisGuide. We also recorded the log of their exploration on VisGuide to further analyze the effectiveness of the recommendations. On average, the entire study lasted 90 minutes per participant.

6.1.3 Measures. The **questionnaire** comprised ten 5-point Likert-scaled items designed based on previous studies on evaluating visualization systems [24, 29, 30, 37] to assess VisGuide in terms of (1) the effectiveness of its visual design, which measures whether the proposed design helps each user create visualization trees efficiently in terms of tree layout, insight hints, being easy-to-understand, and flexibility; (2) the content of generated results, which was designed to capture users' self-evaluation of their findings, including comprehensiveness, insightfulness, and quality; and (3) the system usability, which measures users' subjective overall experiences of using VisGuide regarding guidance, usability, and usefulness.

Additionally, to verify the **effectiveness of the recommendation system**, we calculated the percentages of the top three

[3]Downloaded from *https://ourworldindata.org/covid-vaccinations.*

recommended charts that were actually selected by the participants on average. During **interviews**, we asked the participants about their perception of recommended charts, the interpretation of tree structure, and their overall experience. All questions asked in the questionnaire and interviews are listed under supplemental materials (Appendix C).

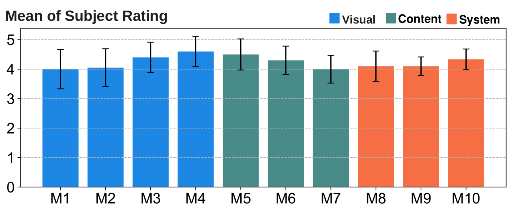

Figure 7: Mean of participant's 5-point Likert scale rating on VisGuide with 95% confidence interval. M1: Tree layout (M:4.0; SD:0.6); M2: Insight hints (M:4.05; SD:0.64); M3: Easy to understand (M:4.4; SD:0.51); M4: Flexibility (M:4.6; SD:0.51); M5: Comprehensiveness (M:4.5; SD:0.52); M6: Insightful (M:4.3; SD:0.48); M7: Quality (M:4.0; SD:0.47); M8: Guidance (M:4.1; SD:0.51); M9: Usability (M:4.1; SD:0.31); M10: Usefulness (M:4.3; SD:0.35).

6.1.4 Questionnaire Results and Recommendation Effectiveness. Figure 7 summarizes the questionnaire results. Among the 10 measures, the participants assigned the highest ratings to visual flexibility (M4) and content comprehensiveness (M5). For recommendation effectiveness, 78.8% (290 of 368) of the charts in the visualization trees were selected from the top three recommended charts. These results support the effectiveness and utility of interactive interfaces when constructing meaningful visualization trees from raw data in a flexible way.

6.1.5 Interview Results. **VisGuide helps the systematic construction of visualization trees.** All participants reported that the Drill-down and Comparison charts helped them identify the causes of events and check if other measures or data subsets also reflected the same phenomenon (D1, D2). As P10 (F, 21) noted, *"I usually inspected drill-down charts first to identify the cause of some extreme value in the specific subset, then I added the comparison charts to verify whether the phenomenon in a subset was a special event or just a global trend by comparing multiple sub-trees."*

Expressive visual design and the capturing of data insights guide users' exploratory processes. All participants commented that the visualizations presented in VisGuide were easy to understand. Nine of the 10 participants also reported that the system's hints helped them grasp data trends and helped them choose their next directions for exploration (D4). *"I am interested in looking into the extreme points of data, and I was able to select my next charts based on those highlighted hints."* (P10). *"The overall information in a chart helps me distinguish whether the trend of a data subset is unusual or normal compared to the overall distribution."* (P9, F, 21). Eight of the 10 participants reported that VisGuide was able to guide their explorations when they encountered bottlenecks. The tree structure also reminded them of their motivation to explore, which allowed them to branch out a new exploration from any sub-tree (D2).

The usability of VisGuide. All participants indicated that they had benefited from VisGuide's efficiency during the exploration process: *"It is convenient that I can create a complex event tree by adding and deleting charts with just a single click."* (P1, M, 21). They pointed out the benefits of being able to directly show their exploration results to their collaborators, especially because the tree layout included both effective charts organization and presentation processes.

Participants' other suggestions for improving our system. First, the system can support more expanded types of recommendations, which can increase the flexibility of exploration in VisGuide. Moreover, they also wanted to manually adjust some details of visualizations in the system, such as rearranging the layout or adding some text next to charts to remind them of the findings and help them make more informative infographics.

6.2 Study 2: Validating User-Oriented Recommendation

In this study, we assess if the recommendations can adapt to different users' preferences and if the model transfer method we proposed can provide better initial recommendations for a new dataset. Thus, each user needs to explore two datasets in different domains. To study the benefits of the proposed user-oriented recommendation component, we designed a between subject study to compare VisGuide against a baseline method. The baseline method consists of a simple version of VisGuide, which has the same user interface but recommends charts only based on the chart features using a utility function with fixed weights regardless of individual user preference. We do not consider comparing other recommendation systems (e.g., Voyager2 [39] and VisPilot [18]) because of the different representations (dashboard views [39]) and purposes (avoiding fallacies [18]). In addition, both of them do not consider user preferences.

6.2.1 Participants. We recruited 20 participants (12 females, 8 males, average age 23.4) through recruitment messages posted on social media platforms. We financially compensated the participants with a monetary reward for their efforts. All have prior data-analysis experience and have used visualization tools such as Tableau, Power BI, Python, R, or Excel, but none are familiar with either of the experimental datasets AQ and TR. These participants were randomly split into two groups; one group used VisGuide in their explorations and the other used the baseline system.

6.2.2 Procedures. Participants were first asked to perform the same practice task that was used in Study 1 (20 minutes) as a warm-up exercise. Then, in the main study, each participant conducted two 20-minute free-exploration sessions, one for each of the two datasets. For VisGuide, the user-preference model that was trained during the first session was carried over into the second session. The order in which the datasets were presented to a given participant was counterbalanced to avoid ordering bias. After each session of exploration, we interviewed the participants. The entire study took around 90 minutes per person.

6.2.3 Measures. We adopted *Normalized Discounted Cumulative Gain* (NDCG) [8, 21]—which is commonly used to assess the ranking quality—to evaluate how satisfied the participants were with the recommendations. In this study, we consider that if the participants assigned higher score labels to those charts that were ranked higher, it means that they were satisfied with the recommendations. The value of NDCG ranges from 0 to 1, with higher scores representing better recommendation results.

Every time a user requested a recommendation for further charts, VisGuide updated its user-preference model based on the labeled charts received in the "user preference feedback" process (Section 5.1). We defined that a *"recommendation round"* is the course of action of performing this recommendation procedure once. In each recommendation round, an NDCG was computed based on the user's preference feedback regarding the recommended charts. We used this index to represent the participant's satisfaction with the recommendation results. We then analyzed such satisfaction by plotting individuals' NDCG results for each round (Figure 8).

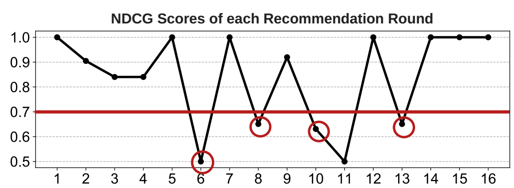

Figure 8: A sample participant's NDCG scores at each recommendation round, using the AQ dataset and in the VisGuide condition. The red line highlight the NDCG threshold, 0.7. The four red circle highlight the four times when the user unsatisfied the recommendation result, which the NDCG score fell below 0.7.

To measure the performance of the model's adaptation to a user's preference change, we designed an index called *"mean rounds to recovery"* (MRTR). The index aims to capture how fast the model's recommendations could adapt to a user's changing preferences that cause the user to be unsatisfied with the recommendation result. We define a "recovery period" as the period of recommendation rounds during which the NDCG recovers from "NDCG < threshold" to "NDCG > threshold". We set the threshold value to 0.7, which corresponds to the case in which a user adds at least two of the top three charts on the recommendation list to a visualization tree.

There may be multiple recovery periods during a user's exploration, and the length of each recovery period varies. To summarize the effectiveness of the model's ability to recover from preference changes, we compute the mean length of the recovery periods that occurred during the exploration. For example, in Figure 8, the user was unsatisfied with the recommendations at the 6th, 8th, 10th, and 13th rounds, and the system adjusts the recommendations to satisfy the user at the 7th, 9th, 12th, and 14th rounds, respectively. The average length of the recovery period, MRTR, for the user was (1+1+2+1)/4=1.25. If MRTR is small, it means that the system can efficiently adjust its recommendation to accommodate a user's preference changes. To summarize the system's performance over multiple users, we compute the average MRTR across all of the participants in the study.

In the interview, we asked the participants to self-evaluate their findings based on how interesting they found data events and whether the findings could help them understand the dataset. Other questions asking during the interviews are listed under supplemental materials.

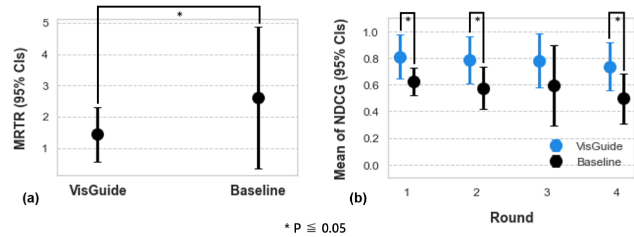

Figure 9: (a) Average MRTR's in the VisGuide and baseline conditions. (b) Mean of the participants' NDCG scores during the first four rounds of the second session (i.e., changing to a new dataset in a different domain.) VisGuide (with its online-learning capacity) had faster recovery and warm-start performance than the baseline model. The asterisks in the charts denote a significance level of 0.05 on the Kruskal-Wallis test.

6.2.4 Recommendation Performance Results. To evaluate the adaptive performance of VisGuide, we performed the Kruskal–Wallis Test on the MRTR of the two conditions as the data are not normally distributed. Figure 9(a) shows the means and 95% confidence intervals of the participants' MRTRs. VisGuide's MRTR was significantly lower than that of the baseline method (p=0.02), indicating that VisGuide was able to adapt to changing user preferences over a lower number of rounds (1.44 on average compared to 2.61 for the baseline system). VisGuide's performance could also be seen as more robust in the sense that the variability in its MRTR was smaller. Together, the above results indicate that VisGuide's recommendations satisfied most participants, whereas the baseline recommendations were not as obvious (D3). Moreover, VisGuide has strength in terms of online, dynamic adaptation when users change their preferences during the exploration process.

Next, to evaluate the warm-start performance of our transfer mechanism, we compared NDCG scores of the first four rounds from the second session between VisGuide and the baseline. As the applied label decaying method will adjust a chart's label score after the fifth recommendation round (see section 5.3.2), only the first four rounds can represent the performance of the transfer mechanism. Therefore, we only considered the NDCG scores in the first four rounds. A Kruskal–Wallis test was used due to the small sample size of each group. Figure 9(b) shows the means and 95% confidence intervals of the participants' NDCG scores for the rounds in question. In the first, second, and fourth rounds, VisGuide received significantly higher NDCG scores than the baseline method did (F=4.5, p=0.03; F=3.83, p=0.05; F=5.2, p=0.02). These results imply that importing a pre-existing user preference model is likely to provide better initial recommendations to users of data-event exploration systems who are analyzing a dataset for the first time.

6.2.5 Exploration Experience and Quality of the Exploration Results. Some participants reported their experiences with the recommendation system in the interview. For example, one stated that *"I figured out that the system recommended the charts that I wanted at the top* *rank, so I can just add the first candidate chart and have a smooth exploration"* (P3, VisGuide). *"The chart I usually wanted to select would be fixed in a certain range of ranking, and would not be ranked higher as my interaction increases"* (P1, baseline). This participants' feedback provides complimentary support regarding the results of the quantitative measurements; VisGuide's users are more satisfied with the recommendation charts than the baseline's users.

During the interview, most of the participants under both conditions had positive responses to their exploration results. The self-rated scores of their findings were 3.9 (VisGuide) and 3.75 (baseline) out of 5, on average. On the other hand, the participants in the VisGuide group could usually build one more visualization tree than those in the baseline group within the limited exploration time (VisGuide: M=3.05, SD=1.23, baseline: M=2.05, SD=1.43). This implies that VisGuide's user-oriented recommendation is more effective in improving the exploration efficiency than the baseline, which does not support user-oriented recommendation.

6.2.6 Proper and Unbiased Guidance. One concern of having a recommendation system is that it may misguide the user, leading to explorations that have systematic biases. We analyzed the participants' generated visualization trees to examine whether VisGuide imposes biases by pushing them to explore either too deeply or broadly. Judging from the results of two conditions, VisGuide did not cause an unbalanced tree structure. Specifically, the statistics of tree height are VisGuide: M=2.3, SD=0.6; baseline: M=2.6, SD=0.9. The statistics for tree breadth are VisGuide: M=3.5, SD=1.5; baseline: M=4.3, SD=2.0. Although the structure of the trees under two conditions was similar on average, the tree structures in VisGuide are more compact and efficient than those in the baseline based on the number of nodes (VisGuide: M=6.3, SD=4.1; baseline: M=8.7, SD=5.4).

The average numbers of nodes, heights, and breadths of the tree structures in the VisGuide condition are 6.3, 2.3, and 3.5, respectively. This could imply that the user spends equal effort exploring the breadth and the depth. Therefore, we think that VisGuide does not impose any strong bias on user exploration. While it is unlikely that VisGuide causes explorations to be either too broad or too narrow/deep, the mediation between the two is entirely up to the user because VisGuide merely proposes exploration hints. The user has the last say in what chart is to be collected in a visualization tree.

7 DISCUSSION

In this section, we discuss the findings and design implications obtained from the user studies, as well as the limitations that should be explored in future work.

7.1 Benefits of VisGuide

Semi-automatic Creation of Data Events. VisGuide attempts to combine the advantages of authoring and auto-generation systems. Through the design of authoring interactions and contextualized chart recommendations, VisGuide makes it possible for users to create insightful and user-oriented data events while avoiding several tedious operations, as well as facilitating the users to integrate chart organization and presentation through an iterative event-generation process and a tree-structure layout. This enables users to remind themselves of their original exploration intention when they encounter bottlenecks. At the same time, they can more easily see the "big picture" that encompasses the charts selected during an exploration.

Representing a data event using a tree structure has the collateral benefit of recording the exploration path taken by the original creator of that event. Such a representation can be further enriched using the user log of the operations also recorded during the generation of the data event. This could also be useful for scholars who seek a deeper understanding of the strategies that humans use in creating data events.

Personalized Chart Recommendations. One limitation of fully auto-generated results is that they may not meet individual users' different needs. VisGuide instead models the user's utility function with a parameterized model and proposes an online learning method to adjust the feature weights of the model according to individual users' preferences. This user-oriented recommendation model appears to offer users a better data-exploration experience than traditional systems, as evidenced by our questionnaire and interview data – all participants were satisfied with the charts recommended to them during event exploration.

To evaluate this new approach of adaptively learning a personalized user model, we applied the NDCG metric to assess recommendation quality as a ranking problem, as well as developed a new MRTR metric to quantify the rate at which the model adjusts to the user's changing preferences. Furthermore, the results in Figure 9(b) reflect that the ability to transfer the model to different datasets is advantageous as it provides a *warm-start* when exploring a new dataset. Therefore, this paper has not only designed an online personalized learning model in a data-event exploration system but has also proposed a method to measure the recommendation quality and adaptability.

7.2 Design Implications on Implicit Labeling of User Preferences

In VisGuide, the "star" and "heart" buttons are designed to represent the functions "bookmark" and "like", respectively (Section 5.1). We originally envisioned that this design could reduce user efforts but would still acquire useful data for building good user preference models. However, the results from user studies indicate that users seldom used those buttons. Instead, they usually just add the charts that interest them to a visualization tree or delete the charts they no longer find interesting.

As for the reasons for lukewarm user reactions, besides being unfamiliar with "bookmark" and "like" and not remembering the purpose of the buttons, some participants pointed out that they couldn't judge the interestingness just based on one chart; a single chart needs to be put in a bigger context to show interestingness. As a result, bookmarking individual charts is not useful nor actionable. Furthermore, since users did not see any effect when they clicked on the "star" and "heart" buttons, they were not sure whether those charts were saved or how they were being used.

The aforementioned observation suggests that the functions of implicit labeling should be more perceivable, e.g., allowing users to browse their bookmarked charts. On the other hand, despite lacking interactions through the "star" and "heart" buttons, a high degree of satisfaction was still achieved by the user preference model. This is surprising but very encouraging proof that a good user preference model could be adequately trained with user preference data on some major kinds of user interactions.

In the user study, we found that some users (three out of ten) added lots of recommended charts into "Tree View" first and then carefully inspected them to decide whether to keep or delete them. For these users, the visualization sequence generation process is more like the tree pruning process rather than the tree growing process. Such findings could imply that many of the recommended charts are good directions for users to explore but each may need to be further examined in the context. In the future, we may experiment with other implicit labeling techniques—for example, measuring the time that users spend on inspecting a chart—to improve the user preference labeling functionality.

7.3 Limitations and Future Work

Limited Chart Types. In VisGuide, the visualization choices of the chart type are determined using heuristic rules based on data attributes (nominal, temporal, and quantitative). VisGuide currently supports three chart types—the bar, line, and pie charts—as they are the ones that "real users strongly prefer" [20]. In the future, we will support other chart types, such as maps and rank charts to extract more information from the data. We will also adopt automatic visual-encoding recommendation algorithms [5, 10, 37] to decide which chart presentation best suits the data and the individual user.

Limited Recommended Types. While VisGuide currently recommends charts of only two exploration types, drill-down and comparison (Section 5.2), other recommendation types could be added to VisGuide in the future. For example, the similarity or contrast relationships suggested in Calliope [32] can be added in VisGuide to render parallel relations between charts, such as the distribution of confirmed cases of COVID-19 in two countries.

Limited Learning Model. Regarding the learning algorithm of the recommendation model, VisGuide currently employs a linear regression model and models the progressive variations of a user's preferences by using a *decaying ratio* (Eq. 2). The long-term and short-term history of user exploration can be further considered using more advanced machine learning models such as the Long Short-Term Memory (LSTM) model or other Recurrent Neural Network models, which can remember and appropriately apply older preferences when making recommendations. We envision that advanced models could offer more diversified recommendation types or sequential patterns of visualization sequences, such as hierarchical structuring and parallel structuring, that could be created by human analysts (Section 5.2).

More Comprehensive Presentation and Data Story Creation. It is not our original design purpose for the users to use a big tree to represent a data event. Instead, our design is to encourage the users to use multiple trees to represent a more complex data event (via sheet management bar). Nevertheless, we acknowledge that our implementation may become cumbersome if users create a large number of nodes, where limited screen size and the amount of interaction operations needed may make it difficult for users to view and manipulate a large visualization tree. Towards a high-quality presentation of data events, we plan to give VisGuide more interaction functions, such as annotation generation and the ability to reorganize charts into an infographic with appropriate amounts of information. Through further integration of text and graphics, VisGuide's users would be able to understand the generated data events more intuitively and comprehensively. Lastly, we designed VisGuide aiming to guide users to explore data, find insights, and form meaningful data events. It will be our future goal to design a system to re-select and reorganize data events in a suitable order to create a data story.

8 CONCLUSION

In this work, we presented VisGuide, a data exploration system that generates personalized recommendations to assist user discovery and presentation of data events. User studies established that the proposed system effectively guides the creation of informative and user-oriented data events.VisGuide achieves the goals via (a) the data insight hints and the user-oriented chart recommendations based on the data's statistical properties and the visualization sequencing strategies, (b) an online machine learning algorithm that adaptively builds a preference model for each user, and (c) a transfer mechanism that reuses the reusable portion of the learned preference models to reduce the users' labeling efforts when a user explores new datasets. The effectiveness of the user-oriented and dynamic recommendation was verified via a user study that specifically measures how fast the recommendation system can notice changes in user preferences and continue to provide satisfactory recommendations, particularly when the user is exploring a new dataset. The result shows that VisGuide can adapt to users' preferences within 1.44 rounds, on average.

Finally, VisGuide adopts a tree-structured layout to intuitively and flexibly present relations between multiple visualizations that collectively form a data event. With the clear representation of the relations between charts (i.e., the breadth and depth exploration, supported by the user-oriented recommendation), the "growth" of a tree structure effectively logs its creator's exploration path while using VisGuide. The user study further shows that, through the recommendation mechanism, VisGuide can inspire users to explore data in unexpected directions and broaden the scope of the exploration.

REFERENCES

[1] Michael Behrisch, Fatih Korkmaz, Lin Shao, and Tobias Schreck. 2014. Feedback-driven interactive exploration of large multidimensional data supported by visual classifier. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE Computer Society, Los Alamitos, CA, USA, 43–52.

[2] Yoshua Bengio. 2012. Deep Learning of Representations for Unsupervised and Transfer Learning. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, Vol. 27. PMLR, Bellevue, Washington, USA, 17–36.

[3] Siming Chen, Jie Li, Gennady Andrienko, Natalia Andrienko, Yun Wang, Phong H. Nguyen, and Cagatay Turkay. 2020. Supporting Story Synthesis: Bridging the Gap between Visual Analytics and Storytelling. *IEEE Transactions on Visualization and Computer Graphics* 26, 7 (2020), 2499–2516.

[4] Zhe Cui, Sriram Karthik Badam, M Adil Yalçin, and Niklas Elmqvist. 2019. Datasite: Proactive visual data exploration with computation of insight-based recommendations. *Information Visualization* 18, 2 (2019), 251–267.

[5] V. Dibia and Ç. Demiralp. 2019. Data2Vis: Automatic Generation of Data Visualizations Using Sequence-to-Sequence Recurrent Neural Networks. *IEEE Computer Graphics and Applications* 39, 5 (2019), 33–46.

[6] Kyriaki Dimitriadou, Olga Papaemmanouil, and Yanlei Diao. 2016. AIDE: an active learning-based approach for interactive data exploration. *IEEE Transactions on Knowledge and Data Engineering* 28, 11 (2016), 2842–2856.

[7] Cody Dunne, Nathalie Henry Riche, Bongshin Lee, Ronald Metoyer, and George Robertson. 2012. GraphTrail: Analyzing Large Multivariate, Heterogeneous Networks While Supporting Exploration History. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1663–1672.

[8] Y. Feng, J. Xu, Y. Lan, J. Guo, W. Zeng, and X. Cheng. 2018. From Greedy Selection to Exploratory Decision-Making: Diverse Ranking with Policy-Value Networks. In *ACM SIGIR Conference on Research & Development in Information Retrieval*. Association for Computing Machinery, New York, NY, USA, 125–134.

[9] David Gotz and Zhen Wen. 2009. Behavior-Driven Visualization Recommendation. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*. Association for Computing Machinery, New York, NY, USA, 315–324.

[10] Kevin Hu, Michiel A Bakker, Stephen Li, Tim Kraska, and César Hidalgo. 2019. VizML: A Machine Learning Approach to Visualization Recommendation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 128.

[11] Enhui Huang, Liping Peng, Luciano Di Palma, Ahmed Abdelkafi, Anna Liu, and Yanlei Diao. 2018. Optimization for active learning-based interactive database exploration. *Proceedings of the VLDB Endowment* 12, 1 (2018), 71–84.

[12] Jessica Hullman, Steven Drucker, Nathalie Henry Riche, Bongshin Lee, Danyel Fisher, and Eytan Adar. 2013. A deeper understanding of sequence in narrative visualization. *IEEE Transactions on visualization and computer graphics* 19, 12 (2013), 2406–2415.

[13] Jessica Hullman, Robert Kosara, and Heidi Lam. 2017. Finding a clear path: Structuring strategies for visualization sequences. *Computer Graphics Forum* 36 (2017), 365–375.

[14] Manas Joglekar, Hector Garcia-Molina, and Aditya Parameswaran. 2019. Interactive Data Exploration with Smart Drill-Down. *IEEE Transactions on Knowledge and Data Engineering* 31, 01 (2019), 46–60.

[15] Y. Kim, K. Wongsuphasawat, J. Hullman, and J. Heer. 2017. GraphScape: A Model for Automated Reasoning about Visualization Similarity and Sequencing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 2628–2638.

[16] Robert Kosara and Jock Mackinlay. 2013. Storytelling: The next step for visualization. *Computer* 46, 5 (2013), 44–50.

[17] Bongshin Lee, Nathalie Henry Riche, Petra Isenberg, and Sheelagh Carpendale. 2015. More than telling a story: Transforming data into visually shared stories. *IEEE computer graphics and applications* 35, 5 (2015), 84–90.

[18] Doris Jung-Lin Lee, Himel Dev, Huizi Hu, Hazem Elmeleegy, and Aditya Parameswaran. 2019. Avoiding Drill-down Fallacies with VisPilot: Assisted Exploration of Data Subsets. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*. Association for Computing Machinery, New York, NY, USA, 186–196.

[19] Qingwei Lin, Weichen Ke, and Jian-Guang Lou. 2018. BigIN4: Instant, Interactive Insight Identification for Multi-Dimensional Big Data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Association for Computing Machinery, New York, NY, USA, 547–555.

[20] Yuyu Luo, Xuedi Qin, Nan Tang, and Guoliang Li. 2018. Deepeye: Towards automatic data visualization. In *2018 IEEE 34th international conference on data engineering (ICDE)*. IEEE Computer Society, Los Alamitos, CA, USA, 101–112.

[21] Yuyu Luo, Xuedi Qin, Nan Tang, Guoliang Li, and Xinran Wang. 2018. DeepEye: Creating Good Data Visualizations by Keyword Search. In *Proceedings of the 2018 International Conference on Management of Data*. Association for Computing Machinery, New York, NY, USA, 1733–1736.

[22] Jock Mackinlay, Pat Hanrahan, and Chris Stolte. 2007. Show Me: Automatic Presentation for Visual Analysis. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1137–1144.

[23] Rischan Mafrur, Mohamed A. Sharaf, and Hina A. Khan. 2018. DiVE: Diversifying View Recommendation for Visual Data Exploration. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. Association for Computing Machinery, New York, NY, USA, 1123–1132.

[24] Honghui Mei, Wei Chen, Yuxin Ma, Huihua Guan, and Wanqi Hu. 2018. VisComposer: A Visual Programmable Composition Environment for Information Visualization. *Visual Informatics* 2, 1 (2018), 71–81.

[25] Grégoire Mesnil, Yann Dauphin, Xavier Glorot, Salah Rifai, Yoshua Bengio, Ian Goodfellow, Erick Lavoie, Xavier Muller, Guillaume Desjardins, David Warde-Farley, Pascal Vincent, Aaron Courville, and James Bergstra. 2012. Unsupervised and Transfer Learning Challenge: a Deep Learning Approach. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, Vol. 27. PMLR, Bellevue, Washington, USA, 97–110.

[26] Tova Milo and Amit Somech. 2018. Next-Step Suggestions for Modern Interactive Data Analysis Platforms. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Association for Computing Machinery, New York, NY, USA, 576–585.

[27] Olga Papaemmanouil, Yanlei Diao, Kyriaki Dimitriadou, and Liping Peng. 2016. Interactive Data Exploration via Machine Learning Models. *IEEE Data Eng. Bull.* 39, 4 (2016), 38–49.

[28] Xuedi Qin, Yuyu Luo, N. Tang, and Guoliang Li. 2019. Making data visualization more efficient and effective: a survey. *The VLDB Journal* 29 (2019), 93–117.

[29] Donghao Ren, Matthew Brehmer, Bongshin Lee, Tobias Höllerer, and Eun Kyoung Choe. 2017. ChartAccent: Annotation for data-driven storytelling. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE Computer Society, Los Alamitos, CA, USA, 230–239.

[30] Donghao Ren, Bongshin Lee, Matthew Brehmer, and Nathalie Henry Riche. 2018. Reflecting on the evaluation of visualization authoring systems: Position paper. In *2018 IEEE Evaluation and Beyond-Methodological Approaches for Visualization (BELIV)*. IEEE Computer Society, Los Alamitos, CA, USA, 86–92.

[31] E. Segel and J. Heer. 2010. Narrative Visualization: Telling Stories with Data. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (2010), 1139–1148.

[32] D. Shi, X. Xu, F. Sun, Y. Shi, and N. Cao. 2021. Calliope: Automatic Visual Data Story Generation from a Spreadsheet. *IEEE Transactions on Visualization & Computer Graphics* 27, 02 (2021), 453–463.

[33] Arjun Srinivasan, Steven M Drucker, Alex Endert, and John Stasko. 2018. Augmenting visualizations with interactive data facts to facilitate interpretation and communication. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2018), 672–681.

[34] Charles D. Stolper, Bongshin Lee, Nathalie Henry Riche, and John Stasko. 2016. *Emerging and Recurring Data-Driven Storytelling Techniques: Analysis of a Curated Collection of Recent Stories*. Technical Report MSR-TR-2016-14. Microsoft.

[35] Bo Tang, Shi Han, Man Lung Yiu, Rui Ding, and Dongmei Zhang. 2017. Extracting Top-K Insights from Multi-Dimensional Data. In *Proceedings of the 2017 ACM International Conference on Management of Data*. Association for Computing Machinery, New York, NY, USA, 1509–1524.

[36] Edward R. Tufte. 2001. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT.

[37] Yun Wang, Zhida Sun, Haidong Zhang, Weiwei Cui, Ke Xu, Xiaojuan Ma, and Dongmei Zhang. 2019. DataShot: Automatic Generation of Fact Sheets from Tabular Data. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2019), 895–905.

[38] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. 2016. Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (Jan 2016), 649–658.

[39] Kanit Wongsuphasawat, Zening Qu, Dominik Moritz, Riley Chang, Felix Ouk, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2017. Voyager 2: Augmenting Visual Analysis with Partial View Specifications. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 2648–2659.

[40] X. Zhang, X. Ge, and P. K. Chrysanthis. 2019. Leveraging Data-Analysis Session Logs for Efficient, Personalized, Interactive View Recommendation. In *2019 IEEE 5th International Conference on Collaboration and Internet Computing (CIC)*. IEEE Computer Society, Los Alamitos, CA, USA, 110–119.

[41] Xiaozhong Zhang, Xiaoyu Ge, Panos K. Chrysanthis, and Mohamed A. Sharaf. 2021. ViewSeeker: An Interactive View Recommendation Framework. *Big Data Res.* 25, C (jul 2021), 15 pages.

[42] Jian Zhao, Shenyu Xu, Senthil Chandrasegaran, Christopher James Bryan, Fan Du, Aditi Mishra, Xin Qian, Yiran Li, and Kwan-Liu Ma. 2021. ChartStory: Automated Partitioning, Layout, and Captioning of Charts into Comic-Style Narratives. *IEEE Transactions on Visualization and Computer Graphics* PP (2021), 1–1.

[43] Zhenpeng Zhao, Rachael Marr, and Niklas Elmqvist. 2015. Data comics: Sequential art for data-driven storytelling. *tech. report* PP (2015).

[44] Çagatay Demiralp, Peter J. Haas, Srinivasan Parthasarathy, and Tejaswini Pedapati. 2017. Foresight: Rapid Data Exploration Through Guideposts. *CoRR* abs/1709.10513 (2017).

A IMPLEMENTATION DETAILS

We implemented VisGuide on a Windows 10 operating system, running with Google Chrome and a monitor with a resolution of 1920×1080. Participants interacted with the VisGuide system by an external mouse. We used Python Flask as our back-end server and utilized javascript, css, and HTML to create the interactive front-end website. The visualization libraries, D3 and chart.js, were used in our system to render the visualization in the user interface.

B PARTICIPANTS' SAMPLE EXPLORATION RESULTS

In our user study, participants created diverse and insightful results to present data stories. The following are more of their exploration results.

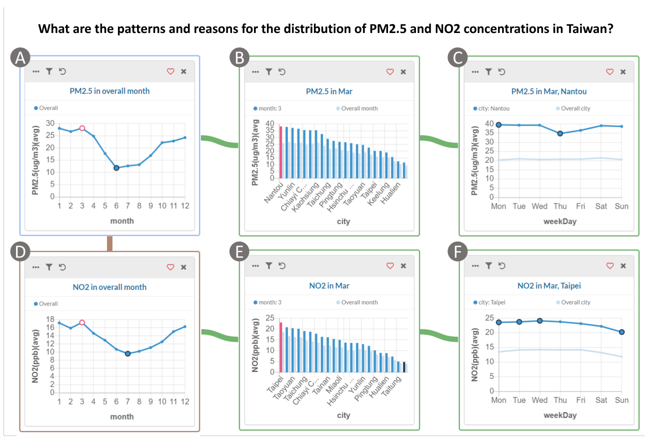

Figure 10: Visualization tree about the source and impact of major air contaminants in Taiwan. This event compares the concentration levels of $PM2.5$ (1st row) and NO_2 (2nd row), shows the concentration varies monthly (charts A and B), and drills down to understand which cities are most affected by the two contaminants (charts B and E) and presents more details by analyzing the variations on different days of week (charts C and F).

B.1 Case 1: Source and impact of major air contaminants in Taiwan (P7, M, 19)

Figure 6 compared two atmospheric contaminants, $PM2.5$ and NO_2, in terms of their concentration levels by month of the year and by city. The event starts by showing the trend of the monthly concentration levels of the two pollutants (charts A and D).

VisGuide has marked the month of March as an insight-worthy data point as March is the month in which both chemicals have the highest concentration levels. Besides, concentration levels of both pollutants are higher in winter than in summer. Directly to the right of charts A and D are two bar charts, B and E, which drill down on the data for the most polluted month, March. Chart B shows that, in the case of $PM2.5$, cities in central and southern Taiwan have higher concentration levels. On the other hand, higher concentrations of NO_2 are to be found in the more populous cities of the north, as shown in Chart E. To further explore the causes of this geographic variation in $PM2.5$ and NO_2, the participant continually drilled down on the most polluted cities of the two contaminants to get a weekly trend (chart C and F). This pair of charts reveals that the concentration of NO_2 drops slightly during weekends, while that of $PM2.5$ does not.

This visualization tree help its creator reason about the trends of these two air pollutants by the usage of the comparison charts and the parallel tree structure. Our participant hypothesized that the concentration of $PM2.5$ might be correlated with the northeast monsoon, seasonal northeast winds from which could have brought suspended particles to southern Taiwan; whereas the concentration of NO_2 might be mainly caused by motor-vehicle exhaust or emissions from factories.

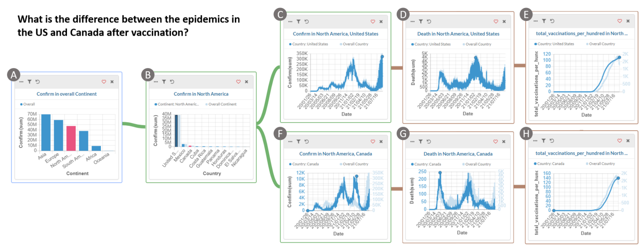

Figure 11: Visualization tree about the different epidemic trends of COVID-19 in US and Canada. Charts C to charts H show the daily new confirmed cases, death cases, and vaccinations per hundred people in the US and Canada, respectively. (Comparison charts (brown links) have been moved to save space.)

B.2 Case 2: Trends of the COVID-19 spread in the US and Canada.

Figure 11 shows the trends of daily new confirmed COVID-19 cases, death cases, and vaccination ratios from 2020 to 2021. The visualization tree provides hints about the relation between confirmed/death cases and vaccinated rate. The event proceeds from an overview of the continents with the COVID-19 cases (chart A), and then drill-down to see the countries in the North America (chart B). The user selected chart C, D, E as to observe the epidemic distribution in the US. Similarly, when the user selected chart F to view Canada, charts G and H were recommended in the first two candidate comparison charts due to the user preference model.

Charts C and F show that, in both countries, the coronavirus broke out in late March 2020, and rapidly increased from October 2020. For the deaths in chart D and G, because of the sudden outbreak of the epidemic, the numbers of deaths were high in the early days. At the peak of the epidemic, the number of deaths in a single day even exceeded 4,000 in the US. In charts E and H, when the vaccines began to be administered generally in January 2021, the daily numbers of infected and dead people were gradually decreasing until June 2021. Chart E also shows that the time for vaccination in the US started earlier than other countries in the North America, while Canada (chart H) proceeded similarly with other countries (i.e., the "Overall Country" information indicated by the light blue line).

Therefore, charts C, D, E and charts F, G, H together show that the epidemic was under better control in the US and Canada when the vaccines began to be administered. However, with the invasion of the mutant viruses (Delta), the numbers of confirmed cases and deaths reached another peak in the US in July 2021. Facing a similar situation (mutation virus Alpha) in April 2021, the number of confirmed cases in Canada also greatly increased, but the death count was well controlled. It's noteworthy that the vaccination rate in Canada then was still low. This is an interesting finding of which the cause needs to be further investigated. This visualization tree helps users identify the interesting data subset by the hints of extreme data points and also help users compare the infected trend through the hints of the overall data trend.

B.3 Case 3: Comparison of different contaminants in air-quality dataset (P4, M, 24)

Figure 12 presents an event about the comparison among different air contaminants in Taiwan. Chart A shows the $PM2.5$ concentration of the cities in central and southern Taiwan have higher concentration levels. Chart B further drill-downs to "Kaohsiung" to see the concentration trend over a year. It can tell that $PM2.5$ is more polluted in winter than summer, which implicates the cause of $PM2.5$ might be related to the seasonal effect. After seeing the "Comparison" chart recommendations of Chart B, the user found that NO_2 had the same concentration trend as the $PM2.5$ (Chart C).

To check if the main polluted area of NO_2 is the same as $PM2.5$, he added Chart D, which is a comparison chart of Chart A and it presents the NO_2 concentration of different cities. After seeing Chart D, he found that the most polluted area of NO_2 is in the central and northern Taiwan, which is different from that of $PM2.5$. Chart E further shows the NO_2 concentration of different stations in the most polluted city, "Taipei". The user found that stations in the main traffic and densely populated area (i.e.,"DaTong", "ZongShan") is more polluted than stations in the sparsely populated area (i.e.,"YangMing" mountain). Besides, from the "Comparison" chart of Chart E, he found that the most polluted area of O_3 is almost opposite of NO_2, where the sparsely populated area is more polluted than the densely populated area (Chart F). To further explore the cause of the O_3, Chart H was recommended from the top of the drill-down candidate charts according to his user preference model, so he added it to see the trend of O_3 over a year. He found that O_3 is more polluted in spring and fall. After creating this results, the user can tell that the cause of $PM2.5$ might be related to seasonal factors, NO_2 might be mainly related to the traffic exhaust and O_3 could be probably due to other factors, such as terrain or climate.

B.4 Case 4: Comparison of PM2.5 and O3 (P6, F, 22)

Figure 13 presents a visualization tree about the comparison between $PM2.5$ and O_3 in Taiwan. Chart A shows the $PM2.5$ concentration is decreasing from 2014 to 2018. From Chart B, the user found $PM2.5$ might be related to the seasonal effect, then she added Chart C and D to inspect the main polluted area in the most polluted month, January, and the less polluted month, August, respectively. She found that the southern cities in Taiwan are more polluted in winter while the northern cities in Taiwan are more polluted in summer. This could imply that $PM2.5$ might be correlated with the northeast monsoon as the suspended particles might be brought to southern Taiwan by the northeast seasonal winds.

After seeing the concentration trend of other contaminants (i.e., NO_2, SO_2, O_3) from 2014 to 2018 in the "Comparison" charts of Chart A, she found that O_3's trend is different from $PM2.5$'s (Chart E) as O_3 did not have a decreasing trend. From Chart F, she found the most polluted seasons of O_3 are spring and fall. To check if O_3 is also related to the northeast monsoon, she clicked on the most and less polluted months, October and July, respectively. Then, VisGuide recommended Chart G and H based on her preference model, so she added them to see the main polluted area of O_3. She found that the polluted areas in Chart G and H are not much different, which can only imply that O_3 may be correlated to seasonal effect but its cause is different from $PM2.5$'s.

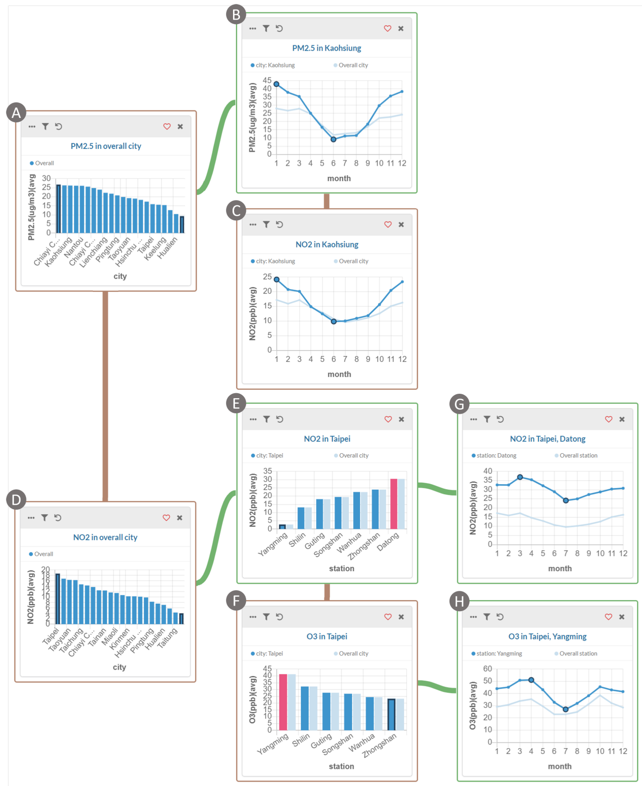

Figure 12: Case 3: Comparison of different contaminants ($PM2.5$, NO_2, O_3) in Taiwan.

B.5 Case 5: Consumer behavior of different genders in the transaction dataset (P1, M, 28)

Figure 14 presents the consumer behavior of different genders. Chart A shows that females spent much more money than males.

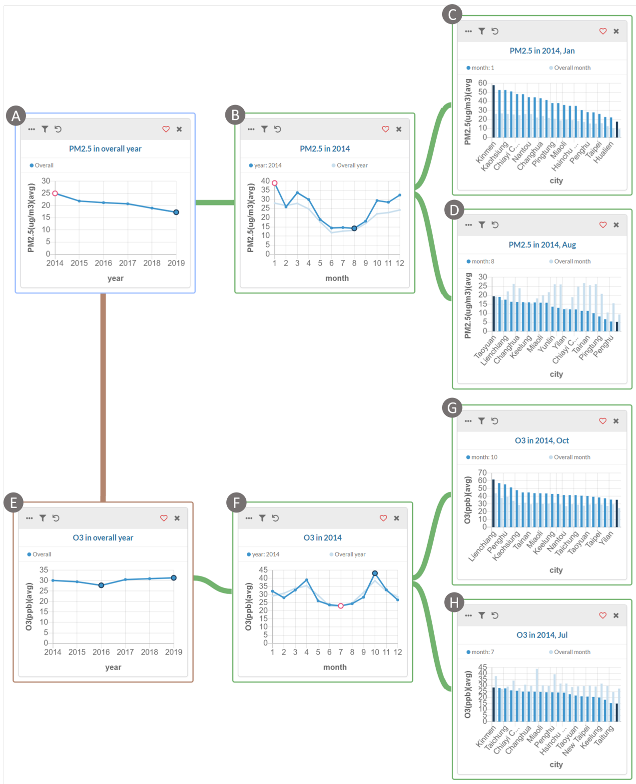

Figure 13: Case 4: Comparison of contaminants, $PM2.5$ and O_3, in the air-quality dataset.

From chart B, we can tell that females also shopped more frequently than males in this department store. Chart C and D present the top-selling merchandise category of females and males, respectively. Although the customers of both genders spent the most money on the luxury goods, we still can tell that females usually purchase famous brands or design brands while males prefer to buy appliances and 3C application.

Figure 14: Case 5: Consumer behavior of different genders in transaction dataset.

B.6 Case 6: Shopping behavior of foreign and native customers in the transaction dataset (P2, F, 23)

Figure 15 presents the event. Chart A reveals that the foreigners spent more money than the natives in average (spending per transaction). Chart B shows that the natives shopped more frequently than the foreigners. These two charts imply the foreigners probably seldom shop in this department store but they spent more money per transaction. Charts C and D indicate that the native females spent much more money than native males while the foreign males spent slightly more money than the foreign females. Furthermore, the participant wondered the consumption trend of foreign and native females, so she added Chart E and F to see their consumption trend over a year. From Chart E, she noticed that the foreign females spent the most money in September, which is different from the overall trend. Besides, Charts G and H show that foreign females usually go to the department store on weekends while the native females often go shopping on weekdays.

Figure 15: Case 6: Consumer behavior of different genders in transaction dataset.

B.7 Case 7: Target customers of specific merchandise category in transaction dataset (P5, F, 20)

Figure 16 presents a data event about the target customers of specific merchandise category. From Chart A, the user found that the sales volume of "Jewelry" is much higher than other categories. To find out the target customers, the user continuously added the drill-down charts. Chart B shows that females spent more money on "Jewelry" than males. From Chart C, the user found that "level-A" customers are the main consumers of "Jewelry". This finding is reasonable as the "level-A" customers can afford luxury more than other levels' customers. Chart D further shows when these level-A, female customers buy "Jewelry", they usually went shopping on weekdays rather then weekends.

B.8 Case 8: Trends of the COVID-19 spread in the US, India, and France.

Figure 17 presents a data event about the trends of the COVID-19 spread in three different countries from 2020 to 2021. Chart A shows

Figure 16: Case 7: Target customers of specific merchandise category in transaction dataset.

the number of confirmed cases in all countries around the world. Chart B further drill-downs to the US to see the daily confirmed cases. Charts C and D compare the death cases and vaccination per hundred people in the US. Chart E shows the country that has the second highest number of daily confirmed cases, India. VisGuide then recommends Charts F and G in order in the comparison candidates when the user clicks the point in Chart E, because VisGuide has learned the user's preference from Chart B to D. Similarly, when the user chooses France (Chart H) as the third country to be compared, Charts I and J are recommended by VisGuide.

It can tell that because of the sudden outbreak of the epidemic, the numbers of deaths were high in the early days, especially in the US and France (Chart C, I). At the peak of the pandemic from October 2020 to January 2021, the whole world was falling into a serious virus raging (light-blue line in Chart B, C), including the US and France. Soon after, in April 2021, the invasion of the mutant virus Alpha also had a serious impact to the world, including India and France. The US did not have a serious epidemic during this period (Chart B, C) because the vaccines were administered earlier than the global average (Chart D). However, when the number of vaccine administrations flattened out, the invasion of another mutant virus Delta led to the second peak of the epidemic in the US during July 2021 (Chart B, C). The other two countries, India and France, were not as severe as before (Chart G, H).

Figure 17: Case 8: Trends of the COVID-19 spread in the US, India, and France. (Comparison charts (brown links) have been moved to save space.)

C QUESTIONNAIRE AND INTERVIEW QUESTIONS

C.1 Questionnaire Questions in Study 1

- **M1**: "Did the tree layout help you organize data stories systematically?"
- **M2**: "Did the insight hints in each chart help you identify data insights efficiently?"
- **M3**: "Was the information presented in each chart easy to understand?"
- **M4**: "Did the interactive chart-selection method and adjustability of aggregation and sorting methods help you create more flexible stories?"
- **M5**: "Did the data stories you found help you understand the dataset comprehensively?"
- **M6**: "Did you find the data stories insight-provoking?"
- **M7**: "Are you satisfied with the data stories you made using the system?"
- **M8**: "Did VisGuide help inspire your next exploration direction?"
- **M9**: "Is the overall system easy to use?"
- **M10**: "Considering all of your past data-analysis experiences, do you think VisGuide is useful for exploring data?"

C.2 Interview Questions in Study 1

(1) "How did you decide about your usage of the Drill-down and Comparison charts?"
(2) "How did you decide to click the "star" and "heart" buttons?"
(3) "How did you interpret the multiple charts in a tree layout?"
(4) "Did you ever become trapped in a recommendation result, and if so, did the system help you escape such a bottleneck?"
(5) "Based on your past experience, how did our system differ from traditional data-exploration tools such as Tableau, Power BI, or Excel?"

C.3 Interview Questions in Study 2

(1) "How did you choose the chart from the candidate recommendation list?"
(2) "How did the system help you improve the efficiency of data exploration?"
(3) "Please give a score of 1 to 5 based on how interesting the event you found and whether it can help you understand the dataset."