STORYBOARD: Navigating Through Data Slices with Hierarchical Summary of Visualizations

Doris Jung-Lin Lee*, Himel Dev*, Huizi Hu, Hazem Elmeleegy, Aditya Parameswaran

Abstract—The task of navigating through a large, multidimensional dataset is a common challenge in exploratory analysis. Not only is manual drill-down and roll-up on data subsets tedious and inefficient for the analyst, the massive space of data subsets, lack of interesting patterns in most data subsets, fallacies of spurious correlations, and pitfalls of statistical paradoxes calls for a systematic and effective way for an analyst to make sense of and navigate through the large space of possible visualizations. In this paper, we present STORYBOARD, an interactive visualization recommendation system that automatically identifies k connected visualizations that summarizes the interesting and informative trends in the dataset to the user. Given a dataset and the x and y axes of interest, STORYBOARD intelligently explores the lattice of equivalent visualizations across data subsets, and recommends interesting and informative visualizations based on an intuitive user-expectation model. The recommended visualizations are then displayed in an interactive dashboard, where the visualizations are organized into a hierarchical layout. Our evaluation study shows that visualization dashboards generated by STORYBOARD are interpretable and leads to higher performance in data analytic tasks compared to the competing baselines.

Index Terms—exploratory data analysis, visualization recommendation.

1 INTRODUCTION

Common analytics tasks, such as causal inference, feature selection, and outlier detection require studying the distributions or patterns at different levels of data granularity [5], [10], [27]. For example, a campaign manager may be interested in the voting patterns across different demographics (based on race, gender, social class etc.) using the 2016 US election exit polls[1] to identify demographic groups for targeted advertisement. Visual analysis is the common approach for performing such analytics tasks, in which an analyst constructs visualizations to capture the distributions at different subsets of data. The goal of this visual analysis is to extract meaningful insights—when a set of visualizations along with human interpretation leads to informative and interesting facts about the underlying distributions.

However, without knowing *what* subset of data contains an insightful distribution, manually exploring distributions from all possible data subsets can be tedious and inefficient. For example, the aforementioned campaign manager could construct bar charts for all possible demographics, where x-axis shows the election candidates and y-axis the percentage of votes for these candidates. Subsequently, he may visually compare these bar charts to understand how voting pattern changes across different demographics. Even after constructing the visualizations for all possible data subsets, which itself is a daunting task, currently there is no systematic way for the campaign manager to make sense of or even navigate through this large space of possible visualizations to draw meaningful insights.

To this end, we present STORYBOARD, an interactive visualization summarization system that automatically selects a small set of visualizations to summarize the distributions within a dataset in an informative manner. Our system is motivated by our observation that when an analyst is aware of the distributions present in different data subsets, she can draw meaningful insights and establish correlations about related visualizations that she has not yet seen with ease. We define this aspect of dataset understanding as *distribution awareness*. For example, based on the vote distributions for different demographics shown in Figure 1, an analyst can infer that most demographic groups have similar voting behaviors for 'Clinton' and 'Trump' (a,b,c,e,f), whereas black demographic groups are strongly skewed towards voting for 'Clinton' (d,g,h). Since human analysts have limited time and memory, it is often impossible to explore visualizations from all data subsets. An ideal summarization system should display visualizations that enables the analyst to achieve *maximal distribution awareness*, from which she can reasonably approximate most of the remaining unseen visualizations in a dataset.

Nevertheless, finding effective visualizations to summarize a dataset is not as trivial as picking individual visualizations that maximizes some statistical measure, such as deviation [24], coverage [21], or significance testing [5], which can often result in misleading summarizations. For example, if the campaign manager uses a deviation based metric to identify insightful distributions [24], he could find that the voting pattern of black females is drastically different from the voting pattern of general female population. Accordingly, he might allocate his advertisement funds to target the black female population. While black females do defy the trends of general females, the comparison is incomplete, since it ignores the fact that black females very closely follows the voting pattern of the black population. Accordingly, the proper demographic to target should be the black population rather than the more specific black female population.

The above example demonstrates a scenario where the selection of an improper reference (female) for comparing the visualization (black female) against results in misleading insights. In STORYBOARD, we formulate an objective where a visualization is *actually* interesting when it deviates from and can not be explained by *even* its most informative reference. Our user study results described in Section 6 shows that this notion of informative interestingness can guide an analyst towards more meaningful stories for further investigation. The contribution of this paper include:

1. https://edition.cnn.com/election/2016/results/exit-polls

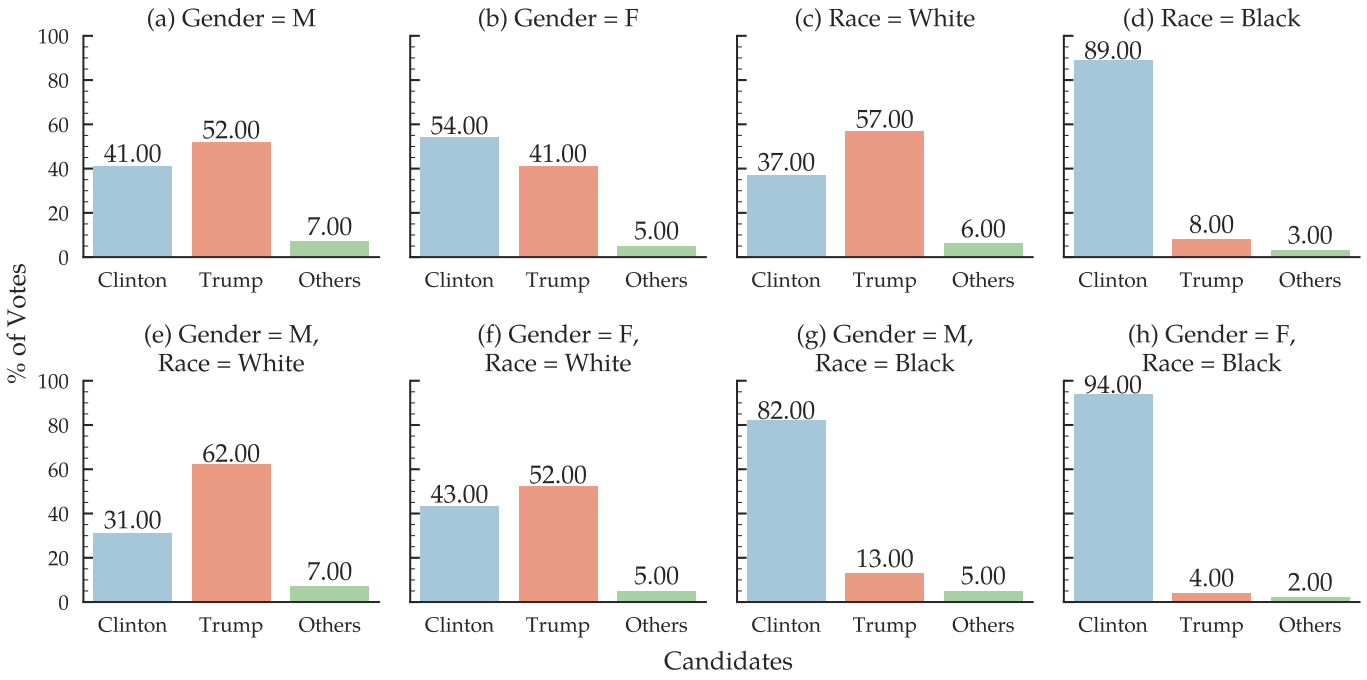

Fig. 1: A set of visualizations from the 2016 Election polls. These visualizations show the percentage of votes for three candidates (Donald Trump, Hilary Clinton, and Others) in different demographic groups (based on race and gender).

- Proposing the novel problem of visualization summarization and use cases highlighting the importance of *distribution awareness* in dataset understanding (Section 2),
- Formulating the structure and utility of the visualization search space (*lattice*) using a user expectation model motivated by our formative study (Section 3),
- Designing efficient algorithms and optimizations to identify a set of informatively connected interesting visualizations (Section 4),
- Presenting an interactive visualization dashboard interface that adopts a simple and intuitive hierarchical lattice layout (Section 5),
- Demonstrate the efficacy of our system through a user study evaluation (Section 6).

2 Distributional Awareness and its Applications

The idea of picking a small set of visualizations to summarize the dataset is motivated by the observation that most of the data distributions in a dataset can be explained by a much smaller set of visualization instances[2]. We define that a set of visualizations instills *distributional awareness* if the visualization helps analyst understand many of the possible visualization distributions and their associated attribute combinations. The goal of visualization summarization is to assemble a dashboard of k visualizations that help analysts become distributionally aware of other unseen perspectives in the dataset. In addition to accelerating the process of manual comparisons across different data subsets described in the introduction, we illustrate a series of common data analytics tasks that would benefit from enhanced distributional awareness through visualization summarization In this section, we use the popular Titanic dataset as a continued example, commonly used for introducing novices to the classification problem in machine learning [2].

Preventing Fallacies in Causal Inference: Drawing inference from observations is important for discovering causal relationships that support or refute a hypothesis, as well as generalizing predictions for unseen data. During exploratory analysis, causal inference based on the incomplete, aggregate information can result in Simpson's paradox [9], whereby an observed trend between variables reverses when conditioned upon an unseen variable.

One example of Simpson's paradox in the Titanic dataset is the survival rate of passengers for third-class passengers versus crew members. Overall, the survival rate of third-class passengers is slightly higher (24.08%) than crews (23.95%). However, when we examine the survival rate of the two classes conditioned on gender as shown in Table 1. We find that for both genders, the survival rate of the crews is higher than third-class passengers.

Gender	Class	Survived	Lost	Survival Rate
M	Third	75	387	16.23%
M	Crew	192	670	22.27%
F	Third	76	89	46.06%
F	Crew	20	3	86.96%

TABLE 1: Survival Rate by Gender and Two Classes

In this case, Simpson's paradox arises because the gender distribution for each passenger class was not shown, so analysts may be misguided into thinking that survival rates of third-class passengers should be equal to or slightly higher than crews. As studied by [3], [9], identification of Simpson's paradox is important as it reveals interesting subgroups which differ from their expected behavior–so much that their aggregated trends are reversed. While the goal of StoryBoard is not the identification of Simpson's

2. This principle is more generally known as the 80-20 rule in economics, e.g. 80% of the wealth (effect) is held by approximately 20% of the population (cause).

paradox, our goal of helping analysts become distributionally aware of their dataset circumvent this issue, since the displayed visualizations should yield a informative view that covers the main patterns in the data. Furthermore, our objective ensures that for a visualization to be selected in the dashboard at least one of the "informative parents" are already in the dashboard to exclude visualizations that could misguide the users into such fallacies.

Feature Selection for Machine Learning: Data scientists often create visualizations to uncover the relationships between their chosen attributes and potential influencer variables to identify attributes that are relevant to the prediction task. Feature selection is a non-trivial problem: analysts seek attribute combinations that are highly discriminative, yet general enough to prevent overfitting and increase model interpretability. While existing classification algorithms such as decision trees highlight some of such cases, as their end-goal is to improve classification score, the reductionist view does not showcase the complex interactions where trends may be changing when additional attributes are added. By becoming distributionally aware of the dataset, analysts can learn key patterns that are associated with particular attributes, such as female passengers on the Titanic has a much higher survival rate than male passengers. As described in the paper, the non-monotonic story paths selected by the context-dependent objective in STORYBOARD Doris: might be too much to mention this here, consider putting this para after models? or as discussion? can help users make more informative judgments about feature importance in the given datasets.

Contextual Outlier Detection and Interpretation: In many data-driven applications, outlier detection plays an important role in identifying groups of instances that are different from the majority. Interpreting *why* a particular outlier was chosen is essential for analysts to reason about the underlying phenomenon resulting in the outlier. Female crew member in the Titanic dataset is one example of such outliers, whose high survival rate is different from that of both female (51.06%) and crews (23.95%), whereas the female third class passengers can be explained using the general observations regarding female. Such outliers may be of interest to the analyst as they indicate the presence of unseen confounds that influence the variables of interest.

3 DATA AND USER MODELS

In this section, we first describe our data model by reporting our data, visualization and query setup, and the underlying lattice of data subsets. We then discuss how analysts explore the lattice through a formative user study, and introduce our user model based on the findings of this study. Finally, we present our problem of finding an informative-and-interesting set of visualizations from the lattice.

3.1 Structure of visualization stories

Data Model: We consider the common visual analytics scenario where a dataset consists of a relational table R with *dimensions* attributes to be filtered upon and *measure* attributes to be aggregated upon. A visualization of this dataset consist of: (i) `X`: x-axis attributes, (ii) `Y`: y-axis attribute, (iii) `C(Z)`: filter constraints that specify the data subset, (iv) `A`: aggregation functions for the x- and y- axes. For example, the aforementioned elections dataset has four attributes: voter's `ID`, voter's `Gender`, voter's `Race`, and the `Candidate` that the voter voted for. As shown in Figure 1, even for the same x- and y- axis attribute, aggregation, and chart type, there can be different visualizations corresponding to different demographic groups (the combination of `Gender` and `Race`). Such visualizations can be written as SQL query: `SELECT X, A(Y) FROM R WHERE C(Z) GROUP BY X`.[3]

Given a set of visualizations V with the same X and Y and different $C(Z)$, we extend the set-theory based *containment* relationships for data subsets and organize the visualizations into a *lattice* as depicted in OLAP data cube literature [28]. A visualization V_i (defined by filters C_a) is a *parent* of the visualization V_i^j (defined by filters C_b) if C_b can be obtained from C_a by adding one additional filter constraint. For example in Figure 1, the visualizations (b) Female and (d) Black are the parents of the (h) Black Female visualization. Based on the containment relationship of visualizations, we can organize the visualizations from V to form a lattice, as exemplified in Figure 2. The lattice contains all visualizations with same x- and y- axes for different data subsets, arranged based on the parent-child relationships between visualizations. The choice of a filter-based data lattice is supported by research in visualization storytelling showing that people prefer visualization sequences structured hierarchically with increasing levels of aggregation [12], [13], [14]. Given this structure describing the space of possible visualizations, we will now discuss how the edge utility of visualizations in this lattice is defined.

3.2 Utility of Visualizations: User Expectation Model

In order to identify which visualizations should be picked from the lattice, we conduct a formative user study to study how the presence of one or more observed parents in the visualization lattice affects an analyst's perception of an unseen visualization. Using these findings, we then model the effective utility of displaying an unseen visualization to a user in the context of seen visualizations.

Formative User Study: We recruited 9 participants in a study to predict the distribution of an unseen visualization with two constraints. Participants were asked to make a prediction regarding an unseen visualization after seeing the first parent displayed and subsequently after seeing the second parent displayed. For the chosen visualization parents, the first parent have data distributions that very closely follows that of the unseen visualization, whereas the second parent differs greatly from the unseen visualization. In this between-group study, one group of participants (G1) were shown the first parent followed by the second parent, whereas another group of participants (G2) were shown the second parent followed by the first parent. We examined how users form their expectation in presence of these observed parents. The results are summarized in Figure 3. Our main findings are: (1) participants naturally form their expectations based on one or more observed parents; (2) seeing a parent that well describes the unseen visualization leads participants to better estimate the unseen visualization; (3) in absence of an informative parent, participants can be misled to form an inaccurate expectation; (4) in presence of both informative and uninformative parents, the variance of expectation increases compared to just seeing the informative parent.

Based on these findings, we model two aspects of the unseen-visualization and observed-parent relationship through its *interestingness* and *informativeness* criteria.

3. Note that our method directly applies to all counting-based aggregation functions on `Y`, such as COUNT, SUM, AVERAGE, PERCENTAGE. However, our method is not directly applicable to other aggregate functions, such as MIN, MAX, MEDIAN.

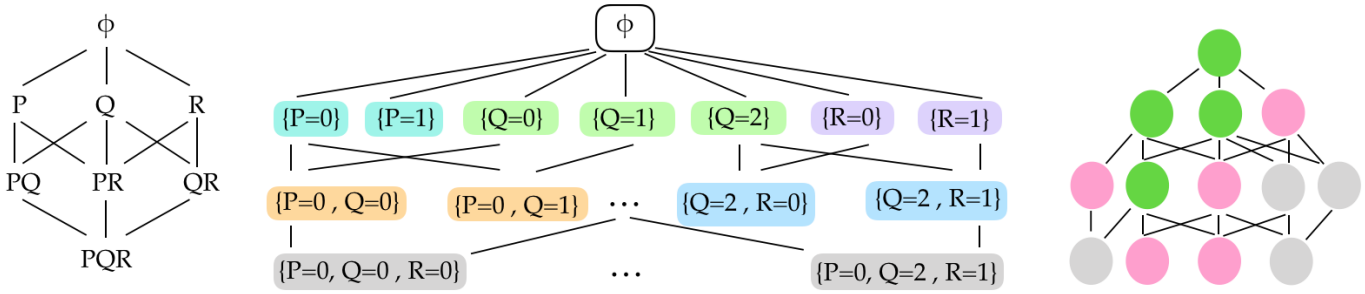

Fig. 2: Left, Middle: Example data subset lattice for a dataset with three binary attributes `{P, Q, R}`. The data subsets that belong to the same attribute combinations are highlighted with same color. ϕ represents the overall distribution (where no filter conditions is applied). Right: Toy example demonstrating the notion of "frontier". Green nodes are selected in the dashboard. The neighbors of the set of selected green nodes are the frontier nodes, shown in pink. Other unpicked nodes in the lattice are shown in gray.

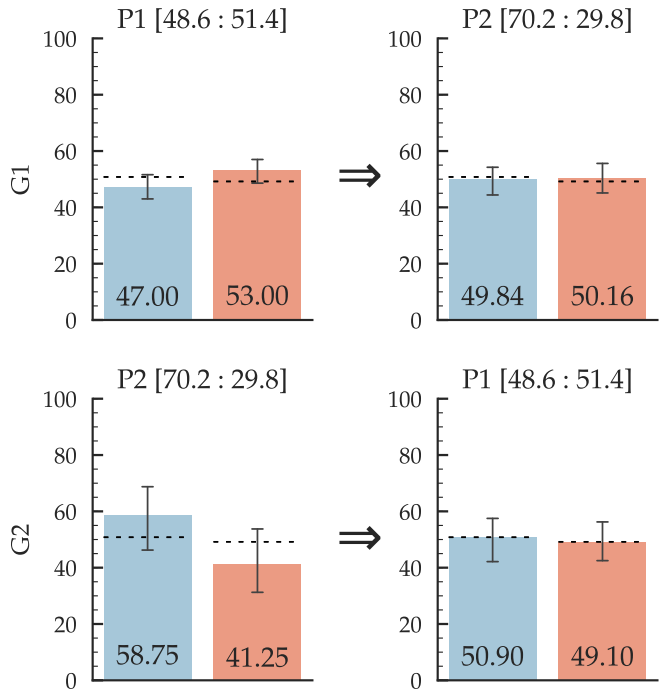

Fig. 3: Bar charts and error bars showing the mean and variance of participants' estimated distribution during the formative user study. Group 1 (G1) participants were shown the informative parent first, followed by the uninformative parent; whereas Group 2 (G2) participants were shown the uninformative parent first, followed by the informative parent. G1 participants closely approximated the unseen visualization immediately after seeing the informative parent. The ground truth values for the unseen visualizations are shown as dotted lines and printed in square brackets in the visualization title.

Informativeness: To model the informativeness of an observed parent in the context of an unseen visualization, we characterize the capability of the parent in predicting the unseen visualization. As highlighted in finding (2), an observed parent is *informative* if its data distribution closely follows the data distribution of the unseen child visualization, since the visualization helps the analyst form an accurate mental picture of what to expect from the unseen visualization. Specifically, we formulate the informativeness of an observed parent V_i^j of an unseen visualization V_i as the similarity between their data distributions measured using a distance function $D(V_i, V_i^j)$. The most informative parents V_i^* of an unseen visualization V_i are the ones whose data distributions are most similar to the unseen.

$$V_i^* = \underset{V_i^j}{argmin}\ D(V_i, V_i^j) \tag{1}$$

Since our finding (4) demonstrates how seeing multiple parents can lead to worser visualization understanding, we allow users to specify a threshold θ to control the degree of similarity for which a parent can be declared as an informative, such that the distance $D(V_i, V_i^{*,\theta})$ corresponding to the informative parents $V_i^{*,\theta}$ are at least $\theta\%$ close to its most informative parent:

$$V_i^{*,\theta} = \{V_i^j : \frac{D(V_i, V_i^*)}{D(V_i, V_i^j)} \geq \theta\} \tag{2}$$

For example, if $\theta = 0.8$, all parents of V_i whose similarity scores are at least 80% compared to V_i^* are deemed as *informative*. In Figure 1, while both visualization (b) and (d) are considered parents of visualization (h), only visualization (d) (Black) are considered the informative parent of visualization (h) (black female population), for any values of $\theta \geq 11\%$ via the Euclidean distance metric. Note that, our proposed system can work with different distance metrics such as cosine similarity and earth mover's distance. Without loss of generality, we chose to use Euclidean distance metric for the remainder of our paper.

Interestingness: While informative parents contribute to the prediction of an unseen visualization, the most interesting visualizations to recommend are those for which *even the informative parents fail to accurately predict the visualization*. Doris: Can we justify this based on our findings? To model the interestingness of an unseen visualization V_i in the context of an observed parent V_i^j, we characterize the deviation between their data distributions using a distance function $D(V_i, V_i^j)$. The unseen visualizations whose data distributions deviate from the observed informative parents are *interesting*. The most interesting unseen visualizations $V_\#$ are the ones that deviate most from their observed informative parents.

$$V_\# = \underset{V_i}{argmax}\ D(V_i, V_i^{*,\theta}) \tag{3}$$

In Figure 1, the most interesting visualization to recommend is the one corresponding to white female voters. This visualization significantly differs from its informative parent—the visualization corresponding to female voters. Doris: The argmax notation not necessary since we're just using this in our utility function. The election example is not convincing, the informative parent of white

female is actually white and not female. Also the differences are not too significant.

Subpopulation size consideration: The danger of spurious patterns and correlations in visualizations that contain small subpopulation size is a well-known problem in exploratory analysis [6]. We take two preventive measures to avoid including such misleading visualization in our dashboards. First, in the lattice generation process discussed in Section 4.2, we allow users to select an 'iceberg condition' [4] (δ) to adjust the extent of pruning on visualizations whose sizes fall below a certain percentage of the overall population size. Second, we downweigh the interestingness edge utility $U(V_i, V_i^j)$ between a parent V_i^j and a child visualization V_i by the ratio of their sizes:

$$U(V_i, V_i^j) = \frac{|V_i|}{|V_i^j|} \cdot D(V_i, V_i^j) \quad (4)$$

3.3 Problem Formulation

Given the lattice data model and the user model for visualization utility described above, the goal of our system is to generate a dashboard by selecting k visualizations from the lattice. We enforce that the generated dashboard satisfies several requirements:

1) Dashboard must include the overall visualization (topmost visualization with no filter applied) to serve as reference to the rest of the visualizations in the dashboard.
2) For each visualization except for the overall, at least one of its informative parents is included within the k visualizations. This excludes the uninformative parents as exemplified in black female example in the dashboard, especially since our findings 3 and 4 show that showing multiple, improper parents can mislead the participants, resulting in a higher variance across their estimations.
3) The selected k visualizations are collectively most "interesting" in presence of their informative parents as measured by the utility in Equation 4.

The problem of finding a connected subgraph in the lattice that has the maximum combined edge utility is known as the maximum-weight connected subgraph problem [4] and is known to be NP-Complete, via a reduction from the CLIQUE PROBLEM [16]. In Section 4.2, we discuss heuristic algorithms used for deriving a locally optimal solution for ensuring interactive runtime.

4 SYSTEM

4.1 System Architecture

We have implemented STORYBOARD as a Flask web application on top of a PostgreSQL database. In Figure 4, we present the system architecture of STORYBOARD, which consists of three core modules: the traversal module, the query module, and the statistics module. The interaction manager deals with the supported user interaction described in Section 5 and sends a request to the lattice module which contains several algorithms for generating and traversing the visualization lattice described in Section 4.2. For generating the visualization lattice, the lattice module passes a list of data subsets corresponding to visualizations to be generated to the query module. The query module translates these visualizations into queries, and then optimizes (by grouping) and executes the queries. The statistics module is an optional module that allows the lattice module to prune low-utility visualizations without actually generating them. Specifically, it generates coarse statistics for the unexplored visualizations based on the current list of explored visualizations. Finally, the dashboard renderer takes the resulting visualizations to be included in the dashboard and perform any rendering preprocessing procedures for display and navigation of the dashboard as described in Section 5.1.

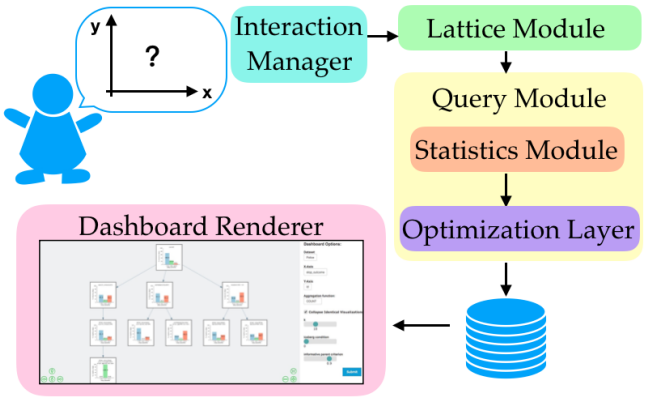

Fig. 4: System Architecture of STORYBOARD. User starts with x and y axes of interest and requests for k visualizations in the dashboard. The request is processed by generating the lattice with the help of the querying module, visualization selection through the lattice traversal algorithms, and finally the dashboard is displayed at the frontend through the dashboard renderer.

4.2 Algorithms

We give an overview of our algorithms by first discussing the approaches to generate the visualization lattice, and then presenting a high-level overview of our traversal algorithms.

Lattice Generation. Our system supports two variants of traversal algorithms based on the lattice generation procedure—offline variants that first generate the complete lattice and then work towards identifying the maximum utility solution, and online variants that incrementally generate the lattice and simultaneously identify the solution. The offline variants are appropriate for datasets with a small number of low-cardinality attributes, where we can generate the entire lattice in a reasonable time; whereas the online variants are appropriate for datasets with large number of high-cardinality attributes, where we incrementally generate a partial lattice.

Lattice Traversal. Given the materialized lattice, the objective of the traversal algorithm is to find the connected subgraph in the lattice that has the maximum combined edge utility. Here, we discuss the *frontier greedy* algorithm which is used for generating the dashboards for our user study and defer our discussion on the details of other algorithms that we have developed to the technical report.

As described in Algorithm 1, our algorithm obtains a list of candidate nodes known as the *frontier* nodes (pink in Figure2 left), which encompasses all neighbors of nodes in the existing subgraph solution. Any of the nodes in the frontier can be added to the current solution since their informative parent is guaranteed to be present in the solution. The `getFrontier` function scans and adds all children of leaf nodes of the current dashboard as part of the frontier. In the online version, it additionally checks for

4. The terminology is used in the discussion of iceberg cubes in OLAP literature [28].

each child whether its informative parent is present in the current dashboard. At each step, our algorithm greedily picks the node with the maximum utility from the frontier to the current solution, and updates the frontier accordingly.

Algorithm 1 Frontier Greedy Algorithm

```
procedure PICKVISUALIZATIONS(k,lattice)
    dashboard ← { V_overall }
    while |dashboard| < k do
        frontier ← getFrontier(dashboard,lattice)
        maxNode ← getMaxUtilityNode(frontier)
        dashboard ← dashboard ∪ {maxNode}
    return dashboard
```

5 User Interaction

Fig. 5: Overview of the STORYBOARD interface for the Police Dataset [1]. Users can select x and y axes of interest, as well as a choice of an aggregation function. Default values are set for system related parameters such as the number of visualizations to show in the dashboard (k), iceberg condition for pruning (δ), and informative parent criterion (θ), which can be adjusted by the users via the sliders if needed.

Figure 5 shows an overview of the STORYBOARD interface. After the user selects the x and y axes of interest, aggregation function, and optional system parameter settings, an initial dashboard of k visualizations is displayed on the canvas, such as the one seen in main canvas of Figure 5. The system provides toolbar buttons with keyboard binding for zooming in, out, and extent, as well as moving around the canvas. Alternatively, users can zoom and pan with mouse click and scroll.

After browsing through the visualizations in the dashboard, users may be interested in getting more information about a particular node. STORYBOARD supports a mechanism for users to request additional summarizations based on a chosen visualization of interest. As shown in Figure 6 (left), the analyst starts with a 5-visualization dashboard on a police stop dataset [1]. The dataset contains records of vehicle and pedestrian stops from law enforcement departments in Connecticut, dated from 2013 to 2015. The analyst learns that for the drivers who had contraband found in the vehicle, the arrest rate for drivers who are 60 and over is surprisingly higher than usual, whereas for Asian drivers the arrest rate is lower. In addition, he is also interested in learning more about the other factor that contribute to high arrest rate: duration=30+min. He clicks on the corresponding visualization and requests for 2 additional visualizations. Upon seeing the updated dashboard in Figure 6 (right), he learns that similar to the selected visualization, any visualization that involves the duration=30+min filter results in high ticketing and arrest rates. This implies that if a police stop lasts more than 30 minutes, the outcome would more or less be the same, independent of other factors, such as driver's race or age. STORYBOARD uses the same models and algorithms as before, except the root node is now set as the selected visualization, rather than the overall visualization. This node expansion capability is similarly motivated by the idea of *iterative view refinement* in other visual analytics system [11], [26], which is essential for the users to iterate on and explore different hypotheses.

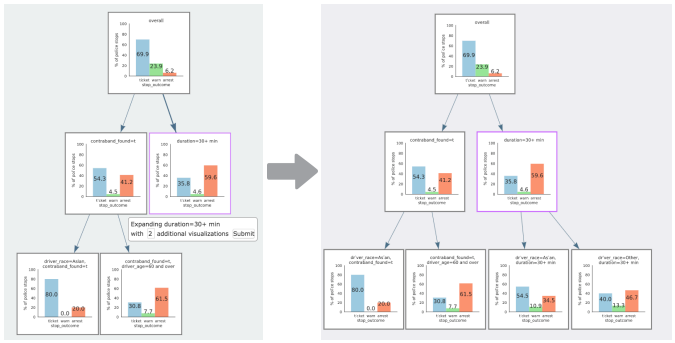

Fig. 6: Left: Original k=5 dashboard with the duration=30+min visualization clicked. A pop-up is displayed to submit the request for additional summary visualizations to be generated. Right: Resulting dashboard after requesting for 2 more visualizations based on the visualization of interest.

5.1 Assistive tools for visualizing large lattices

Due to the amount of space occupied by the hierarchical layout when the number of visualizations gets large, we have developed tools to help users navigate through different parts of the dashboard interactively.

Navigation Minimap: When the user zooms in on the dashboard, an overview mini-map is shown on the upper left-hand side of the canvas to help users identify which region of the dashboard they are currently exploring, as shown in Figure 8.

Collapsed visualizations: One observation that we found across several datasets was that many visualizations had identical distributions, which resulted in lots of wasted space. Apart from their attribute name, these visualizations are not very informative for the users, therefore, we offer an option to collapse these visualization, as demonstrated in Figure 7. A visualization can be collapsed if it has more than one redundant sibling and does not have any children, so that there are no hidden stories due to lower-level dependencies. As shown in Figure 8, collapsed nodes can be easily identified by an orange border and the details of which visualizations are in the collapsed node are displayed when the user hovers over the visualization.

6 User Study Evaluation

6.1 Procedure

Given that our formative study motivated our metrics and constraints used in the problem formulation, we further evaluate the utility of our tool by performing a user study focusing on addressing the research questions:

- RQ1: How effective is our tool at discovering key insights within a given dataset?
- RQ2: How effective is our tool in providing analysts with task-specific insights? (including identifying important features

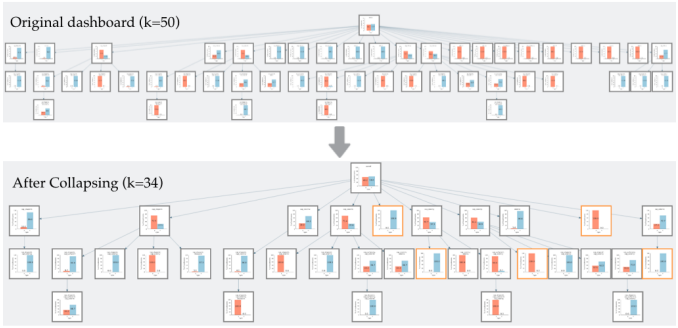

Fig. 7: An example of the k=50 dashboard for the mushroom dataset [22], which contains type={posionous, edible} on the x-axis. The collapsed dashboard (bottom) removed 16 redundant visualizations from the original dashboard (top).

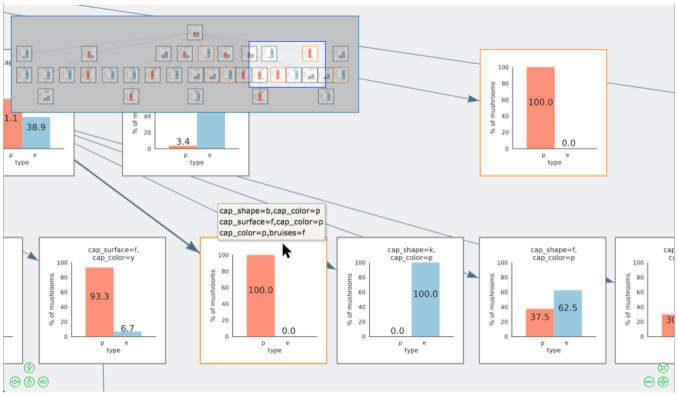

Fig. 8: Zoomed-in version of Figure 7 showing the labels of a collapsed visualization when user hovers over the visualization. The navigation minimap is shown in the top-left to help users navigate through the large dashboard.

for prediction tasks and estimating the distribution of an unseen visualization)

- RQ3: How useful are the visualizations in the recommended dashboard to analysts?

We recruited 18 participants who have prior experience working with data. Participants include undergraduate and graduate students, researchers, and data scientists, with 1 to 14 years of data analysis experience (average = 5.61). There were 8 female participants and 10 male participants. No participants reported prior experience in working with the two datasets used in the study.

In this between-group study, participants are randomly assigned two of the three following conditions.

1) STORYBOARD: The dashboards for this condition is generated by the frontier greedy algorithm (described in Section 4.2) and displayed in a hierarchical layout (as seen in Figure 5). In order to establish a fair comparison with the two other conditions, we deactivated the interactive node expansion and dashboard navigation functionalities described in Section 5, especially since the k=10 dashboard was small enough to function without the navigation tools.
2) CLUSTER: K-Means clustering is performed on the dataset with k clusters, corresponding to k, the number of visualizations to be shown in the dashboard. For each representative cluster, we select the visualization that has the least number of filter conditions for interpretability[5] and display them in a 5x2 table layout.
3) BFS: Starting from the overall visualization, k visualizations is selected in level-wise order: sequentially adding visualizations at the first level with 1-filter combination one at a time, proceeding with the 2-, 3-, etc. filter combinations, until k visualizations have been added to the dashboard. This baseline is designed to simulate the dashboard generated by a meticulous analyst who has generated all possible visualization combinations to display on a visualization dashboard and is scrolling through the dashboard to browse and inspecting every visualization. The chosen visualizations are displayed in a 5x2 table layout in the traversed order.

All generated dashboards had k=10 visualizations. We randomize the ordering for each task combination to prevent confounding learning effects.

The study begins with a 5 minute tutorial using dashboards generated for the Titanic dataset. To prevent participant's bias, participants were not provided an explanation of how the dashboard is generated and why the visualizations were arranged in a particular way. Then, participants proceeded onto the Police dataset [17] , which contains a total of 312948 records of vehicle and pedestrian stops from law enforcement departments in Connecticut, dated from 2013 to 2015. We generate a dashboard of visualizations with bar charts with x-axis as the stop outcome (whether the police stop resulted in a ticket, warning, or arrest/summons) and y-axis as the percentage of police stops that led to this outcome. The attributes in the dataset include driver gender, age, race, and the stop time of day, whether a search was conducted, and whether contraband was found.

Participants were given some time to read through a worksheet containing descriptions of the data attributes. Participants were given an attention check question where they are asked to find and read off the values of a visualization in the dashboard. In the main experiment, participants were asked to accomplish the following tasks in the prescribed order below: Aditya: why were these selected?Doris: I reserved the discussion of why these task were selected to the results and discussion section, since they make more sense in the context on the results. I thought the methods section usually reads more like a recipe.

Retrieval: Participants were asked to talk aloud as they interpret the visualizations in the dashboard and mark each visualization as either interesting, not interesting, or leave it as unselected.

Attribute Ranking: Participants were given a worksheet with all the attributes listed and asked to rank the attribute in order of importance in contributing to one particular x-axis value (e.g. stop outcome = arrest, autism = yes)

Shallow Prediction: Participants were given a separate worksheet and asked to draw an estimate for a visualization that is not present in the dashboard. The visualization to be estimated is "shallow" in the sense that it is a visualization with 2 filter combinations, with one parent present in the given dashboard. After making the prediction, participants are shown the actual data distribution and asked to rate on a Likert scale of 10 how surprising was the result.

Deep Prediction: Similar to the shallow prediction, except that the visualization to be estimated is "deep" in the sense that it is a 3-combination filter, with only one parent in the given dashboard.

5. Due to this requirement, the overall visualization is guaranteed to be picked as one of the displayed visualizations.

The second dataset in the study is the Autism dataset [8], which includes the result of autism spectrum disorder screening for 704 adults. The attributes in the dataset are binary responses to 10 questions that are part of the screening process. Participants are not given the descriptions of the questions nor the answers corresponding to the labels. We generate dashboard visualizations based on whether the participant is diagnosed with autism or not. We repeat the same procedure described above for the Autism dataset. At the end of the study, we asked two open-ended questions regarding the stories and insights that they have learned and what they like or dislike about each dashboard.

6.2 Results

In order to evaluate the efficacy of our system against the two baselines, we will first examine the quantitative results to address RQ1 and RQ2 and then discuss the qualitative findings to address RQ3. Doris: In general, we might have to make better connection between the RQs and the study results.

Retrieval (RQ1): Using the click-stream data logged from the user study, we record whether each user is interested, not interested, or have not selected a visualization in the dashboard. Since we do not have a objective ground truth on which visualization is interesting or not interesting, we devise a voting-based measure that measures how interesting is a visualization amongst all participants. Here i indexes the visualization and j indexes the user. As shown in Equation 5, we assign a vote δ_{ij} of 1 if a user is interested in a visualization, 0 if they leave it unselected, and -1 if they are not interested in a visualization.

$$\delta_{ij} = \begin{cases} 1 & \text{interested} \\ 0 & \text{unselected} \\ -1 & \text{not interested} \end{cases} \tag{5}$$

We obtain a consensus score for each visualization to measure how frequently the visualization is regarded as interesting by summing over all user's vote on that visualization.

$$\text{consensus}(V_i) = \sum_{j \in user} \delta_{ij} \tag{6}$$

Given a consensus measure of how interesting a visualization is, we can define a rating score which measures how good a particular user's rating is, by taking the product of the consensus interestingness score and the rating value, as shown in Equation 7. Intuitively, a rating should be rewarded more if it has retrieved interesting visualization agreed by many other users, likewise, ratings that does not retrieve such visualizations should be penalized more heavily.

$$\text{rating score}(V_{ij}) = votes(V_i) \cdot \delta_{ij} \tag{7}$$

Table 2 summarizes results of rating scores averaged over the tasks that the user performed.

Dataset	STORYBOARD	CLUSTER	BFS
Police	62	52	99
Autism	213	180	114

TABLE 2: Average consensus-agreement score for different algorithm and datasets.

Due to the highly subjective nature of the retrieval task, the interestingness selection for the Police dataset was biased by participant's priors and intuition about the attributes. For example, while all participants who have seen the visualization "duration=30+min" verbally noted that stop duration is a crucial factor that leads to arrest, only 4 users marked it as interesting. 5 participants marked the visualization as not interesting and 4 left it unselected, because the visualization was not very surprising as it agreed with their intuition that "*if the police stop is taking a long time, something has probably gone wrong*".

Since the attributes in the Autism dataset are simply question numbers, participants could not associate any priors to their interestingness selection. In this prior-agnostic case, participants who used STORYBOARD found more visualizations of interest that corresponded to the consensus, indicating that there are more interesting visualizations picked out by STORYBOARD than compared to the two baseline-generated dashboards.

Attribute Ranking (RQ2): To determine attribute importance ranking for a dataset, we computed the Cramer's V statistics between attributes to be ranked and the attributes of interest. Cramer's V test makes use of the chi-square statistics to determine the strength of association between attributes. Using the ranks determined by Cramer's V as ground truth, we compute the normalized discounted cumulative gain (NDCG@k) of each participant's ranking average over all tasks[6], as detailed in Table 3. We see that

Dataset	STORYBOARD	CLUSTER	BFS
Police	0.63	0.45	0.84
Autism	0.50	0.30	0.24

TABLE 3: NDCG@10 scores for the attribute ranking task.

STORYBOARD performs better than clustering in both cases. Since clustering seeks for a set of visualization that exhibits diversity in the shape of the data distribution, it results in visualizations with many filter combination, which is hard to interpret without appropriate context to compare against. BFS performs better than STORYBOARD in the Police dataset, but not in the Autism dataset. BFS may have performed better than STORYBOARD in the Police dataset for a combination of two reasons: 1) since BFS exhaustively displays all attributes sequentially, for the Police dataset it had happened to select several of the important attributes (related to contraband and search) to display as the first 10 visualizations and 2) as discussed earlier, some participants had priors on the data attribute which influenced their ranking. However, with a budget of k=10, only visualizations regarding questions 1-5 fit in the dashboard for the Autism dataset, so the poor ranking behavior comes from the fact that the BFS generated dashboard failed to display the important attributes (questions 6 and 9) given the limited budget.

Attribute ranking tasks are common in feature selection and other data science tasks, in general, our results indicate that using STORYBOARD, users gain a better understanding of variable influence and correlation.

Prediction (RQ2): In order to measure how accurate participants' decisions are, we computed the Euclidean distance between their predicted distributions and ground truth data distributions. As shown in Figure 9 (top), all the shallow predictions made by using information from the STORYBOARD is closer to the actual distribution compared to the baselines. This aligns with our findings in the formative study and indicates that users are able to more accurately reason about how unseen data would behave with STORYBOARD.

STORYBOARD did not perform as well compared to the baselines for the Autism deep prediction task. One possible reason

6. Since participants are asked to examine all attributes, the k for NDCG@k corresponds to total number of attributes in that dataset.

for this is due to the fact that the shallow and deep prediction tasks for the Autism dataset were correlated. Therefore, after learning about the insights that answering 1 on question 9 results in a very high probability for an autism diagnosis, some participants made use of that information when tackling the subsequent deep prediction task. By discussing with the baseline participants on how they have obtained the prediction estimates, they described how surprised they were by the finding in the shallow prediction and therefore adjusted the autism diagnosed values to be higher to compensate for their mistake in the subsequent deep prediction task.

As shown in Figure9 (bottom), in general, we find that participants who used STORYBOARD reported that they were less surprised when the unseen visualization is revealed, which again indicates that participants had a more accurate mental model of prediction.

We also compute the mean and standard deviation of the participant's prediction aggregated across the same task. In this case, low variance implies that any user who reads the dashboard is able to provide consistent predictions, whereas high variance implies participants have relied on different priors or guessing to perform the prediction, often because the dashboard did not convey a clear data-driven story. These trends can be observed in Figure 10, where the prediction variance amongst participants who used STORYBOARD is much lower than the variance from the baselines.

6.3 Qualitative results (RQ3)

We analyzed the transcriptions of the study recordings through an open coding process by two of the authors. For each task performed by the participants, a binary-valued code is assigned to indicate whether or not the participant engaged in the particular event (action or thought process). We will refer to participants engaging in a dashboard created by algorithm={1,2,3}={STORYBOARD, CLUSTER, BFS } on dataset ={A,B}={Police, Autism} with the notation [Participant.DatasetAlgorithm].

STORYBOARD promotes distribution-awareness by provoking comparisons against more informative contextual references.

We first examined the thematic codes regarding how participants understood the context of the visualization distribution. In particular, we were interested in the types of visualizations that participants compared against in order to form their expectations regarding how other visualizations should be distributed. We define this property of visualization understanding as *distribution-awareness* and the visualizations that are compared against as the *contextual reference*. Via the thematic coding, we uncovered four main classes of contextual references, described below using the example visualization `gender=F, race=White, age=21-30` (in order of most to least similar):

1) Parent : Comparison against a visualization with one filter criterion removed (e.g. `gender=F, race=White`)
2) Siblings : Comparison against a visualization that share the same parent. In other words, the filter types are the same, but with one criterion that inherit a different value. (e.g. `gender=M, race=White, age=21-30`)
3) Relatives : Comparison against a visualization that share some common ancestor (excluding overall), but not necessarily the same parent. In other words, these visualizations share at least one common filter type, but with more than one criterion that inherit a different value. (e.g. `gender=F, race=White, age=60+, search conducted=T`)
4) Overall : Comparison against the distribution that describes the overall population (no filters applied).

Studying participants' use of contextual reference reveals inherent challenges in dashboard selection through BFS and CLUSTER. As shown in Table 4, for CLUSTER, participants mainly compared against relatives and the overall. Since CLUSTER optimizes for diversity of shape distributions amongst the visualization, the selected visualization had up to 4 filters and were disconnected from each other. For this reason, in many cases participants could only rely on relatives and overall as contextual references to gain distribution-awareness. For example, [P4.A2] dislike how "a lot of [the visualizations] are far too specific. [Pointing at visualization consisting of 4 filters with a 100% bar for warning] This is not very helpful. You can't really hypothesize that all people are going to be warned, because it is such a specific category, it might just be one person". He further explained how he "would not want to see the intersections (visualizations with multi-variable filters) at first and would want to see all the bases (visualization with one variable at a time) then dig in from there." In addition, the lack of informative contextual reference in the CLUSTER dashboard is reflected in the high variance and deviation of the predicted visualization results.

Algorithm	Overall	Parent	Sibling	Relative	Total
STORYBOARD	11	12	8	0	31
CLUSTER	8	4	0	7	19
BFS	8	0	5	1	14

TABLE 4: Number of participants who made use of each contextual parents, summed across the two datasets. Participant behavior shows a similar trend in individual datasets.

For BFS, most comparisons were among overall and siblings. Due to the sequential, level-wise picking approach, in all cases for the BFS dashboard generated, the overall corresponded to the immediate parent, so they are not explicitly recorded as parent. While the overall and sibling comparisons can be informative, due to the limited budget k, not all first-level visualizations were displayed in the dashboard. These incomplete comparisons can result in flawed reasoning, as observed in the Autism shallow prediction task described earlier. In contrast, for STORYBOARD, users mainly compared against the overall and parents, while some also exploited sibling comparison information to make a less certain guess. We also find that more participants make comparisons in total using STORYBOARD than compared to CLUSTER and BFS.

Hierarchical layout leads to more natural contextual comparisons compared to table layout.

As described in the previous section, contextual parents are important in establishing distribution-awareness for understanding the dataset. Participants cited hierarchical layout as one of the key reasons why it was easier to follow contextual reference in STORYBOARD. Based on the hierarchical layout in STORYBOARD, users were able to easily interpret the meaning of the dashboard, even though they were never explicitly told what the edge connections between the visualizations meant. For example, [P1.A1] stated that "the hierarchical nature [is] a very natural flow...so when you are comparing, you don't have to be making those comparisons in your head, visually that is very pleasing and easy to follow." Likewise, P9 described how the hierarchical layout she saw for the Autism dataset was a lot easier to follow than the Police dataset shown in the table layout:

> If I had to look at this dataset in the format of the other one, this would be much more difficult. It was pretty hard for me to tell in the other one how to organize the tree, if there was even a tree to be organized. I like this layout much better, I think this layout allows me to approach it in a more

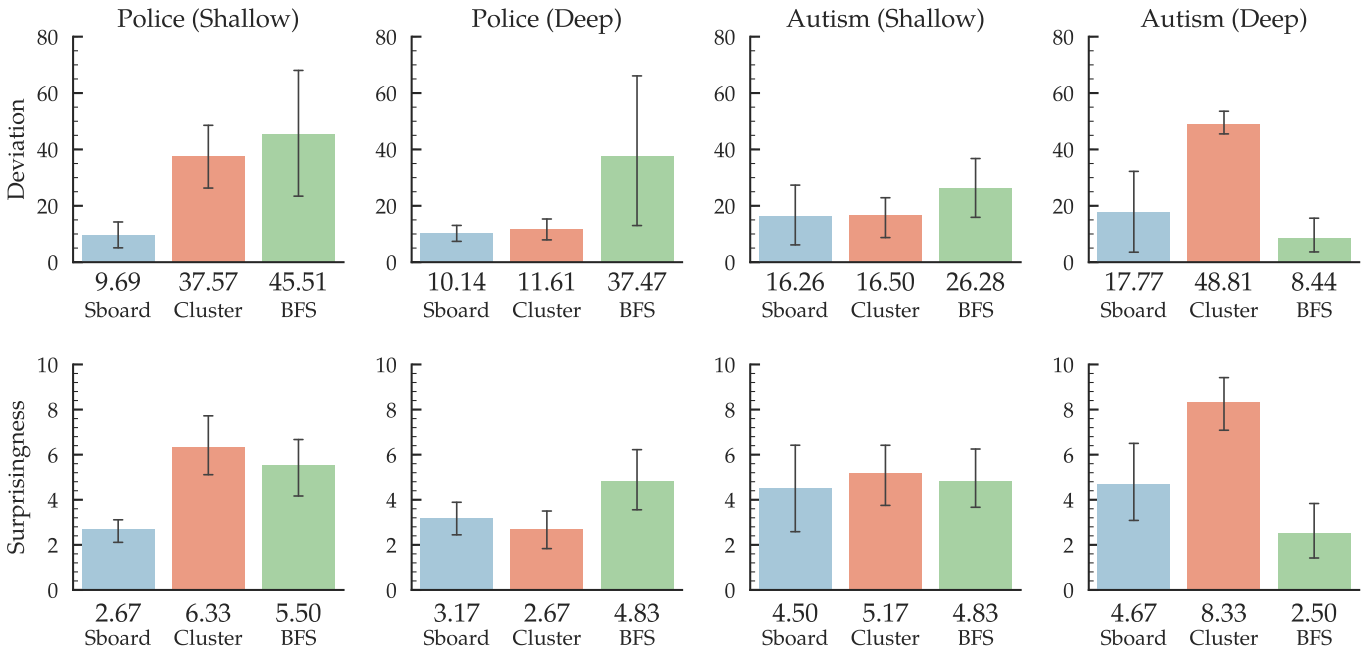

Fig. 9: Top: Euclidean distance between predicted and ground truth. Bottom: Surprisingness rating reported by users after seeing the actual visualizations on a Likert scale of 10.

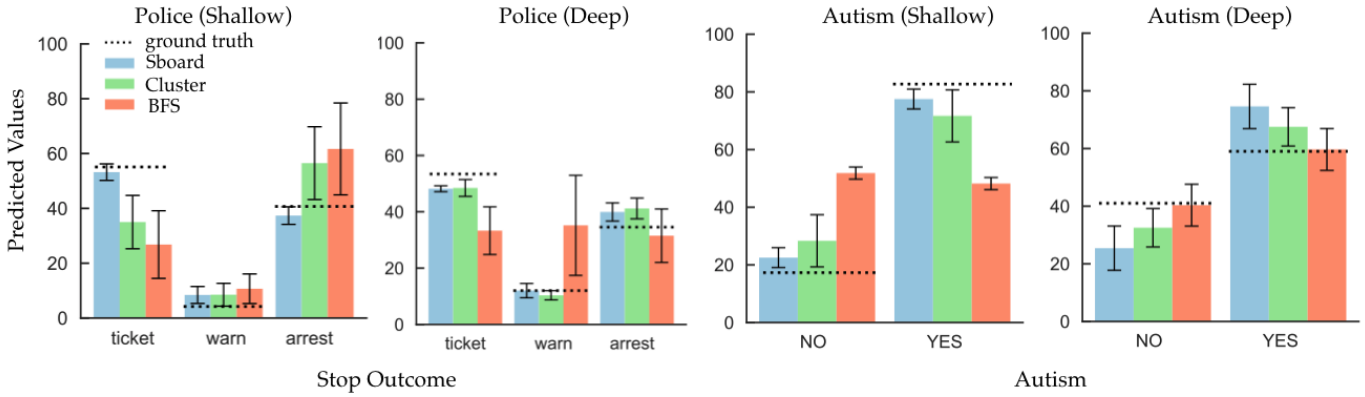

Fig. 10: Mean and variance of predicted values. Predictions based on STORYBOARD exhibits lower variance (as indicated by the error bars) and great proximity to the actual values (dotted).

> meaningful way. I can decide, what do I think matters more: the overall trend? or the super detailed trends? and I know where to look to start, in the other one, every time I go back to it, I would say, where's the top level, where's the second level? I mentally did this. Like when you asked me that first question, it took much longer to find it, because I literally have to put every chart in a space in my head and that took a lot longer than knowing how to look at it.

At the end of the study, some participants for BFS and CLUSTER sketched and explained how they would like the layout of the visualizations to be done. Participants expressed that they wanted "groupings" or layouts that arranged visualizations with the same attribute together. Other participants advocate for isolating the overall visualization outside of the dashboard table for facilitating easier comparisons. Both of these provides further motivation for our hierarchical layout and the idea of the collapsed visualizations as described in Section 5.

Since we did not inform participants how the dashboards were generated, it was also interesting to note that some participants thought that the dashboards were hand-picked by a human analyst and described what this person's intentions were (e.g. "It seems like the researcher who created this dashboard was specifically looking at people of Asian descent and people who are 60 or older." [P7.A1]). We encoded this phenomenon by looking at instances where a participant either explicitly referring to a person who picked out the dashboard or implicitly described their intentions through personal pronouns. A total of 5 different participants referred to the dashboard generated by STORYBOARD as generated by a human, whereas there was only 1 participant for CLUSTER and none for BFS made such remarks. At the end of the study, many were surprised to learn that the STORYBOARD dashboard was actually picked out by an algorithm, indicating that STORYBOARD could automatically generate convincing dashboard stories that were similar to a dashboard that was authored with intention.

Improper contextual reference can lead to misleading insights.

While comparisons are essential for data understanding, choosing the wrong contextual reference for comparison could lead to misleading insights. In particular, when a visualization that is composed of multiple filter conditions is shown in a dashboard

created using CLUSTER, 5 out of 12 participants for both datasets had a hard time interpreting the meaning of the filter, whereas there was only 1 for STORYBOARD and BFS. This is due to the fact that CLUSTER dashboards are seemingly random to the users, whereas BFS and STORYBOARD both have some natural ordering. In addition, when examining visualizations with many filters and contain bars that exhibit extreme values (bars with 100% or 0% in one or more categories), 4 CLUSTER participants did not realize that charts with multiple filters may have a smaller subpopulation size. This issue stems from the fact that the contextual reference used for comparison was the overall population, however the unseen parent subpopulation may have behaved very differently. The fallacy was observed to be more severe for the Autism dataset, where participants had less intuition on the expected attribute behavior. In contrast, 6 of the participants using STORYBOARD recognized that while these extreme-valued visualizations may be interesting, they were less certain due to the unknown subpopulation size and should be investigated further. For example, [P1.A1] noted that a visualization with warning=100% caught her eye, "but I don't know what the N is, maybe it's one person, this makes me a little skeptical, that makes me want to go back to the raw data and look at what is the N and what drives something so drastic?". Since BFS dashboards only displayed first-level visualizations, participants for BFS did not see such visualizations during the study session, so we did not see signs of this fallacy for BFS participants.

Limitations of STORYBOARD

As described earlier, since the details of how the dashboard was obtained was not explained to the users during the study, some users expressed that they were initially confused by STORYBOARD since not all variables were present in the dashboard and others found it confusing that the addition of filters did not always correspond to the same variables. For example, [P2.A1] criticized how the dashboard was intentionally selected to be biased:

> I feel like this one, not all the data is here, so we are already telling a story, you are trying to steer the viewer to look at certain things. And the focus seems to be on where the arrest rate is high. You probably could have found other things that led to ticket being high, but you didn't pull those out. You are trying to see if there are other factors that leads to more arrests.

This sentiment is related to participants' desire to perform their own ad-hoc querying alongside the dashboard to inspect other related visualizations for verifying their hypothesis. For example, [P7.A1] wanted to inspect all other first-level visualizations for driver's race to assess its influence. [P7.A1] expressed that while he had learned many insights from the dashboard, "the only thing I don't like is I can not control the types of filter, which is fixed." Outside the context of the user study, it is essential to explain how STORYBOARD are picking the visualizations in a easy and interpretable manner to establish a sense of summarization guarantee for the users and help them make better inferences with the dashboard.

As discussed earlier, subpopulation size is important in establishing the 'credibility' of a visualization. While subpopulation size is taken into account implicitly in our objective (as described in Section 3.2), we should design interfaces that can convey the notion of subpopulation size in our dashboard, either explicitly displayed as text when hovering over the visualization or changing the size or background color of the visualizations to encode subpopulation size.

7 DISCUSSION

7.1 Statistical Paradoxes

Doris: make title full sentences Visualizations are powerful representations for studying different distributions or patterns in a dataset, but our human intuition could often mislead us when it comes to interpreting those patterns [6], [25]. Several statistical paradoxes can lead analysts to draw incorrect conclusions from observed visualizations, including Simpson's paradox as discussed in the introduction. The key reason why many of these paradoxes emerge is the *incompleteness* of the observed data or lack of focus on relevant informative subsets of the data. For example, Simpson's paradox arises in the presence of an unseen confounding variable. We assert Doris: too strong of a sentence that distributional awareness can be useful in avoiding such statistical paradoxes. If an analyst is aware of all distributions in a given dataset, he/she is less prone to many statistical paradoxes. However, given the large number of dimensions and high cardinality of these dimension in modern datasets, it is not possible for an analyst to explore and memorize all distributions. Therefore, a more evolved approach is to be aware of the exceptional distributions. In this work, we propose a first step towards this goal, where we identify the exceptional distributions in terms of their informative references. The remaining (unseen) distributions in the dataset are rather unsurprising and can be inferred from the visualizations in the dashboard. Doris: I would recommend first talk about issue with large dimension + danger of multiple hypothesis testing + incomplete testing, point out problem, then talk about how our system resolves this.

7.2 Structural Insight

Our proposed dashboard consists of a hierarchy of visualizations, where each visualization is linked to its most informative parent. The shape or structure of the hierarchy contains useful information that augments the information learned from the visualizations and aid distribution awareness and understanding. Doris: what's interesting here is that while many work have looked at visualization presentation, layout of presentation never considered, we find in Sec 5 that this is actually important and can encode info. For example, the depth and branching factor of the hierarchy could inform a user regarding the configuration of insights. Deep hierarchies contain long paths, i.e., insights are present at lower level visualizations with multiple constraints. In contrast, bushy hierarchies (with high branching factor) contain cases where multiple visualizations have the same informative parent and they differ from that parent. Doris: do we have examples from the study that support this? We assert that the depth and branching factor could be a meaningful constraint in our problem formulation Doris: too strong of a sentence. Some applications for example, funnel exploration require studying deep hierarchies, whereas others for example, building decision trees require studying bushy hierarchies. A natural extension of our current problem formulation is to allow users to select the depth and branching factor for the hierarchy.

7.3 Other Visualization Lattices

In this work, we explore the space of data subsets to generate our visualization lattice. Note that it is possible to explore the space of dimension attributes in x-axis to generate a different visualization lattice. In particular, given a combination of dimension attributes $X = \{X_1, \ldots, X_n\}$, adding one or more new dimensions in X will generate a new combination. An ancestor-descendant relationship exists between these dimension combinations, following

the same principles of Section 3.1. These relationships lead to a new lattice, which we call the dimension combination lattice. Our informative deviation based approach could be used for traversing the dimension combination lattice. However, we observe that most users do not visualize more than two attributes in x-axis. Therefore, traversing the dimension combination lattice is not very useful for most applications. Doris: I think 6.2,6.3 don't tie well with the rest of the paper. It sounds like stretching our own ideas rather than being motivated by the work done in this paper. Other potentially more relevant discussion: distribution awareness and how it might be useful in other contexts? Decision trees?

8 Related Works

Storytelling with visualization sequences: Visualizations are often arranged in sequence to narrate a data story. Existing work on visualization sequences and storytelling have studied the structures of narrative visualizations [13], [23], effects of augmenting exploratory information visualizations with narration [7] and, more recently, ways to automate the creation of visualization sequences [12], [14]. Most of these work have adopted a linear layout (motivated by slidedecks) to present the visualization sequences. Hullman et al. [13] found that most people prefer visualization sequences structured hierarchically based on shared data properties such as levels of aggregation. Kim et al. [14] models relationships between charts by empirically estimating transition (edge) cost between moving from one visualization (node) to another. They find that participants preferred "*starting from the entire data and introducing increasing levels of summarization*". Our work is the first to automatically sequence visualizations in a hierarchical layout for summarizing the space of data subsets.

8.1 Visualization recommendation

Visualization recommendation systems select appropriate visualizations to show based on an objective function. The metrics considered by these systems can largely be divided into two categories: perceptual or data-driven. The first type of recommendation system selects visualizations based on its visual effectiveness and expressiveness [15], [26]. Our work is more related to the latter category of systems which uses statistical measures computed based on the underlying data subset, such as cognostics or deviation. Anand et al. [5] used randomized permutation tests to automatically select partitioning variables to display visualizations exhibiting patterns that are different from the input visualization as determined by its cognostic score. Vartak et al. [24] finds interesting visualizations by a deviation-based measure between the user's query view and reference view, given a query of interest. While both existing systems require the analyst to input a visualization of interest as a query, our paper extends the deviation-based idea to establish user's expectation using informative parent enabling StoryBoard to traverse the visualization lattice in search of a connected, maximally informative and interesting story without the need for an input query.

8.2 Data Exploration of OLAP Data Cubes

The challenge of manual, unguided search in online analytical processing (OLAP) applications have been well studied in the context of data cube exploration by Sarawagi et al. [18], [19], [20]. To address this challenge, they simplify the search by identifying "interesting" regions of a data cube. These techniques includes precomputed statistics accounting for the surprisingness attributed to neighboring paths to cell and amount of deviation from constrained maximum entropy-based expectations. While these interesting sub-cubes correspond to finding filter combinations for constructing the aggregate visualization in StoryBoard, our lattice search space enforces fixed x, y and aggregation as well as connectedness during traversal to discover more interpretable stories.

9 Conclusion

References

[1] Connecticut racial profiling prohibition project data portal. http://ctrp3.ctdata.org/. Accessed: 2018-07-16.

[2] Titanic: Machine learning from disaster. Kaggle.

[3] N. Alipourfard, P. G. Fennell, and K. Lerman. Can you Trust the Trend? *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining - WSDM '18*, pp. 19–27, 2018. doi: 10.1145/3159652.3159684

[4] E. Althaus, M. Blumenstock, A. Disterhoft, A. Hildebrandt, and M. Krupp. Algorithms for the Maximum Weight Connected k-Induced Subgraph Problem. 5573:313–321, 2009. doi: 10.1007/978-3-642-02026-1

[5] A. Anand and J. Talbot. Automatic Selection of Partitioning Variables for Small Multiple Displays. 2626(c), 2015. doi: 10.1109/TVCG.2015.2467323

[6] C. Binnig and L. D. Stefani. Towards Sustainable Insights or why polygamy is bad for you. *Cidr*, 2017.

[7] J. Boy, F. Detinenne, and J.-D. Fekete. Storytelling in Information Visualizations : Does it Engage Users to Explore Data? *CHI 2015*, pp. 1449–1458, 2015.

[8] F. Fayez Thabtah. Autism screening adult data set. UCI machine learning repository, 2017.

[9] Y. Guo, C. Binnig, T. Kraska, and T. U. Darmstadt. What you see is not what you get ! Detecting Simpson ' s Paradoxes during Data Exploration. *HILDA 2017 - Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, 2017.

[10] J. Heer and B. Shneiderman. Interactive Dynamics for Visual Analysis. *Queue*, 10(2):30, 2012. doi: 10.1145/2133416.2146416

[11] E. Hoque, V. Setlur, M. Tory, and I. Dykeman. Applying Pragmatics Principles for Interaction with Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics*, (c), 2017. doi: 10.1109/TVCG.2017.2744684

[12] J. Hullman, S. Drucker, N. Henry Riche, B. Lee, D. Fisher, and E. Adar. A deeper understanding of sequence in narrative visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2406–2415, 2013. doi: 10.1109/TVCG.2013.119

[13] J. Hullman, R. Kosara, and H. Lam. Finding a Clear Path : Structuring Strategies for Visualization Sequences. 36(3), 2017.

[14] Y. Kim, K. Wongsuphasawat, J. Hullman, and J. Heer. GraphScape: A Model for Automated Reasoning about Visualization Similarity and Sequencing. *Proc. of ACM CHI 2017*, 2017. doi: 10.1145/3025453.3025866

[15] J. D. Mackinlay, P. Hanrahan, and C. Stolte. Show Me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1137–1144, 2007. doi: 10.1109/TVCG.2007.70594

[16] A. G. Parameswaran, H. Garcia-Molina, and J. D. Ullman. Evaluating, combining and generalizing recommendations with prerequisites. *Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM '10*, p. 919, 2010. doi: 10.1145/1871437.1871555

[17] E. Pierson, C. Simoiu, J. Overgoor, S. Corbett-Davies, V. Ramachandran, C. Phillips, and S. Goel. A large-scale analysis of racial disparities in police stops across the united states, 2017.

[18] S. Sarawagi. Explaining differences in multidimensional aggregates. *Proceedings of the VLDB Endowment*, pp. 42–53, 1999.

[19] S. Sarawagi. User-adaptive exploration of multidimensional data. *Proc of the 26th Intl Conference on Very Large*, pp. 307–316, 2000.

[20] S. Sarawagi, R. Agrawal, N. Megiddo, G. V. A. V. Univ Politecn Valencia, and E. T. H. Z. O. S. S. I. Edbt Fdn. Discovery-driven exploration of OLAP data cubes. *6th International Conference on Extending Database Technology (EDBT 98)*, pp. 168–182, 1998.

[21] A. Sarvghad, M. Tory, and N. Mahyar. Visualizing Dimension Coverage to Support Exploratory Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):21–30, 2017. doi: 10.1109/TVCG.2016.2598466

[22] J. Schlimmer. Mushroom data set. UCI machine learning repository, 1987.

[23] E. Segel and J. Heer. Narrative visualization: Telling stories with data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1139–1148, 2010. doi: 10.1109/TVCG.2010.179

[24] M. Vartak, S. Rahman, S. Madden, A. Parameswaran, and N. Polyzotis. SEEDB : Efficient Data-Driven Visualization Recommendations to Support Visual Analytics. 2015.

[25] E. Wall, L. M. Blaha, L. Franklin, and A. Endert. Warning, Bias May Occur: A Proposed Approach to Detecting Cognitive Bias in Interactive Visual Analytics. *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2017.

[26] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):649–658, 2016. doi: 10.1109/TVCG.2015.2467191

[27] E. Wu and S. Madden. Scorpion: Explaining Away Outliers in Aggregate Queries. *Proceedings of the VLDB Endowment*, 6(8):553–564, 2013. doi: 10.14778/2536354.2536356

[28] D. Xin, J. Han, X. Li, Z. Shao, and B. W. Wah. Computing iceberg cubes by top-down and bottom-up integration: The starcubing approach. *IEEE Transactions on Knowledge and Data Engineering*, 19(1):111–126, 2007. doi: 10.1109/TKDE.2007.250589