STORYBOARD: Navigating Through Data Slices with Hierarchical Summary of Visualizations

ABSTRACT

The task of navigating through a large, multidimensional dataset is a common challenge in exploratory analysis. Due to limitations on the number of visualizations that an analyst can examine at one time, the narrow scope of drill-downs can often lead to inductive fallacies. In this paper, we present STORYBOARD, an interactive visualization recommendation system provide safe guarantee during drill-down exploration by picking the proper visualization reference that leads to interesting and informative trends. 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. 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.

KEYWORDS

exploratory data analysis, visualization recommendation.

1 INTRODUCTION

To understand a multi-dimensional dataset, analysts often apply OLAP (Online Analytical Processing) operators to explore the space of attributes [9]. A common OLAP task includes generating visualizations to gain an overview of the data, then drilling down to interesting subsets to generate more visualizations. For example, a campaign manager may be interested in understanding the voting patterns across different demographics (say, race, gender, social class) using the 2016 US election exit polls[1]. A natural first step is to generate a bar chart for the entire population, where x-axis shows the election candidates and y-axis the percentage of votes for these candidates. He can then drill down to specific demographics of interest, say gender-based demographics by generating bar charts for female voters. In this exploration process each drill-down may lead to insights, which derive from the observed visualizations. As shown in Figure 1, an analyst can either arrive at the Black Females visualization by going through the purple or orange drill-down path. At random, an analyst that followed the purple path may be surprised at how drastically the Black Female voting behavior

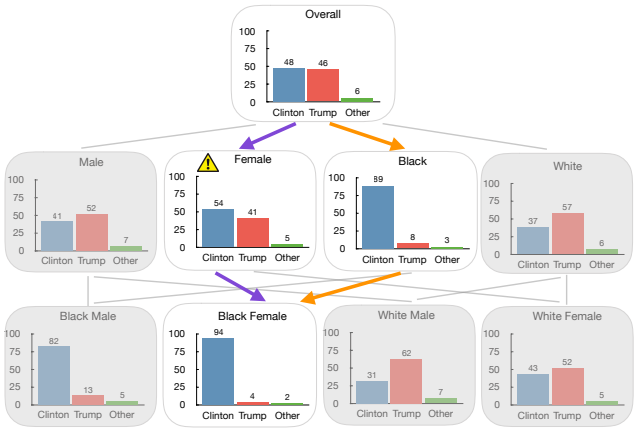

Figure 1: Example data subset lattice illustrating the misleading factor fallacy along the orange path as opposed to the informative purple path.

differs from the vote distribution for females. This behavior is no longer surprising if the analyst had went down the orange path, where the proper reference (vote distribution for Black) explains the behavior of the Black Female distribution. The misleading insight is a result of the order in which the drill-down has been performed. This example demonstrates a case of *drill-down fallacy*, which results from potentially confounding factors not explored along a drill-down path.

When an analyst explore a dataset by randomly selecting the attribute to drill down on, they may not come across the proper reference visualization that explains the behavior of the visualization of interest. Thus, they are at risk of falling prey to the drill-down fallacy. A naive solution to avoid this fallacy is to explore all potential pathways along the drill-down path. For example, generating and exploring visualizations for both race and gender based demographics, before exploring any of their combinations. Unfortunately, this approach does not scale with increasing number of factors in the drill-down path.

In this paper, we develop a tool to help users explore a dataset while avoiding improper references that may lead to drill-down fallacy. Our tool automatically identifies the best possible drill-down paths that lead to *informative insights*, and summarizes the paths. The challenge of building such tool includes considerations for how each visualization influences user's perception on other visualizations and selecting a set of visualization that are collectively interesting amongst

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

a large set of visualization. To address this challenge, we develop a notion of *informativeness*, defined as the capability of an reference visualizations to explain the visualization of interest. Informative visualizations helps users identify meaningful insights that arise from something *actually interesting* about the data (instead of confounding variables), thereby preventing users from the drill-down fallacy. Our user study result demonstrates that our tool that make use of this notion of informativeness can guide an analyst towards meaningful insights. The contribution of this paper include:

- Introducing the novel concept of *informativeness* that helps avoid drill-down fallacy in data exploration (Section 3),
- Designing a tool that automatically identifies the best possible drill-down paths based on informative insights, and summarizes those (Section 4),
- Demonstrating the efficacy of our system through a comprehensive user study evaluation (Section 5).

2 PROBLEM FORMULATION

In this section, we first describe how analysts explore the space of visualizations through drill-downs and introduce a common fallacy that arises when analysts have limited time and attention to examine all possible factors that contribute to the observed visualization. Then, we discuss how to resolve the problem of finding informative references along a drill-down path.

Research in visualization storytelling shows that people prefer hierarchically structured visualizations with increasing levels of aggregation [11, 12, 14]. In order to find meaningful insights, analysts often drill-down to explore data at different levels of granularity by adding one filter at a time. For each data subset that they encounter, they may want to visualize the data distributions through a bar chart. When analysts perform a drill-down by adding one additional filter, they naturally look towards the last visualization that they have seen (known as the 'parent') to establish what they expect to see in the current visualization (known as the 'child'). In this case, a parent is any visualization that can be obtained by removing one filter constraint from the child. For example in Figure 1, the visualizations Female and Black are the parents of the Black Female visualization.

As analysts perform drill-downs, they may be misguided by child visualizations that highly deviate from one of its parents, if one of the other potential factors that explain seemingly-anomalous behavior is overlooked (i.e. not along the chosen drill-down path). As exemplified by the exploration along the purple path in Figure 1, we refer to this phenomena as *drill-down fallacy*, since this type of fallacy arises from the inductive nature of the drill-down operation. While such fallacies can be prevented if the analyst exhaustively browses through all possible parents of any visualization that he encounters in the dataset, the prohibitively large number of visualizations and limited memory and attention of analysts make this task impractical.

Due to these challenges, our goal is to develop a mechanism that would *provide safe guarantee by picking the proper informative parent* as a reference when analysts navigate through the space of data subsets. 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. 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.

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

We regard a visualization as informative if its distance falls within a user-defined threshold θ% 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 in Figure 1, while both visualization Black and Female visualizations are considered parents of the Black Female visualization, only the Black visualization are considered the informative parent of the 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.

3 SYSTEM

Motivation and Objective

While the concept of informativeness is useful for helping analysts determine which parent is the proper informative reference for a given visualization, in practice, users often do not have a preconceived knowledge of what visualizations would lead to useful insights. The ultimate goal of exploration is to discover insights, in particular finding visualizations that are *interesting* and lead to those insights. However, without knowing *what* subset of data contains an insightful distribution, manually exploring distributions from all possible data subsets can be tedious and inefficient. In order to accelerate the process of manual drill-down, our goal is to develop a

system that automatically selects a small set of *interesting* visualizations to summarize the distributions within a dataset in an *safe and informative* manner.

We first highlight three of the challenges (*significance, safety, saliency*) that we face when building such a system and how they are each addressed in our system objective.

Safety: As discussed in Section 2, failure to select the proper reference for a given visualization can lead to the drill-down fallacy. To resolve this issue, we require that for every visualization except for the overall, at least one of its informative parents must be included within the set of selected visualizations. This enforces that every selected reference visualization is guaranteed to be informative.

Saliency: We want to select visualizations that are *visually-salient*, in other words, the visualization distribution is *interesting* if differs from the distribution of its parents. The use of distance-based metrics to quantify surprisingness or interestingness have been widely adopted in past work [7, 13, 19]. To model the interestingness of an visualization V_i in the context of its parent V_i^j, we characterize the deviation between their data distributions using a distance function $D(V_i, V_i^j)$. From the safety criteria, all parents in the dashboard are guaranteed to be informative, therefore the reference would not be misleading.

Significance: The danger of spurious patterns and correlations in visualizations that contain small subpopulation size is a well-known problem in exploratory analysis [5]. We take two preventive measures to avoid picking these misleading visualizations that are 'insignificant' in size. When constructing the visualization lattice, we allow users to select an 'iceberg condition' [2] (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 $D(V_i, V_i^j)$ between a parent V_i^j and a child visualization V_i by the ratio of their sizes $U(V_i, V_i^j) = \frac{|V_i|}{|V_i^j|} \cdot D(V_i, V_i^j)$.

Given these objectives, we select k visualization to include in our dashboard that represent a connected set of visualizations that are collectively safe, salient and significant, based on maximizing the utility $U(V_i, V_i^j)$. 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 [3] and is known to be NP-Complete, via a reduction from the CLIQUE PROBLEM [16]. Next, we discuss heuristic algorithms used for deriving a locally optimal solution for ensuring interactive runtime.

[2]The terminology is used in the discussion of iceberg cubes in OLAP literature [22].

Algorithms

We discuss algorithms used for generating the visualization lattice, and then present a high-level overview of our traversal algorithms to selecting the k-connected maximum-weighted subgraph.

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 maximum combined edge utility. Here, we discuss the *frontier greedy* algorithm which is used for generating the dashboards for our user study and defer the details of other algorithms that we have developed to the technical report.

The frontier greedy algorithm obtains a list of candidate nodes known as the *frontier* nodes, 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 present in the solution. To obtain the frontier nodes, the algorithm 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 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 amongst the frontier nodes to add to the current solution, and updates the frontier accordingly.

Algorithm 1 Frontier Greedy Algorithm

```
1: procedure PICKVISUALIZATIONS(k,lattice)
2:     dashboard ← { V_overall }
3:     while |dashboard| < k do
4:         frontier ← getFrontier(dashboard,lattice)
5:         maxNode ← getMaxUtilityNode(frontier)
6:         dashboard ← dashboard ∪ {maxNode}
       return dashboard
```

User Interaction

Given the selected visualizations, we render the dashboard visualizations in an interactive frontend interface, as shown in Figure 2. The system allows users to inspect the visualization dashboard through panning and zooming with navigation

Figure 2: a) Overview of the STORYBOARD interface for the Police Stop dataset. 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. b) User clicks on the duration=30+min visualization to request 2 additional visualization. c) A preview of the added portion of the resulting dashboard is shown.

buttons, mouse clicks, and keyboard bindings. Users can also select the x and y axes of interest, aggregation function, and optional system parameter settings to generate a dashboard.

After browsing through visualizations in the dashboard, users may be interested in getting more information about a particular node. STORYBOARD allows users to request additional summarizations based on a chosen visualization of interest as the new starting point for analysis. As shown in Figure 2a, the analyst starts with a 7-visualization dashboard on the Police Stop dataset [17]. 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, she is also interested in learning more about the other factor that contribute to high arrest rate: duration=30+min. In Figure 2b, she clicks on the corresponding visualization and requests for 2 additional visualizations. Upon seeing the updated dashboard in Figure 2c, she 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 factors such as driver's race or age. To generate the expanded dashboard, 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 motivated by the idea of *iterative view refinement* in other visual analytics system, which is essential for the users to iterate on and explore different hypotheses [10, 21].

4 USER STUDY EVALUATION

Methods

We 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 visualizations of interest?
- RQ2: How effective is our tool at helping user evaluate the importance of attributes within a given dataset?
- RQ3: How effective is our tool at guiding users towards safe and informative visualization references?

We recruited 18 participants with 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 dashboards with k=10 visualizations generated by following conditions.

STORYBOARD: The dashboards for this condition is generated by the frontier greedy algorithm (described in Section 3) and displayed in a hierarchical layout (as seen in Figure 2). 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 3,

especially since the k=10 dashboard was small enough to function without the navigation tools.

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 exhaustively inspects all possible visualization combinations. The chosen visualizations are displayed in a 5x2 table layout in the traversed order.

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[3] and display them in a 5x2 table layout. This baseline is designed to showcase a diverse set of pattern distributions within the dataset.

We randomize the ordering for each task combination to prevent confounding learning effects. The study begins with a 5 minute tutorial using dashboards generated from the Titanic dataset [1]. To prevent bias across conditions, 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 Stop datasetWe 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) and y-axis as the percentage of police stops that led to this outcome.

Participants were given some time to read through a worksheet containing descriptions of the data attributes. Then, they were given an attention check question where they are given a verbal description of the visualization filter and asked about the distributions for the corresponding visualization in the dashboard. After understanding the dataset and chart schema, participants were asked to accomplish the following tasks in the prescribed order below:

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. This task was intended to measure how well are participants at retrieving interesting visualizations (RQ1).

Attribute Ranking: Participants were given a worksheet with all the attributes listed and asked to rank the attributes in order of importance in contributing to a particular outcome (e.g. factors leading to arrest or autism diagnosis). Attribute ranking tasks are common in feature selection and other data science tasks. The goal of this task is to measure how well participants understand the relative importance of each attribute in contributing towards an outcome (RQ2).

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 considered "shallow" if 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 the result was.

Deep Prediction: This task is similar to the shallow prediction, except that the visualization to be estimated is "deep" in the sense that it has 3 filter combinations, with only one parent in the given dashboard. Both prediction tasks measure how accurate participants are at predicting an unseen visualization (RQ3).

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 diagnostic 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 study 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.

Quantitative 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 interestingness is a subjective measure, we devise a popularity-based metric that measures how interesting a visualization is amongst all participants. We assign the selection made by user j to visualization i with score δ_{ij} of 1 if a user is interested, 0 if they leave it unselected, and -1 if they are not interested. We obtain a consensus score for each visualization to measure how frequently the visualization is regarded as interesting by summing over all users' vote on that visualization.

$$\text{consensus}(V_i) = \sum_{j \in user} \delta_{ij} \tag{3}$$

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 4. Intuitively, a rating should be rewarded more if it has retrieved interesting visualization agreed by

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

many other users, likewise, ratings that does not retrieve such visualizations should be penalized more heavily.

$$\text{rating score}(V_{ij}) = \text{consensus}(V_i) \cdot \delta_{ij} \quad (4)$$

Table 1 summarizes results of rating scores averaged over the tasks that the user performed.

Dataset	STORYBOARD	Cluster	BFS
Police	1.03	0.87	1.65
Autism	3.55	3.00	1.90

Table 1: 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 BFS (p=0.003) and CLUSTER (p=0.09).

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[4], as detailed in Table 2. We see that STORYBOARD performs better than clustering in both cases. Since clustering seeks visualizations that exhibit 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 diagnostic 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. In general, our results indicate that using STORYBOARD, users gain a better understanding of attribute influence and importance.

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

Table 2: NDCG@10 scores for the attribute ranking task.

Prediction (RQ3): Since we can not directly compare between misleading and informative drill-down paths for RQ3, we use the prediction tasks as a proxy for how informative the selected visualizations in the dashboard are. The informativeness is measured by how accurate participants are at predicting an unseen visualization. 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 the first column of Figure 3, 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. The right two columns in Figure 3 also show 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 the unseen visualizations.

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

We also compute the variance of participants' predictions across the same task. In this case, low variance implies that

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

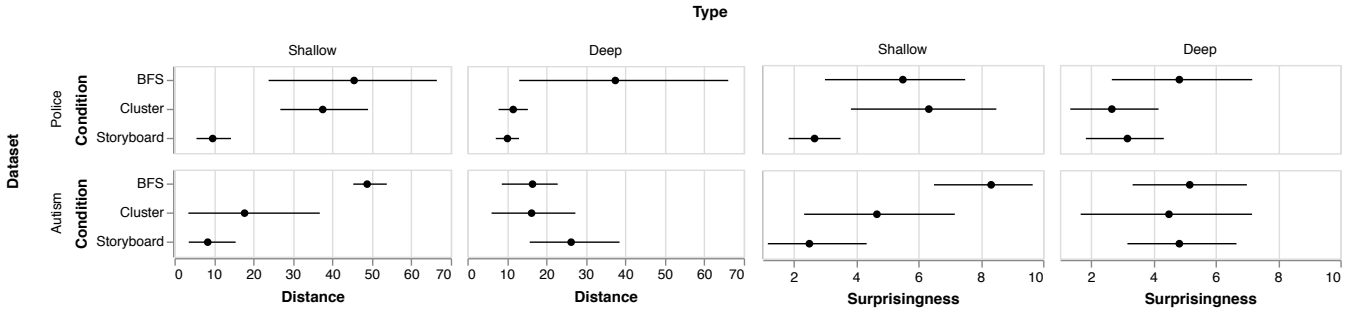

Figure 3: On the left two columns, Euclidean distance between predicted and ground truth. In general, predictions made using STORYBOARD is closer to the ground truth. On the right two columns, surprisingness rating reported by users after seeing the actual visualizations on a Likert scale of 10. In general, STORYBOARD participants had a more accurate mental model of the unseen visualization and therefore reported less surprise than compared to the baseline.

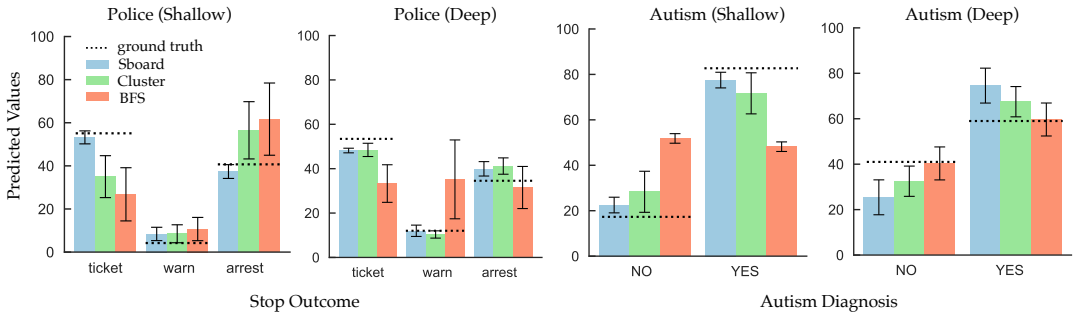

Figure 4: Mean and variance of predicted values. Predictions based on STORYBOARD exhibits lower variance (as indicated by the error bars) and great proximity to the ground truth values (dotted).

any user who reads the dashboard is able to provide consistent predictions, whereas high variance implies that the dashboard did not convey a clear data-driven story that could guide their predictions. So instead, participants relied on different priors or guessing to form their prediction. These trends can be observed in Figure 4, where the prediction variance amongst participants who used STORYBOARD is generally lower than the variance from the baselines.

5 DISCUSSION

To understand how useful are the visualizations in the recommended dashboard to analysts, 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 3, 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 many filters) at first and would want to see all the bases (univariate summaries) 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 3: Number of participants who made use of each contextual parents, summed across the two datasets. Participant behavior shows a similar trend in individual datasets. STORYBOARD participants made more comparisons in general and against parents compared to the baseline.

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 for deep predictions. 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 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 who saw table layouts 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 3.

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 similar to a dashboard that was authored with human 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 the "significance" portion of our objective, 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.

6 RELATED WORK

Our work draws from, and improves upon, past research in multidimensional data exploration, fallacies in visual analytics, and visualization storytelling.

Guided Exploration of Multidimensional Data

Given a dataset, tools such as Spotfire and Tableau supports automatic generation of visualizations (e.g. Show Me [15] by Tableau) based on perceptual graphical presentation rules. A more recent body of work automatically selects visualizations based on statistical measures, such as scagnostics and deviation. Given a scatterplot, Anand et al. [4] applies randomized permutation tests to select partitioning variables that reveals interesting small multiples using scagnostics. Given an input view (bar chart), Vartak et al. [19] finds other interesting bar charts that deviate form the input chart using a deviation-based measure. Our work extends the deviation-based measure to formulate user expectation. However, unlike the existing works, we concentrate on informativeness, which enables our systems to prevent the drill-down fallacy.

Preventing Biases and Statistical Fallacies

Visualizations are powerful representations for studying patterns in a dataset; however, cognitive biases and statistical fallacies could mislead us when it comes to interpreting those patterns [2, 20, 23]. Wall et al. [20] presents six metrics to systematically detect and quantify bias from user interactions in visual analytic systems. These metrics are based on coverage and distribution, which focus on the assessment of the process by which users sample the data space. Alipourfard et al. [2] presents a statistical method to automatically identify Simpson's paradox by comparing statistical trends in the aggregate data to those in the disaggregated subgroups. Zgraggen et al. [23] presents a method to detect Multiple Comparisons Problem (MCP) in visual analysis. In this paper, we concentrate on a novel type of fallacy during drill-down exploration that had not yet been addressed by past work.

Storytelling with Visualization Sequences

Visualizations are often arranged in a sequence to narrate a data story. Existing work on visualization sequences and storytelling have studied the structures of narrative visualizations [12, 18], effects of augmenting exploratory information visualizations with narration [6] and, more recently, ways to automate the creation of visualization sequences [11, 14]. Most of these work have adopted a linear layout (motivated by slidedecks) to present the visualization sequences. Hullman et al. [12] 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 organize visualizations in a hierarchical layout for summarizing the space of data subsets.

REFERENCES

[1] [n. d.]. Titanic: Machine Learning from Disaster. Kaggle.

[2] Nazanin Alipourfard, Peter G. Fennell, and Kristina Lerman. 2018. Can You Trust the Trend?: Discovering Simpson's Paradoxes in Social Data. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM '18)*. ACM, New York, NY, USA, 19–27. https://doi.org/10.1145/3159652.3159684

[3] Ernst Althaus, Markus Blumenstock, Alexej Disterhoft, Andreas Hildebrandt, and Markus Krupp. 2009. Algorithms for the Maximum Weight Connected k-Induced Subgraph Problem. 5573 (2009), 313–321. https://doi.org/10.1007/978-3-642-02026-1

[4] Anushka Anand and Justin Talbot. 2015. Automatic Selection of Partitioning Variables for Small Multiple Displays. 2626, c (2015). https://doi.org/10.1109/TVCG.2015.2467323

[5] Carsten Binnig and Lorenzo De Stefani. 2017. Towards Sustainable Insights or why polygamy is bad for you. *8th Biennial Conference on Innovative Data Systems Research (CIDR '17)* (2017).

[6] Jeremy Boy, Francoise Detinenne, and Jean-Daniel Fekete. 2015. Storytelling in Information Visualizations : Does it Engage Users to Explore Data? *CHI 2015* (2015), 1449–1458.

[7] Michael Correll and Jeffrey Heer. 2016. Surprise! Bayesian Weighting for De-Biasing Thematic Maps. *IEEE Transactions on Visualization and Computer Graphics* 2626, c (2016), 1–1. https://doi.org/10.1109/TVCG.2016.2598618

[8] Fadi Fayez Thabtah. 2017. Autism Screening Adult Data Set. UCI Machine Learning Repository.

[9] Jim Gray, Surajit Chaudhuri, Adam Bosworth, Andrew Layman, Don Reichart, Murali Venkatrao, Frank Pellow, and Hamid Pirahesh. 1997. Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. *Data Mining and Knowledge Discovery* 1, 1 (01 Mar 1997), 29–53. https://doi.org/10.1023/A:1009726021843

[10] Enamul Hoque, Vidya Setlur, Melanie Tory, and Isaac Dykeman. 2017. Applying Pragmatics Principles for Interaction with Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics* c (2017). https://doi.org/10.1109/TVCG.2017.2744684

[11] 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. https://doi.org/10.1109/TVCG.2013.119

[12] Jessica Hullman, Robert Kosara, and Heidi Lam. 2017. Finding a Clear Path : Structuring Strategies for Visualization Sequences. 36, 3 (2017).

[13] Laurent Itti and Pierre Baldi. 2009. Bayesian surprise attracts human attention. *Vision Research* 49, 10 (19 May 2009), 1295–1306. https://doi.org/10.1016/j.visres.2008.09.007

[14] Younghoon Kim, Kanit Wongsuphasawat, Jessica Hullman, and Jeffrey Heer. 2017. GraphScape: A Model for Automated Reasoning about Visualization Similarity and Sequencing. *Proc. of ACM CHI 2017* (2017). https://doi.org/10.1145/3025453.3025866

[15] Jock D. 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. https://doi.org/10.1109/TVCG.2007.70594

[16] Aditya G. Parameswaran, Hector Garcia-Molina, and Jeffrey D. Ullman. 2010. Evaluating, combining and generalizing recommendations with prerequisites. *Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM '10* (2010), 919. https://doi.org/10.1145/1871437.1871555

[17] E. Pierson, C. Simoiu, J. Overgoor, S. Corbett-Davies, V. Ramachandran, C. Phillips, and S. Goel. 2017. A large-scale analysis of racial disparities in police stops across the United States. https://openpolicing.stanford.edu/data/

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

[19] Manasi Vartak, Sajjadur Rahman, Samuel Madden, Aditya Parameswaran, and Neoklis Polyzotis. 2015. SEEDB : Efficient Data-Driven Visualization Recommendations to Support Visual Analytics. (2015).

[20] Emily Wall, Leslie M Blaha, Lyndsey Franklin, and Alex Endert. 2017. 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). https://www.cc.gatech.edu/{~}ewall9/media/papers/BiasVAST17.pdf

[21] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2016. Voyager: Exploratory

Analysis via Faceted Browsing of Visualization Recommendations. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (2016), 649–658. https://doi.org/10.1109/TVCG.2015.2467191

[22] Dong Xin, Jiawei Han, Xiaolei Li, Zheng Shao, and Benjamin W. Wah. 2007. Computing iceberg cubes by top-down and bottom-up integration: The starcubing approach. *IEEE Transactions on Knowledge and Data Engineering* 19, 1 (2007), 111–126. https://doi.org/10.1109/TKDE.2007.250589

[23] Emanuel Zgraggen, Zheguang Zhao, Robert Zeleznik, and Tim Kraska. 2018. Investigating the Effect of the Multiple Comparisons Problem in Visual Analysis. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 479, 12 pages. https://doi.org/10.1145/3173574.3174053