Uncovering User Interest during Data Exploration via Unsupervised Clustering

ANONYMOUS AUTHOR(S)

The emergence and increasing usage of large and dense datasets pose a challenge to both the user and visualization designers. As the visualization becomes more congested with information, it grows difficult for the user to make observations due to information overload. One solution that has garnered immense interest in recent years is to create intelligent systems that learn the user's interest and aid in data exploration. However, inferring high-level interest is still an open challenge. In this paper, we present a technique for uncovering potential data points of interest based on user interactions by learning natural groups in a given dataset via unsupervised clustering. We validate our technique's ability to discover user interest with two crowd-sourced interaction datasets. We discuss and demonstrate how to incorporate our approach into an adaptive visual system that supports users during data exploration.

ACM Reference Format:
Anonymous Author(s). 2020. Uncovering User Interest during Data Exploration via Unsupervised Clustering. 1, 1 (September 2020), 16 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The emergence and increasing usage of large and dense datasets pose a challenge to both users and visualization designers. For example, in visualizations that present individual data points (rather than showing statistics and aggregates), data density can cause overlapping, which could partially or fully obstruct some information from the user's view. Humans can only process a limited amount of information at a time, and information overload is a significant concern in visualization [16]. Although there are several methods for improving visual clutter such as filtering and sampling [23], these methods primarily focus on "big data". Naïvely applying data reduction methods to smaller datasets can remove relevant elements or exaggerate irrelevant ones.

Simultaneously, the areas of *Analytic Provenance* and adaptive systems have garnered immense attention in the visual analytics community [38]. What drives this body of work is the belief that monitoring low-level behavioral data can enable real-time prediction of high-level goals [29]. Furthermore, understanding and modeling user interactions could provide critical insights that will facilitate the design of new or improved methods to support users in data exploration [3, 10]. Although this body of work demonstrates the ability to infer and predict various behavioral patterns such as next clicks [26, 27] and bias [26, 37], there are few examples of adaptive techniques for supporting data analysis in real time [38]. The existing techniques are limited to applications such as data prefetching [3, 20], visualization recommendation [13], interface guidance [5, 6], and improving data selection [10].

Building on this body of work, we posit that we can learn a user's 'data interest' by tracking and analyzing the historical record of interactions with the data as users explore using the visualization. We hypothesize that such inference could drive applications such as filtering suggestions and other intelligent techniques to present data in a way that is easier to understand and explore.

Manuscript submitted to ACM

In this paper, we introduce a technique for uncovering data points of interest based on user interactions by learning natural groups in a given dataset via unsupervised clustering. We define *interest* as the sub-features that are most relevant to the user during data exploration. To discover the interest of users, we consider two distinct problems. First, we seek to create data models that represent beliefs about the natural groupings in the dataset that might drive a user's interest. Second, we seek to determine the set of data sub-features that are most relevant to the the user at time t, as well as the portion of the data that belongs to that subspace. We present a simple clustering and ranking approach that will uncover high-level interests of the user.

We validate our technique's ability to discover data interest using two crowd-sourced interaction datasets. Our framework was able to uncover the main exploration goals of users in both close-ended and open-ended tasks. Depending on the task, we were able to predict that future interactions were among the points in the predicted cluster 73% to 100% of the time. Finally, we suggest a potential approach to minimize information overload and aid users in their exploration. We present the proof of concept example that automatically provides filtering suggestions based on the inferred data interest. Our prototype system actively observes mouse clicks to uncover features of interest and filters the visualization (on request) to only show the data points that match the criteria. We argue that this approach has the potential to minimize information overload and accelerate data exploration.

In summary, our work offers the following contributions:

- We introduce an algorithm that discovers potential data points of interest based on user interactions by learning natural groups in a given dataset via k-means.
- We show how this technique can uncover high-level interest from low-level interest across different tasks types.
- We discuss and demonstrate how our technique could be incorporated into an adaptive visualization system that supports users during data exploration by providing filter recommendations in real-time using the model's predicted interests.

2 RELATED WORK

At a high-level, the work in this paper seeks to create a model of user interactions with a goal of providing support to users data exploration. The incorporation of such techniques in visual systems promises to increase users' ability to quickly discover patterns within the data. In this section, we discuss prior work in visual analytics that model and guide the users the during data exploration process.

2.1 Modeling the User

A variety of work in the visual analytics community which aims to model the actions and attributes of users has been proposed [38]. For example, Steichen et al. [35] and Brown et al. [4] demonstrated that it is possible to inferring user characteristics from interaction data. Work by Healey and Dennis [18] dynamically identifies and tags data elements in their visualization that are of potential interest based on the user's actions using Bayesian classification. Recent work by Ottley et al. [27] observed users' mouse interaction and succeeded in modeling the attention of users during visual data exploration with a hidden Markov model. The authors found that the user's next click for a given task was included in their model's prediction set of 100 data points (5% of the data) 95% of the time. From these findings, we can see that interaction data holds valuable information about the user.

In addition to the creation of predictive models of the user's action and attributes, there exists several work that involves building models using tracked interactions to provide support and guide the user in their visual exploration.

For example, the visualization recommendation system proposed by Gotz and Wen [13] observes user interactions during the user's task and utilizes a rule-based technique to detect meaningful interaction patterns. Their system infers the user's intended visual task from the patterns and automatically suggests an alternative visualization to the user that supports the inferred task more directly than the current visualization. Work by Dabek and Caban [8] involves a grammar-based model that learns from user interactions. Their technique builds a set of rules from patterns in the interaction data and applies these rules in form of suggestions to users in order to guide them along their visual analytic process. In this paper, we leverage interactions from users to uncover interest during data exploration.

2.2 Detecting Bias

Prior work has investigated ways to detect and quantify user bias in visual analytics systems. For example, Wall et al. [37] developed of set of metrics to mitigate the negative effects of cognitive biases by bringing attention to them throughout the analysis process . They identify six biases that are effective to detect user behavior, namely *vividness criterion, absence of evidence, oversensitivity to consistency, coping with evidence of uncertain accuracy, and persistence of impressions based on discredited evidence*. They measure bias by formulating Markov models of interaction behavior that focuses on combinations of types of interaction with objects of interaction. Gotz et al. [12] describes Adaptive Contextualization, which is an approach that monitors the user for selection bias and provides interactive tools to assess and avoid selection bias-related problems.

Most conceptually similar to the approach in this paper is recent work by Monadjemi et al. [26] which detects user bias during data analysis via *Bayesian model selection*. They defined competing models based on the features in the dataset, and used the observed interactions as model evidence. The showed that their framework can achieve not only bias detection but can predict future interactions and summarize analytic sessions.

While cognitive biases in the context of visual analytics often have a negative connotation, in a human-in-the-loop system, user bias can enable a better understanding of underlying user intentions and intuitive decision-making processes. The methods in this paper build on prior work in two ways. First, we leverage bias detection to infer interest. We aim to produce human-readable descriptions of a user's high-level data interest. Second, we present a novel approach to bias detection that utilizes off-the-shelf machine learning techniques. In this paper, we measure user interest based on the sub-features that appear the most relevant to observed interactions by applying k-means to find natural groupings within the data.

3 PROBLEM DEFINITION

We begin by introducing the theoretical foundation for our technique. We take a data-centered approach to modeling user behavior that builds on the theories proposed by Purchase et al. in [30]. We exploit the fact that *visualization is data*, and make the following assumptions:

- We assume that the visualization allows the user to perceive the features and patterns that exist in the dataset.
- We also assume that the visualization enables the user to interact with the dataset.

Thus, analyzing patterns in a dataset could reveal patterns that are observable by and from the user. To formalize our problem, we define some basic notation used in this paper. We define $\mathcal{D} = \{d_1, ..., d_N\}$ to be a dataset of size N. Each data point d_i has a finite set of M features $F = \{f_1, ..., f_M\}$. We propose a user interaction modeling technique with a generic two-step approach. The first step involves a preproccesing phase that applies an unsupervised machine learning technique to uncover natural groupings and patterns in the dataset. The most common unsupervised learning method

is cluster analysis which includes popular techniques such as k-means, Gaussian mixture models, and Hierarchical clustering. We will use the output of the algorithms to create *data models* that are organized by their characteristic set. The second step becomes a simple lookup task as we observe user interactions. Given the set of interactions D_I, we actively select a *characteristic set* $\{f_1, f_2, ...\}$ that best explains the patterns of interactions that we observe for a given user.

4 FRAMEWORK

As mentioned in Section 3, our framework works in two steps: clustering and ranking. In this section, we detail how to apply unsupervised clustering to datasets and the metrics that are used to rank the clusters and predict user interest.

4.1 Clustering

We first filter the features in dataset D that are likely to add noise during the model building phase. There are many valid techniques for performing feature selection from high-dimensional data [11], we choose a basic approach that removed low- and high-variance features, i.e. features with similar or unique values across all the samples [14]. We then create data models by performing exhaustive clusterings on the $2^d - 1$ possible feature subsets, where d is the number of remaining features in our dataset. For our clustering algorithm, we adopted k-means [17] from Python's scikit-learn library [28], but other clustering methods can be used in this framework. A potentially complicating factor is deciding the number of clusters, k. While many techniques exist for finding the "optimal" k [2, 24, 31, 33, 34, 36], we choose k to be the number of unique sub-feature value combinations found within the cluster. This is method of determining k is uncommon. However, it is justified by the fact that we are not looking for the optimal number of clusters. Having this value as our k will allow the data model to contain all the possible combinations of sub-features within the dataset. To gauge user interest, we define $\mathcal{G}$ to be a set of clusters derived from our clustering algorithm. $\mathcal{G} = \{G_1, G_2, \cdots, G_n\}$, where $n = 2^d - 1$,the number for possible feature subsets. Additionally, each G_n is a superset of clusters where $G_n = \{g_1, g_2, \cdots, g_k\}$, for a variable value of k. We also define a discrete time index t. At the start of using the visualization, $t = 0$. This index will then increment every time a participant clicks a visual element. Thus c_t is the click event at time t. Now, given a set of observed clicks, our goal is to predict the cluster that best describes the user's interactions.

4.2 Ranking

We focus on clicks as the sole indicator of interest, however, our method can be expanded to include data points from other means of interactions such as hovers, zooms, etc. Every time an interaction with a data point occurs, our algorithm takes as input the set of clicked data point clicks up to time t, $C_t = \{c_i\}_{i=1}^{t}$ and outputs a ranking of $\mathcal{G}$ that is ordered by their *relevance* score. We calculate each cluster's relevance score by summing four specific metrics that provide information about each of the groupings:

- The **percentage-in** metric represents the proportion of clicked points that appear in a cluster. See Figure 1 for a visual explanation of the metric. Percentage-in shows how important the cluster is to the user based on the overall points that have been interacted with.
- The **percentage-of** metric represents the proportion of the cluster that the clicked points make up. See Figure 1 for a visual explanation of the metric. Percentage-of shows how important the cluster is to the user based on the the points that have been interacted with within the cluster.

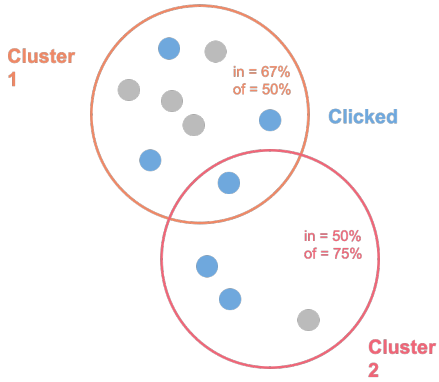


Fig. 1. Visual explanation of percentage-in and percentage-of metrics used to calculate relevance scores of clusters. In this example dataset, there are a total of eleven data points. Out of the eleven, the users has interacted with six (represented as the blue points). Cluster 1 contains a total of eight data points, where four of the points are marked as clicked. To calculate the *percentage-in* metric for Cluster 1, we divide the total number clicked points in the cluster by the overall number of clicked points to get $4/6 = 0.67$. To calculate the *percentage-of* metric for Cluster 1, we divide the total number of clicked points in the cluster by the total number of points in the cluster to get $4/8 = 0.5$.

- The **silhouette score** metric is the calculated silhouette score of the k-means model the cluster originated from. The silhouette score measures how similar a point is to its cluster compared to other clusters. This is a ratio of the Euclidean distances to the cluster centers, normalized so that a value of 1 is a perfect match to its cluster and a value of -1 is a perfect mismatch.
- The $\boldsymbol{k}$ metric is the k value used to run k-means model the cluster originated from.

Thus, $relevance = perc\_in + perc\_of + sil\_score + k$.

Predicting Data Interest. Our goal at each time stamp is to predict the user's possible data interest via cluster ranking, given the set of interactions previously observed. This means that for a given timestep t, the algorithm chooses a cluster (and its data points) with the highest relevance value given set of clicks $C_t = \{c_i\}_{i=1}^{t}$.

5 END-TO-END EXAMPLE

In this section, we consider an end-to-end example with a small dataset to demonstrate our framework from Section 4. Let D be a set of ten available apartment locations in Brooklyn, NYC which contains information about the *neighborhood*, *number of bedrooms*, *lease term*, and *pet limit* .

D = {(Greenpoint, 1, 12, 2), (Williamsburg, 1, 12, 1), (Greenpoint, 1, 15, 2), (Williamsburg, 2, 15, 2), (Williamsburg, 2, 12, 2), (Greenpoint, 1, 12, 1), (Williamsburg, 2, 12, 1), (Williamsburg, 1, 15, 1), (Greenpoint, 1, 15, 1), (Greenpoint, 2, 15, 1)}

We take the set of features F = { *neighborhood, rooms, lease term, pet limit* } and find all the possible feature combinations. A user may explore this dataset based on any combination of these features so we must construct $2^4 - 1 = 15$ models with k-means with each feature combination subset.

Assume a user first interacts with the dataset by clicking on a point. The algorithm passively observes and initiates the interaction set, C_1 ={(Greenpoint, 1, 12, 2)}. With this interaction set, we compute the relevance score for each of the clusters in our set of models, G. The top ranked cluster with the highest relevance score after this first interaction

is the cluster with sub-feature combinations, {*lease term*=12, *pet limit* =2}. The algorithm infers that the user may be interested in apartments that have a lease length of 12 and a pet limit of 2. The user proceeds to click another point and increases the interaction set to C_2 = {(Greenpoint, 1, 12, 2), Greenpoint, 1, 15, 2)}. With those two clicks, the algorithm predicts that the user is interested in apartments in Greenpoint that has a pet limit of 2. By the third interaction (C_3 = {(Greenpoint, 1, 12, 2) and Greenpoint, 1, 15, 2), (Greenpoint, 1, 12, 1)}, the algorithm notices the main sub-feature combination driving this exploration and uncovers that the user is interested in leasing an apartment in Greenpoint for 12 months.

6 MODEL VALIDATION

With our algorithm, the objective is to uncover high-level data interests by observing user interactions. We validate our model using two user study datasets the were collected and made available by Ha et al. [15], and Ottley et al. [27]. Both datasets contain proxies for interest. The Ottley et al. [27] dataset included the ground truth *closed-ended* or goal-directed tasks the their participants performed. For the Ha et al. [15] study data, their participants perform *open-ended* search tasks and were prompted to record insights from their explorations. With these two datasets, we examine how well our model uncovers latent data interest and whether it is feasible to leverage the inferences for filtering suggestions. We evaluate our model by examining the following research questions:

RQ1: *Do the top clusters match the ground-truth tasks?* By leveraging ground truth tasks and user-reported insights as proxies for interest, we examine our approach's robustness at uncovering latent interest during closed-ended and open-ended tasks.

RQ2: *Do the top clusters match future interactions?* We investigate the feasibility of real-time filtering suggestions, by inspecting how often the top-ranked cluster at time *t* contained the observation made at *t+1*.

6.1 Close-ended tasks with Map of Crimes in St. Louis, MO

To evaluate our model's ability to accurately identify features of interest with close-ended tasks, we applied our algorithm to the user study dataset collected by Ottley et al. [27].

6.1.1 St. Louis Crimes Dataset. The St. Louis Crime Dataset was pulled from the database of crimes maintained by the Metropolitan Police Department of St. Louis [25]. This dataset consists of the all reported crimes committed in the month of March 2017. There are 1,951 crime instances and eighteen attributes which consists of the unique id, longitude, latitude, street address, date, neighborhood id, crime code, and category of the reported incident. Eight types of crime exist in the database: homicide, theft-related, assault, arson, fraud, vandalism, weapons, and vagrancy.

6.1.2 Experimental Setup of Ottley et al. In their user study, Ottley et al. [27] captured mouse click data as participants interacted with a crime map visualization of St. Louis. They recruited 30 participants via Mechanical Turk who performed search tasks with an interface similar to Figure 5. Each participant completed three categories of search tasks that were designed to encourage them to either click on: (1) *Type-Based* data points with similar crime categories (e.g. Homicide, Assault, etc.), (2) *Geo-Based* locations (data points that are in the same vicinity), or (3) *Mixed* data points that are of the same crime category and in the same vicinity. Their study subjects completed two tasks for each category, resulting in a total of six tasks. The tasks presented to the subjects were:

(1) Out of all the cases of Homicide, one case differs from the other cases with regard to time. What is the time of that case?

(2) How many cases of arson occur during PM?
(3) There are four types Theft-Related crime in the red shaded region: Larceny, Burglary, Robbery and Motor Vehicle Theft. Count the number of cases of Robbery in the red shaded region.
(4) There are two types of Assault: Aggravated and Non-Aggravated assault. Count the number of Non-Aggravated Assault in the red shaded region.
(5) Count the number of crimes that occur during 7:00 AM - 12:30 PM in the red shaded region.
(6) Count the number of crimes during AM in the red shaded region.

Questions (1) and (2) were type-based tasks, (3) and (4) were geo-based tasks and (5) and (6) were mixed tasks. Consistent with Ottley et al. [27], we filtered the data to hold only the interactions of participants who successfully answered the task questions.

6.1.3 Application of Technique. Three out of the eight features remained after the preprocessing phase and feature selection: {category, district, neighborhood}. With the finalized set of features, we subset the data based on all the possible combinations of the features. There are a total of $2^3 - 1 = 7$ combinations ({category}, {district}, {neighborhood}, {category, district}, {district, neighborhood}, {category, neighborhood}, {category, district, neighborhood}). With each feature subset, we create data models via k-means. Once all the clusters were formed, we tested our framework with the respective interaction data for each task.

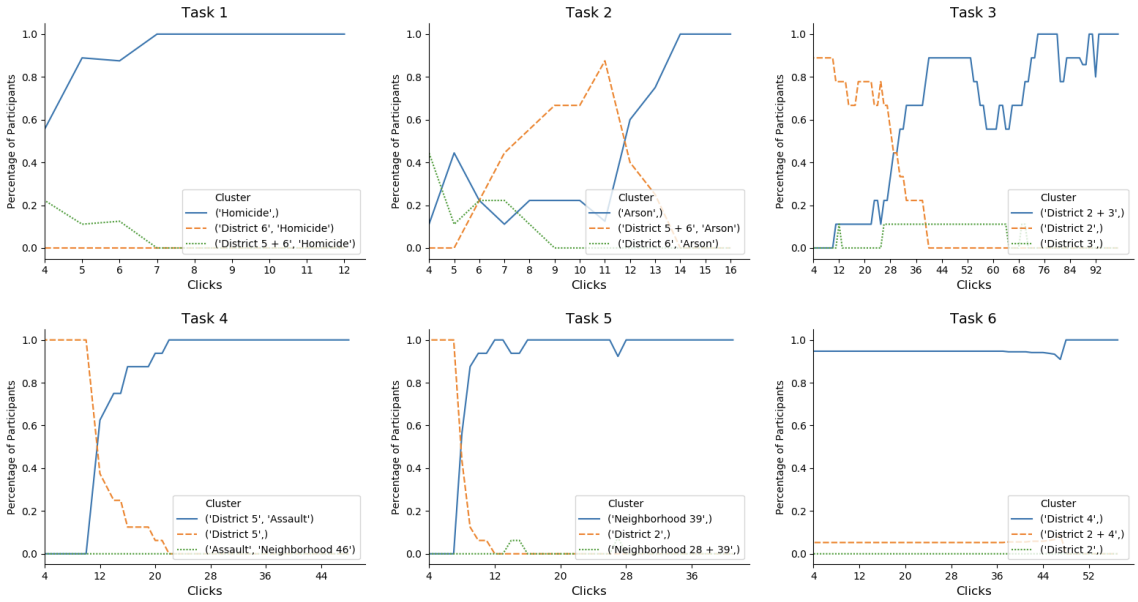


Fig. 2. The top predicted cluster's percentage among participants over time for each task from Ottley et al.[27]. For 5 out of 6 tasks, the ground-truth cluster was, on average, the top prediction. For the third task, the technique was only able to partially predict the ground-truth cluster.

6.1.4 Results. **Do the top clusters match the ground-truth tasks?** For our analysis, we sought to evaluate whether the top cluster matches the ground-truth task from the user study. The ground-truth clusters for the six tasks were:

(1) Homicide, (2) Arson, (3) District 2 & 3, Theft-Related, (4) District 5, Assault, (5) Neighborhood 39, and (6) District 4. If the top cluster at time t matches the participant's actual task, we consider this a success. We wanted adopt the most conservative definition for a task. For instance, if the goal was to "explore all homicides in Neighborhood 50", the predicted cluster is the one with the characteristic set {Homicide, Neighborhood 50}. Figure 2 summarizes our findings for each task in the dataset and shows the algorithm's accuracy as a function of time. To allow time for the algorithm to learn, we begin our predictions at $t = 4$. The line represents the overall top cluster for the tasks. For four out of six tasks, we observe that the top cluster for most participants matched the ground-truth and the technique promptly achieves a high accuracy rate for a large percentage of participants. For Tasks 2 and 3, the results were more ambiguous. On closer inspection of these two tasks, we observed that the algorithm predicted other similar clusters that may also describe the observed interactions. For example, the ground-truth characteristic set for *Task 2* was {Arson}. However, for $t = 10$, the algorithm tended to recommend a smaller characteristic set {Arson, District 5 & 6} about 80% of the time. On further inspection, it is possible that this characteristic set was indeed representative of the observations as the cases of *Arson* in the dataset were disproportionately located in Districts 5 and 6. For *Task 3*, the technique was only able to predict a portion of the ground-truth characteristic set, District 2 & 3, Theft-Related. The massively unequal amount of theft-related crimes in the dataset could have contributed to the absence of the theft-related cluster in our top prediction. Out of the 1,951 crime instances, theft-related crimes make up around 60% of the data. The users also may have clicked on crime instances in Districts 2 and Districts 3 that were not classified as theft-related.

Do the top clusters match future interactions? For a more fine grained analysis, we evaluated the quality of the algorithm's predictions by examining whether observed future interactions were among the points in the top cluster. Similar to the previous analysis, we began our predictions at $t = 4$. If the click at $t + 1$ falls within the highest ranked cluster, we consider this a success. To calculate accuracy, we divide the number of successful predictions by the total number of predictions made. Table 1 summarizes our results for all participants for each of the six tasks. We observed an overall accuracy that ranged between 73% and 100% depending on the type of tasks, resulting with an average accuracy of 90%.

Table 1. Average top cluster size and percentage of the user's next click belonging in the top cluster for the six tasks from Ottley et al. [27].

	Task	Accuracy	Size
Type-Based	1	0.8472	71.403
	2	0.7265	10.345
Mixed	3	0.9298	469.47
	4	0.9943	150.397
Geo-Based	5	0.9501	76.895
	6	1.0	414.364

6.2 Open-ended Tasks with Map of Restaurants in Toronto, ON

When introduced to a unfamiliar visualization, it is uncommon to have specified tasks/goals immediately at the start of interaction. Generally, we formulate insights and find patterns after taking some time to explore the data. So, to evaluate our model's ability to identify features of interest with open-ended tasks, we applied the algorithm to the user study dataset collected by Ha et al. [15]. The research agenda of Ha et al. was motivated by the need to improve data exploration for "small" but crowded data visualizations.

6.2.1 Yelp Open Dataset. The dataset used in the experiment was released by Yelp Inc. [39] in 2018. It contains a vast amount of information about business and reviews, however, for the purposes of the study, we reduced the dataset down to restaurants in Toronto, Ontario due to the high-density of data. Eight attributes of the restaurants were selected for use in the experiment: a unique ID, the name of the restaurant, the address of the restaurant, the exact coordinates of the restaurant, the average score of all reviews of the restaurant (ranked out of 5), the price of the restaurant (ranked

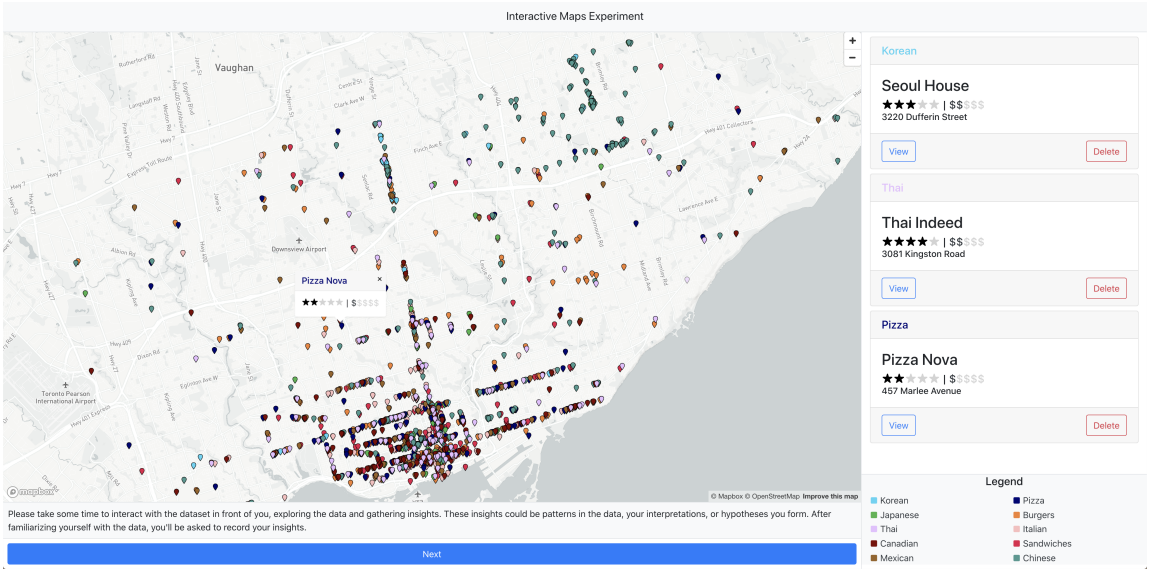


Fig. 3. The interface from the user study in Ha et al. visualizing the dense collection of restaurants on Yelp in Toronto, ON.

out of 5), and the "main cuisine" of the restaurant. The first seven attributes were included in the dataset; the eighth attribute, the "main cuisine", was hand-annotated by the authors, chosen from a set of cuisines provided in the dataset for each restaurant, and determined by independent research into the restaurant. Finally, the top ten most prevalent cuisines were chosen for use in the experiment: cuisines originating from Korea, Japan, Thailand, Canada, Mexico, Italy, and China, as well as pizza, burger, and sandwich shops. Restaurants that fell into one of these main cuisine categories were included in the final experiment, resulting in a total of 2,915 restaurants.

6.2.2 Experimental Setup of Ha et al. Prior work by Kern and Ottley [19] proposed managing overplotting by presenting a mixed-initiative information visualization system. The visual interface for this experiment is shown in Figure 3. Their system uses a hidden Markov model algorithm, developed by Ottley et al. [27], to capture and predict user attention. As a user interacts with the system, the visualization responds by adaptively re-drawing datapoints, bringing datapoints that the user is likely to be interested in to the foreground, and sending "uninteresting" datapoints to the background. Participants in the user study where participants were instructed to freely interact with the map visualization. They were also encouraged to write down as many (or as few) insights that they found about the data at any point during the experiment. The open-ended nature of this experiment was chosen to observe the users' instinctual behavior. Replicating the experimental setup by Kern and Ottley [19], Ha et al. [15] collected the interaction data of participants. Throughout the exploration process, the users were not guided by an external force/task to steer their interactions. In order to establish, general ground-truths for each session, we mapped a set of clicks to every self-reported insight written by a user and treated the features mentioned in the insights as the "ground-truth" to create multiple interaction sessions. However, the sessions were heavily filtered as Mechanical Turk data is notoriously noisy [21] and we chose to evaluate our framework with three interaction sessions.

6.2.3 Application of Technique. Three features remained the preprocessing phase and feature selection: {category, price range, stars}. With the finalized set of features, we subset the data based on all the possible combinations of the features. There are a total of $2^3 - 1 = 7$ combinations ({category}, {price range}, {stars}, {category, price range}, {price range, stars}, {category, stars}, {category, price range, stars}) and use each subset to train a data model. With each feature subset, we create data models via k-means. Once all the clusters were formed, we tested our framework with the respective interaction data for each session.

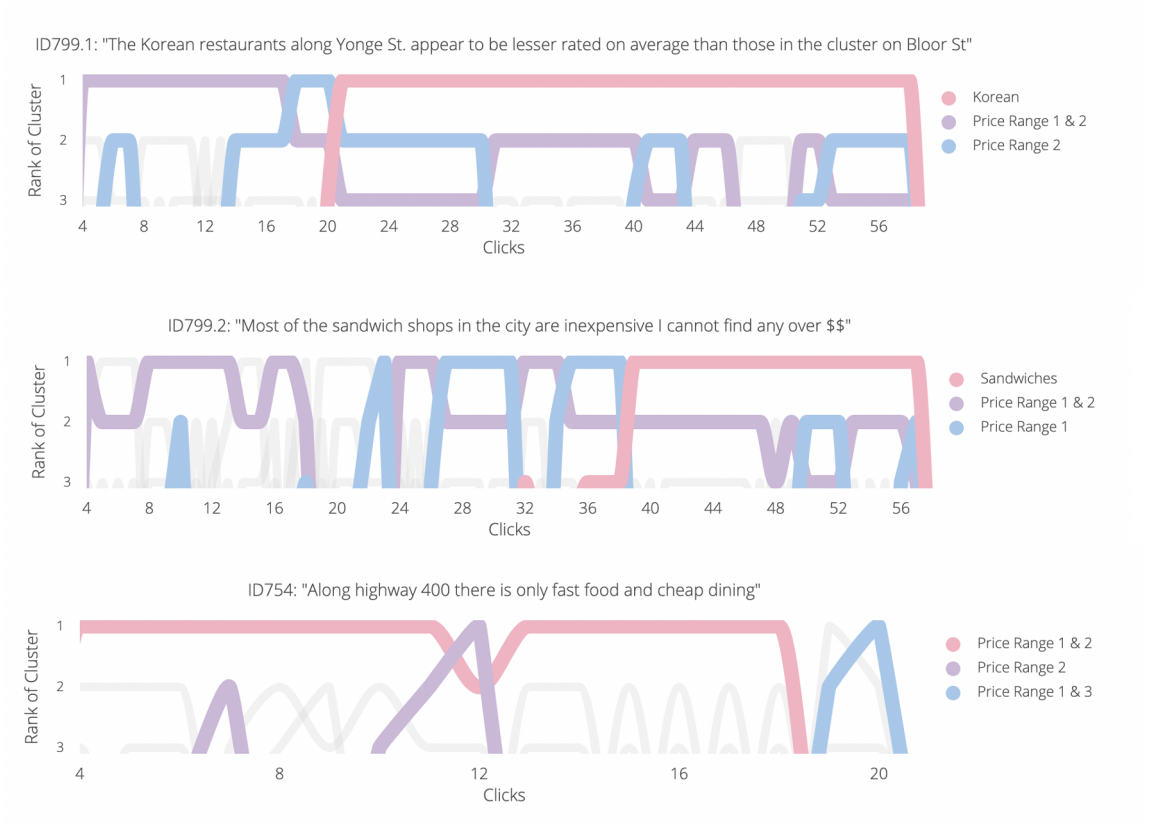


Fig. 4. Overall top three clusters predicted for each session from Ha et al. [15]. For all the interaction session we considered, the algorithm's top three predicted clusters reflect ground-truth insight. The pink line represents the overall top cluster for each session. The purple line represents the second top cluster and the blue line represents the third. The ranking of the clusters are also embedded in the order of the legend going from first to third. Other clusters that did not qualify for a place in the top three clusters are also represented in the background in gray.

6.2.4 Results. **Do the top clusters match the ground-truth tasks?** As we mentioned in Section 6.2.2, we used three insights sessions from two participants (**ID 754** and **ID 799**) to validate our framework. The insights recorded by the participants after exploration were:

ID 754: "Along highway 400 there is only fast food and cheap dining."

ID 799.1: "The Korean restaurants along Yonge St. appear to be lesser rated on average than those in the cluster on Bloor St."

ID 799.1: "Most of the sandwich shops in the city are inexpensive I can't find any over $$"

To allow time for the algorithm to learn, we begin our predictions at $t = 4$. Figure 4 summarizes our findings for each user's exploration session and shows the algorithm's overall top three ranked clusters over time. For all of the exploration sessions, we observe that the top three predicted clusters reflect the main features mentioned in the ground-truth insight. Even though a set task was not assigned to users at the start of exploration, the algorithm was able to understand and uncover the features that drove their interests.

Do the top clusters match future interactions? With the same method described in Section 6.1.4, we calculate the average percentage of the user's next click belonging in the highest ranked cluster. Table 2 summarizes our results for each user's interaction sessions. We observed an overall accuracy that ranged between 87% and 100%, resulting with an average accuracy of 90% for the open-tasked interactions.

Table 2. Average top cluster size and percentage of the user's next click belonging in the top cluster for all user sessions from Ha et al. [15].

User Session	Accuracy	Size
ID 754	0.8824	2687.764
ID 799.1	1.0	934.600
ID 799.2	0.8704	1366.759

7 EXAMPLE APPLICATION

In this section, we present how the proposed technique could be integrated in an adaptive system using a map-based visualization (see Figure 5).

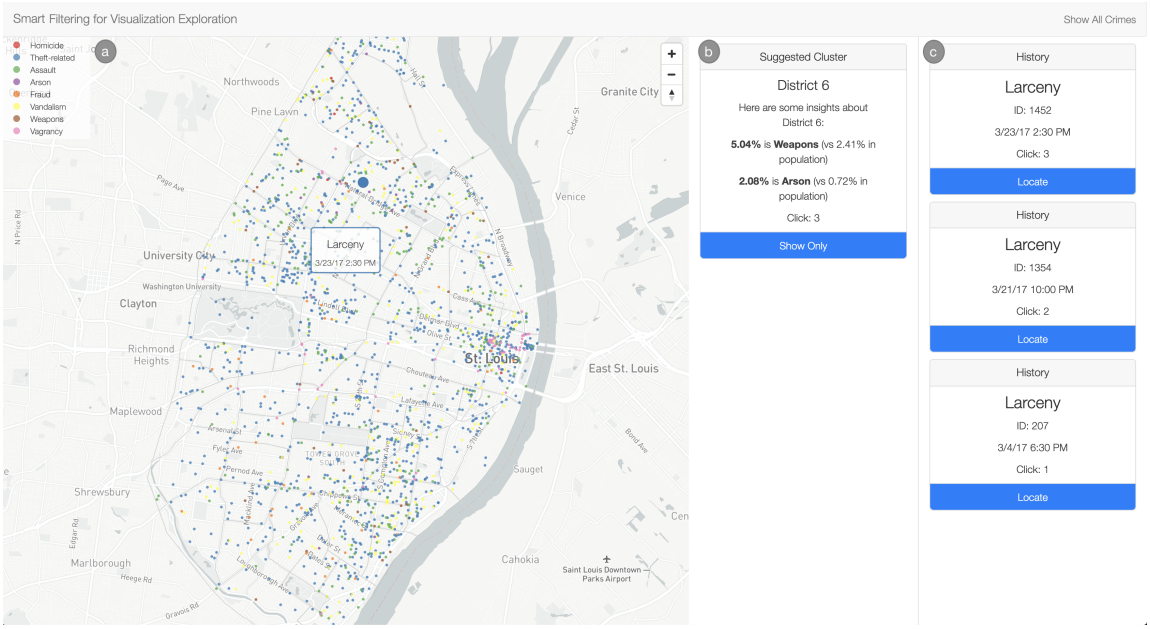
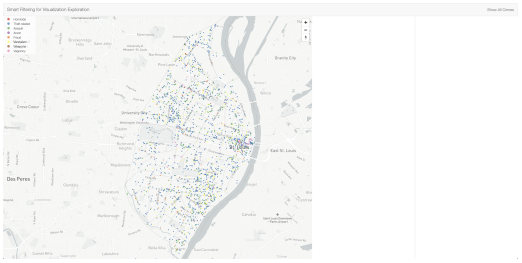


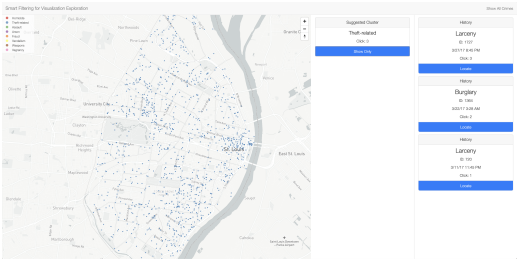
Fig. 5. The interface for our example application, Smart Filtering for Visual Exploration. (a) The map visualization of St. Louis crimes where each instance of a crime is represented by a circle. The color represents the category of the crime. (b) The cluster suggestions offered from the online filtering algorithm. The "Show Only" button filters out all data points not in the suggested cluster. The "Show All Crime" button will undo the filter selection. (c) The user's history of clicks/interactions. The "Locate" button helps the user find the specific data points from previously interaction.

7.1 The System

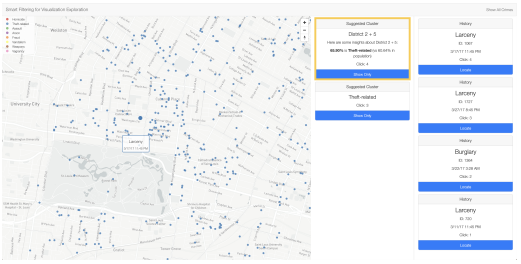
The example application utilizes the predicted interest of users from the algorithm to provide *filtering* suggestions during data exploration. Filtering is a natural way to combat information overload in a visual system [1, 32, 40]. However, there is a lack of techniques that can actively filter the data to the user's interests. This method is more intuitive than the standard filtering tools in visual systems as it does not involve the user manipulating any parameters, but it still requires the user to specifically input what they are looking for. The goal of this example application is to passively observe the user's interactions and actively create filter suggestions in real-time with minimal input from the user. On request by the user, the system will filter the visualization to only show the data points that match the criterion. Using this interface, a hypothetical user may start clicking on points of interest on the map view that represent a reported crime for a given time period. As the user interacts with data elements, the system starts modeling the user's interests. By the third click, the system actively suggests filtering options in the side bar and keeps a history of the data points that the user has interacted with. For certain filtering options, the system will provide the user statistically significant insights about the cluster. At any time during the exploration process, a user can preview the filter option by hovering over a given suggestion or remove irrelevant points by clicking "Show Only". They can also refer back to the specific points that they looked at by pressing the "Locate" button.



(a) The example application visualizing the city of St. Louis with reported crimes before Melanie starts interacting.



(b) After a couple of interactions, the system has suggested a filtering option that allows Melanie to focus on only theft-related crimes.



(c) The system provides a new filter suggestion as well as insights about theft-related crimes in districts 2 and 5 based on Melanie's interaction, highlighted by the yellow border.

Fig. 6. A depiction of our example application, Smart Filtering, throughout the usage scenario described in Section 7.2. Smart Filtering is an adaptive system that allows users to explore map-based visualizations. As the user explores the data with the system, it predicts filtering options the user may be interested in.

7.2 Usage Scenario

Melanie is interested in opening a new restaurant in St. Louis. She is thinking about the location of her restaurant and is concerned about crimes that occur in the city. With the Smart Filtering system, she observes the St. Louis crime data from the Metropolitan Police Department of St. Louis [25]. She notices that there are a lot of points on the map and is a little overwhelmed with the amount of data shown. She starts interacting with theft-related crimes first because it seems to be the most commonly occurred crime. After interacting with a couple of theft-related data points, she gets a suggestion from the system which detected

her interest in theft-related crimes (see Figure 6b). She proceeds to take the suggestion of the system and proceeds to filter out other types of crimes to explore only theft-related crimes. She is now better able compare these crimes across different neighborhoods and continues her search. Melanie heard that districts 2 and 5 have many advantages for businesses such as restaurants, so she starts looking at theft-related crimes around these locations. The system now infers that she is interested in districts 2 and 5 based on her interactions, and gives her insights on these districts. She is now aware that theft-related crimes make up 65.9% of crimes in these two districts and reconsiders where she would like to establish her restaurant (see Figure 6c).

Even though this scenario provides only a hypothetical illustrative example, the ideas generalize to other exploratory or decision making tasks. Dense visualizations can obstruct the user's exploratory process. A collaboration between humans and machines could potentially lead to facilitation of exploration and discovering patterns within the data in order to offer an approach to model users and combat information overload.

8 DISCUSSION

We began this paper by examining issues related to information overload and the promise of intelligent visualization systems that can assist the user during exploration. This paper introduced a straightforward approach for detecting data interest that leverages off-the-shelf machine learning techniques. The fundamental assumption made in this work is that we can use the natural groupings in data to predict a user's interest.

The evaluations use tasks and user-reported insights as proxies from users' data interest. Overall, our analysis demonstrated that we could use k-means clustering and ranking via the metrics proposed in Section 4.2 to infer data interest. In the closed-ended dataset, we observed that, in most cases, the cluster that we labeled as ground truth was the top predicted cluster for a large portion of our participants. However, is Tasks 2 and Tasks 3, the overall prediction accuracy was lower. This is because the evaluation of our framework is more nuanced than it appears. For simplicity, we selected a single cluster as 'ground truth,' and our analysis does not consider the clusters' relationship and that there can be multiple valid ground truth clusters. Let's consider Tasks 2 as an example. The study participants examined the Arsons in the dataset, which were also disproportionately located in District 5 and 6. An alternative analysis strategy is the consider all combinations of a ground truth set.

In the case of predicting future interactions, we demonstrated that our technique could achieve high accuracy. However, the size of the top predicted clusters were sometimes large, especially for the open-ended task. Table 1 and table 2 shows the average of cluster sizes for the top predicted cluster for each task from Section 6.1.4 and 6.2.4 respectively. Although the top clusters appear to be relevant to the participants' reported insights, the large cluster sizes might indicate that the algorithm could not narrow the user's data interest. It is indeed possible that the open-ended tasks elicited more exploration instead of exploitation of the data—however, future work in needed to disambiguate these results.

We chose not to compare our framework's next click prediction with established baseline for bias detection and future interaction prediction. The main reason for the lack of comparisons was because Ottley et al. [27] used top k points for prediction while our k is dynamic and depends on the size of the top ranked cluster. If we consider the size of the top cluster as k it should be feasible to compare our technique to Ottley et al. [27] in a more leveled playing field. Wall et al. [37] and Monadjemi et al. [26] detect biases while our method detects high-level interests. There is a growing body of work in the area of user modeling, yet it is still difficult to directly compare approaches due to the varying assumption and goals of each paper.

Clustering is a common machine learning technique, and much like selecting a subset of features, there is a myriad of techniques for performing clustering [22]. In this paper, the choice of *k*-means is convenient, off-the-shelf, but also substitutable. It is feasible to use any unsupervised clustering algorithm for the clustering step in described in Section 4.1. For example, one possibility is to apply the expectation-maximization (EM) algorithm [9] to create a probabilistic model of clustering. This fuzzy approach would allow each data element to have a variable degree of membership in each of the output clusters (data models). Furthermore, using EM could allow us to capture uncertainty in the underlying data and potentially obtain more robust results. We believe that there are other methods beyond clustering to model users' interests. We could potentially use the bias detection algorithms that were proposed by Wall et al. [37] or Monadjemi et al. [26] to infer interest.

9 LIMITATIONS AND FUTURE WORK

This paper is only an initial step towards uncovering user interactions with unsupervised clustering. In this section, we detail the limitations of our framework and suggest ideas for future improvement.

A huge limitation of our method is computational complexity. Both of the datasets we used to validate our approach contained a small number of attributes, which made uncovering user intent easy. It will be a challenge to accommodate datasets with larger dimensionality. The more features that are considered, the more data models and groupings the algorithm will need to learn. Additionally, the algorithm calculates the relevance score and sorts the clusters with every interaction. With hopes to incorporate this to adaptive systems and aid users in real-time, we must consider more sophisticated approaches to modeling the user that will retain features that are relevant to their exploration goals and tasks without the exhaustive computational cost.

There are many valid approaches in the machine learning literature for selecting features (see [7] and [14] for a survey of prior work). In our framework, we chose a low-variance method to remove irrelevant features that could add noise to the modeling step. While our ultimate goal is to remove features that could degrade the clustering process, our chosen method removed all continuous features (e.g., longitude and latitude). Conveniently, for the St. Louis dataset, categorical features such as "neighborhood" and "district" were sufficient for encoding geographical similarities. However, this may not be the case for all datasets. Under some circumstances, it might be helpful to either include all featureås in the dataset in the learning process or allow practitioners to customize the features in the data models.

Another limitation which is also applicable to many existing interactive solutions, is disregard for alternative suggestions. The system always directs user attention to the number one intent at the expense of recommending other relevant secondary interests. For example, consider a user who is looking to go to a rooftop Japanese restaurant in the fifth district that has great reviews. The user will look for his specific choice first. Based on the results, he might extend his options to the sixth district, or be open to going to a Korean restaurant. Our system is not optimized to infer search priority and secondary intents. An improvement could be to create a more sophisticated algorithm that assigns priorities to tasks based on user intent. Moreover, the metrics could be refined by comparing them to baseline decision-making behavior and prevalence of different cognitive biases to better understand user behavior.

10 CONCLUSION

In this paper, we introduce a straightforward two-step approach to uncovering user interest. We focused on developing a generalizable framework for automatically learning relationships within the data that is being visualized in order to mitigate information overload. By applying intuitive unsupervised clustering techniques like *k*-means to learn natural groupings within data and simple ranking methods, we were able to create an algorithm that passively infers user

interest through interactions. We evaluated our work using two crowd-sourced interaction datasets and demonstrated that that our technique is able to discover the main sub-features that are driving the user's exploration in both open and close-ended tasks.

REFERENCES

[1] Christopher Ahlberg, Christopher Williamson, and Ben Shneiderman. 1992. Dynamic queries for information exploration: An implementation and evaluation. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 619–626.

[2] Hirotugu Akaike. 1974. A new look at the statistical model identification. *IEEE transactions on automatic control* 19, 6 (1974), 716–723.

[3] Leilani Battle, Remco Chang, and Michael Stonebraker. 2016. Dynamic prefetching of data tiles for interactive visualization. In *Proceedings of the 2016 International Conference on Management of Data*. 1363–1375.

[4] Eli T Brown, Alvitta Ottley, Helen Zhao, Quan Lin, Richard Souvenir, Alex Endert, and Remco Chang. 2014. Finding waldo: Learning about users from their interactions. *IEEE Transactions on visualization and computer graphics* 20, 12 (2014), 1663–1672.

[5] Davide Ceneda, Theresia Gschwandtner, Thorsten May, Silvia Miksch, Hans-Jörg Schulz, Marc Streit, and Christian Tominski. 2016. Characterizing guidance in visual analytics. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2016), 111–120.

[6] Davide Ceneda, Theresia Gschwandtner, and Silvia Miksch. 2019. A review of guidance approaches in visual data analysis: A multifocal perspective. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 861–879.

[7] Girish Chandrashekar and Ferat Sahin. 2014. A survey on feature selection methods. *Computers & Electrical Engineering* 40, 1 (2014), 16–28.

[8] Filip Dabek and Jesus J Caban. 2017. A Grammar-based Approach for Modeling User Interactions and Generating Suggestions During the Data Exploration Process. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 41–50.

[9] Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* 39, 1 (1977), 1–22.

[10] Chaoran Fan and Helwig Hauser. 2018. Fast and accurate cnn-based brushing in scatterplots. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 111–120.

[11] Imola K Fodor. 2002. *A survey of dimension reduction techniques*. Technical Report. Lawrence Livermore National Lab., CA (US).

[12] David Gotz, Shun Sun, and Nan Cao. 2016. Adaptive Contextualization: Combating Bias During High-Dimensional Visualization and Data Selection. In *Proceedings of the 21st International Conference on Intelligent User Interfaces - IUI '16* (Sonoma, California, USA). ACM Press, 85–95.

[13] David Gotz and Zhen Wen. 2009. Behavior-Driven Visualization Recommendation. In *Proceedings of the 14th International Conference on Intelligent User Interfaces - IUI '09*. ACM Press, 315–324.

[14] Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *Journal of machine learning research* 3, Mar (2003), 1157–1182.

[15] Sunwoo Ha, Adam Kern, Melanie Bancilhon, and Alvitta Ottley. [n.d.]. Expectation Versus Reality: The Failed Evaluation of a Mixed-Initiative Visualization System. ([n. d.]). arXiv:2009.06019 http://arxiv.org/abs/2009.06019

[16] Graeme S. Halford, Rosemary Baker, Julie E. McCredden, and John D. Bain. [n.d.]. How Many Variables Can Humans Process? *Psychological Science* 16, 1 ([n. d.]), 70–76.

[17] John A Hartigan and Manchek A Wong. 1979. Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28, 1 (1979), 100–108.

[18] Christopher G. Healey and Brent M. Dennis. 2012. Interest Driven Navigation in Visualization. *IEEE Transactions on Visualization and Computer Graphics* 18, 10 (2012), 1744–1756.

[19] Adam Kern and Alvitta Ottley. 2020. The Effects of Mixed-Initiative Visualization Systems on Exploratory Data Analysis. *Journal of machine learning research* (2020). https://openscholarship.wustl.edu/eng_etds/523/

[20] Meraj Ahmed Khan and Arnab Nandi. 2019. Flux capacitors for JavaScript deloreans: approximate caching for physics-based data interaction. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*. 177–185.

[21] Robert Kosara and Caroline Ziemkiewicz. 2010. Do Mechanical Turks dream of square pie charts?. In *Proceedings of the 3rd BELIV'10 Workshop: Beyond time and errors: Novel evaluation methods for information visualization*. 63–70.

[22] Martin HC Law, Mario AT Figueiredo, and Anil K Jain. 2004. Simultaneous feature selection and clustering using mixture models. *IEEE transactions on pattern analysis and machine intelligence* 26, 9 (2004), 1154–1166.

[23] Zhicheng Liu, Biye Jiang, and Jeffrey Heer. 2013. imMens: Real-time visual querying of big data. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 421–430.

[24] Geoffrey J McLachlan and Kaye E Basford. 1988. *Mixture models: Inference and applications to clustering*. Vol. 84. Marcel Dekker.

[25] Metropolitan Police Department of St. Louis. 2017. St. Louis Crime Data. http://www.slmpd.org/crime{_}mapping.shtml

[26] Shayan Monadjemi, Roman Garnett, and Alvitta Ottley. [n.d.]. Competing Models: Inferring Exploration Patterns and Information Relevance via Bayesian Model Selection. ([n. d.]). arXiv:2009.06042 http://arxiv.org/abs/2009.06042

[27] Alvitta Ottley, Roman Garnett, and Ran Wan. 2019. Follow The Clicks: Learning and Anticipating Mouse Interactions During Exploratory Data Analysis. *Computer Graphics Forum* 38, 3 (2019), 41–52.

[28] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research* 12, Oct (2011), 2825–2830.

[29] William A Pike, John Stasko, Remco Chang, and Theresa A O'connell. 2009. The science of interaction. *Information visualization* 8, 4 (2009), 263–274.

[30] Helen C Purchase, Natalia Andrienko, Thomas J Jankun-Kelly, and Matthew Ward. 2008. Theoretical foundations of information visualization. In *Information Visualization*. Springer, 46–64.

[31] Jorma Rissanen. 1983. A universal prior for integers and estimation by minimum description length. *The Annals of statistics* (1983), 416–431.

[32] Manojit Sarkar and Marc H Brown. 1994. Graphical fisheye views. *Commun. ACM* 37, 12 (1994), 73–83.

[33] Gideon Schwarz et al. 1978. Estimating the dimension of a model. *The annals of statistics* 6, 2 (1978), 461–464.

[34] Padhraic Smyth. 1996. Clustering Using Monte Carlo Cross-Validation.. In *Kdd*, Vol. 1. 26–133.

[35] Ben Steichen, Giuseppe Carenini, and Cristina Conati. 2013. User-Adaptive Information Visualization - Using Eye Gaze Data to Infer Visualization Tasks and User Cognitive Abilities. In *Proceedings of the 18th International Conference on Intelligent User Interfaces - IUI '13*. ACM Press, 317—-328.

[36] John Stutz and Peter Cheeseman. 1996. AutoClass-a Bayesian approach to classification. *FUNDAMENTAL THEORIES OF PHYSICS* 70 (1996), 117–126.

[37] E. Wall, L. M. Blaha, L. Franklin, and A. Endert. 2017. Warning, Bias May Occur: A Proposed Approach to Detecting Cognitive Bias in Interactive Visual Analytics. In *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*. 104–115.

[38] Kai Xu, Alvitta Ottley, Conny Walchshofer, Marc Streit, Remco Chang, and John Wenskovitch. 2020. Survey on the analysis of user interactions and visualization provenance. *STAR* 39, 3 (2020).

[39] Yelp. 2018. Yelp Open Dataset. yelp.com/dataset

[40] Ji Soo Yi, Youn ah Kang, and John Stasko. 2007. Toward a deeper understanding of the role of interaction in information visualization. *IEEE transactions on visualization and computer graphics* 13, 6 (2007), 1224–1231.