Capturing User Intent when Brushing in Scatterplots

Kiran Gadhave
University of Utah

Jochen Görtler
University of Konstanz

Zach Cutler
University of utah

Carolina Nobre
Hardvard University

Oliver Deussen
University of Konstanz

Miriah Meyer
University of Utah

Jeff Phillips
University of Utah

Alexander Lex
University of Utah
alex@sci.utah.edu

Abstract

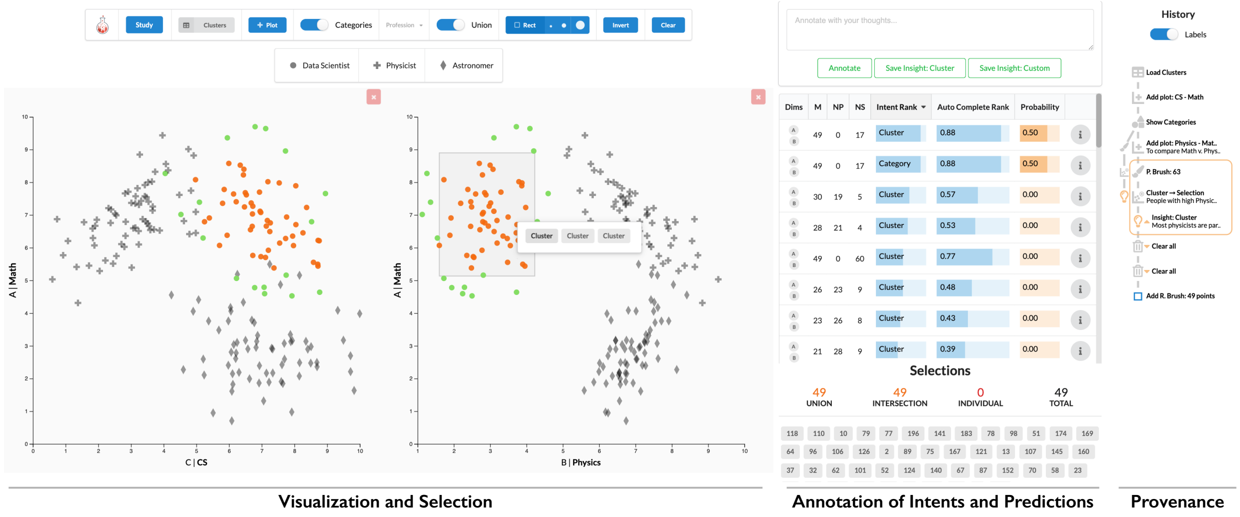
Being able to capture or predict a user's intent behind a brush in a visualization tool has important implications in two scenarios. First, predicting intents can be used to auto-complete a partial selection in a mixed-initiative approach, with potential benefits to selection speed, correctness, and confidence. Second, capturing the intent of a selection can be used to improve recall, reproducibility, and even re-use. Augmenting provenance logs with semi-automatically captured intents makes it possible to save the reasoning behind selections. In this paper, we introduce a method to infer intent for selections and brushes in scatterplots. We first introduce a taxonomy of types of patterns that users might specify, which we elicited in a formative study conducted with professional data analysts and scientists. Based on this, we identify algorithms that can classify these patterns, and introduce various approaches to score the match of each pattern to an analyst's selection of items. We introduce a system that implements these methods for scatterplots and ranks alternative patterns against each other. Analysts then can use these predictions to auto-complete partial selections, and to conveniently capture their intent and provide annotations, thus making a concise representation of that intent available to be stored as provenance data. We evaluate our approach using interviews with domain experts and in a quantitative crowd-sourced study, in which we show that using auto-complete leads to improved selection accuracy for most types of patterns.

1 Introduction

When experts interact with a visual analysis system, they are frequently guided by a domain-specific analysis goal, such as identifying a gene that could be a drug target. In service of answering this question, they execute a series of tasks, such as selecting a set of correlated items for detailed analysis. In contrast to the high-level goal of answering a domain-specific question, these intermediate tasks are often based on patterns in the data: for example, selecting outliers, clusters, or items that correlate. Such a carefully constructed selection of items based on a higher level but domain-agnostic structure reflects a reasoning process, an intent by the analyst. We refer to the motivation behind these actions as *mid-level intents* of an analyst. Mid-level intents are distinct from higher-level intents in that they are free of context and based solely on the data. They are also distinct from low-level intents, such as hovering over an item to read its label. In this paper, we introduce methods to infer these mid-level intents for brushes in scatterplots.

Why is capturing mid-level intents important? First, inferring intents based on partial selections can be used to **auto-complete** selections. To select outliers, for example, analysts would only have to brush a few examples and could then auto-complete the selection, instead of painstakingly brushing them individually. Auto-complete can also be used to correct a selection. For example, if an analyst intended to select a cluster, reviewing the predicted cluster might reveal points that should be added to the selection.

Second, making mid-level intents available in provenance data improves **recall and reproducibility** of analysis processes conducted with visualization tools. By capturing such processes at a higher abstraction than just low-level interactions, they become more transparent when revisited either by the original analyst or a collaborator. Hence, analysis sessions that capture intents are more justifiable and likely to increase trust in the process. Down the line, such rich provenance data also has the potential to enable **re-using visual analysis sessions** on modified or updated data. For example, when an analyst first removes outliers



Dims	M	NP	NS	Intent Rank	Auto Complete Rank	Probability
A B	49	0	17	Cluster	0.88	0.50
A B	49	0	17	Category	0.88	0.50
A B	30	19	5	Cluster	0.57	0.00
A B	28	21	4	Cluster	0.53	0.00
A B	49	0	60	Cluster	0.77	0.00
A B	26	23	9	Cluster	0.48	0.00
A B	23	26	8	Cluster	0.43	0.00
A B	21	28	9	Cluster	0.39	0.00

Selections

49 UNION
49 INTERSECTION
0 INDIVIDUAL
49 TOTAL

118 110 10 79 77 196 141 183 78 98 51 174 169
64 96 106 126 2 89 75 167 121 13 107 145 160
37 32 62 101 52 124 140 67 87 152 70 58 23

Load Clusters
Add plot: CS - Math
Show Categories
Add plot: Physics - Mat..
To compare Math v. Phys..
P. Brush: 63
Cluster → Selection
People with high Physic..
Insight: Cluster
Most physicists are par..
Clear all
Clear all
Add R. Brush: 49 points

Figure 1: Scatterplots showing three dimensions of a dataset. An analyst has brushed points in the right scatterplot based on a pattern they see (orange points). Our system predicts possible intents that are ranked by their match to the current selection. The points in green show a cluster that is recommended by our system based on the selection. When an analyst accepts this suggestion, a semantically rich log entry is stored in a provenance graph, shown on the right. [Interactive Figure]

before proceeding with an analysis, that action could be translated into a rule, which then could be used to automatically remove outliers from an updated dataset.

Our primary contribution is **a set of methods to detect and capture these mid-level intents** for brushes and selections. Even though we believe that our approach generalizes broadly, we use scatterplots and tabular data as common and important representatives of visualization techniques and data types, respectively. To identify the types of patterns that map to these mid-level intents, we conducted a formative interview study with scientists that regularly use scatterplots in their research. We then select data mining algorithms that are suitable to detect these patterns in a dataset and introduce methods to address the potentially large space of dimensions and parameters. We develop three approaches to score the output of the algorithms relative to an analyst's selections.

Our secondary contribution is **an implementation of these methods in an interactive visualization technique and leveraging them for auto-complete and provenance tracking**. By presenting ranked predictions of patterns for a selection, we create a mixed-initiative approach that lets analysts easily capture their mid-level intent by verifying a prediction. We provide the means to annotate these mid-level intents to tie them to higher-level domain goals and capture all of this information in a provenance graph.

To validate that we can successfully predict mid-level intents, we conduct a large, crowd-sourced quantitative study testing the correctness of the auto-completion of patterns in an analysis process.

We demonstrate the usefulness of our approach in a set of examples and validate our system using feedback sessions with domain scientists while analyzing their own data.

2 Related Work

Our work is related to predicting intents in different contexts, data-aware brushes and selections, provenance tracking, and annotation of visual analysis processes, which we discuss in the following.

2.1 Predicting User Intents

Inferring user intent has been studied in various contexts. For example, Myers [33] proposes methods for inferring operations and source code from demonstrations when implementing graphical user interfaces. More specific to data analysis, Gotz and Zhou [21] study analysts' activities and model them in four tiers, from high level *tasks*, to *sub-tasks*, to *actions*, to *events*. This classification is the inspiration for our categorization into high-level, mid-level and low-level intents. Actions, which correspond to our mid-level intent, are composed of a type, an intent, and parameters. They represent an executable, semantic step, such as a query, that bridges the high-level human cognitive ability and the low level user interactions. Gotz and Zhou implement this framework in a prototype, HARVEST, that captures such actions. In contrast to our work, however, Harvest captures *that* an action was taken, but not *why*. A related tool that also captures actions is SensePath [34]. A key difference to Gotz and Zhou's work is that SensePath is optimized to support qualitative data analysis, i.e., it is made for analysts to use the log of semantic actions in qualitative coding.

Dou et al. [10] argue that much of the reasoning process during a visual analysis session can be inferred by humans from inspecting user interactions, yet it is unclear whether this can be leveraged by automatic methods [36]. Brown et al. [4] have shown that user performance and certain personality traits can also be inferred from analyzing user interactions. A thread of work is concerned with predicting future events in an analysis process, for example to enable guidance [5]. Ottley et al. [37], for example, predict future clicks on items based on an interaction history. Steichen et al. [45] and Gingerich and Conati [18] show that it is possible to predict lower level tasks, such as *retrieve value* from eye gaze data. This differs from our goal of predicting the intent of a current selection.

A common goal for intent prediction is view specification, i.e., the selection of data (sub)sets and suitable visual encodings. Systems such as Tableau's Show Me [30] use data properties to predict useful visual encodings. Natural language interfaces for view specification attempt to extract user intents from language [49] and extract configurations for a view. Saket et al. [42] predict user intents for view specification from demonstrations, such as assigning a color to a dot in a scatterplot, based on which their system infers the intent of mapping an additional variable in a dataset. Their follow-up work [41] demonstrates that users seamlessly switch between manual and mixed-initiative approaches. All of these approaches are similar in the sense that they attempt to predict intent, but none of them is concerned with predicting intents of user selections based on patterns in data.

2.2 Data-Aware Brushes and Selections

Selections, and the related concepts of brushes, queries, and filters specify a subset of data items. Most selections are defined either by explicit clicks on individual items, "paint-brushes" that select all elements under a brush tool, geometric brushes, such as rectangles or lassos, or textual queries. More advanced, data-driven brushes have also been proposed. For example, Fan and Hauser [14] introduce a method for fast brushing based on neural networks, where they estimate an intended selection based on simple sketches. While they do not predict intents based on these brushes, a method like theirs could be used to improve brushing in our system.

Data-aware selections are actions that are defined in *data space* [9, 26, 31]. For a selection, for example, this means that it is described by conditions, not by a list of items. Dynamic queries [43] are commonly realized in a data-aware way: all items that fit certain conditions, defined e.g., via sliders, are considered to be in the query results. Certain types of brushes [2] can be realized in a data-aware way. A rectangular brush in a scatterplot, for example, easily translates into the necessary conditions. Many selections (and other actions) are, however, realized by direct reference, e.g., by pointing at items, hence they are defined in *item space*. Actions that are defined in item space have several disadvantages: they cannot be generalized to apply to updating data, and they cannot be used to semantically explain a selection. Data-aware actions, in contrast, are robust to changes, can be used to explain and justify an action, and can be used in various ways to support an analyst, e.g., by relaxing a selection [24], or for re-use in a different context [48].

Most data-aware selections are realized by deriving rules directly from a brush. In more general cases, rules for data-aware selections are harder to derive. It is possible, however, to derive the pattern of a selection (what makes the item in a selection belong to each other and different from everything else) algorithmically. Xiao et al., for example create "knowledge representations" of selections in communication networks [51]. This is similar in spirit to our work, yet, Xiao et al's knowledge representations are limited to simple clauses and are not concerned with higher-level patterns in the data.

2.3 Provenance

Provenance in the context of data analysis refers to the history of an artifact, such as a dataset, a computational workflow, or an insight. Ragan et al. [40] discuss different purposes of provenance, including recall, replication (reproducibility), presentation, and collaboration (among others), but do not discuss re-use. Ragan et al. also characterize the different types and purposes of provenance. They distinguish between the provenance of data, the provenance of visualization, the provenance of interaction, the provenance of insights (which captures analytical findings), and the provenance of rationales (which captures the reasoning behind any decisions made). Most provenance tracking techniques are limited to the former three, while insight and rational provenance can currently only be achieved using manual annotation.

With regards to provenance tracking, there are two distinct approaches: (1) tracking the history of an analysis to achieve provenance (process-based), and explicitly modeling a visualization workflow (workflow-based) [15].

Workflow-based approaches are common in large scale scientific data processing [8] in systems such as Galaxy [19] or SCIRun [38]. Workflow approaches are also common for specifying the visualization pipeline, for example for volumetric data [1], networks [12], and tabular data [52]. A benefit of workflow-based systems is that they explicitly capture rules and thus can be reused easily. However, even these rules do not typically capture higher-level semantics or intents.

Process-based approaches are the alternative to explicitly modeling workflows. They provide analysts with a interactive visualization systems while tracking the analysis process in the background [36]. Many visualization systems support the tracking of a history for the purpose of action recovery (undo/redo), so we limit our discussion to systems that explicitly target provenance. Examples include the graphical histories by Heer et al. [25] or CzSaw [27]; both render prior states as thumbnails. There are also various tools that represent histories as node-link diagrams [22,28,44,47], as well as methods to automatically detect key states in an analysis process [16], or retrieve prior states based using search [46]. However, in all of these cases the tracked information is based on interaction logs and lacks any higher-level semantics.

2.4 Annotation

One approach to capture user intents and semantics is through **note taking and annotation**. Annotations are common in visualizations designed for presentation, but are not frequently integrated in exploratory visualization tools, with notable exceptions (e.g., [13,22,23,25,29,50]). Some systems, such as GeoTime [13], enable complex annotations and also provide contextual information. Click2Annotate [6] provides templates to quickly annotate predefined, common patterns. InsideInsights [32] enables analysts to manually document insights hierarchically and link them to a provenance history. While manual notes, documentation, and annotations can capture analysts' reasoning and insights, creating and maintaining them is associated with a burden on the analyst and thus a lack of scalability [21]. Furthermore, user-generated annotations are not readable by machines and hence cannot be leveraged for re-using analysts processes. Hence, in this paper, we will use semi-automatic, data-aware insight characterization, combined with manual annotations to capture intent and reasoning and enable re-use of analysis processes.

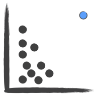
3 A Taxonomy of Patterns for Selecting Data Points

When analyzing data, users have intentions at different levels of abstraction. We are specifically interested in the aforementioned mid-level intents behind brushes or selections of data items in scatterplots, which are still semantically rich but domain agnostic [20].

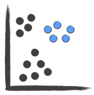
To define a taxonomy of patterns that map to these intents, we first developed an initial classification based on the literature and our own experiences working with scatterplots. We then validated and extended the initial classification through a formative study that observed what types of patterns are commonly used in data analysis workflows. The goal of the study was to identify types of patterns experts care about when analyzing scatterplots, independent of a specific system or a selection modality. We conducted the study with six domain experts at the University of Utah that regularly use scatterplots in their data analysis, recruited using a convenience sample of domain experts we had interacted with professionally, and their research groups. The study was reviewed by the institutional review board and classified as exempt from full review. Our inclusion criteria were: (1) regular use of scatterplots, and (2) and a willingness to share scatterplots or data used in scatterplots. We identified six participants from nursing, astrophysics, chemical engineering, psychiatry & population health, and surgery. The participants included one graduate student, one research scientist, and four faculty members.

In the study we provided the participants with paper printouts of scatterplots of their own data and asked them to describe and highlight (using a pen) the kinds of patterns they find interesting. One participant was not available for an in-person meeting but used a pen and touch interface to mark up his scatterplots. The interviews were video recorded and then transcribed. The transcriptions were coded by two independent coders using a seeded codebook developed from the initial classification of patterns: outliers, clusters, categories, multivariate optimization, and range queries. The coders did not fully agree on the number of codes and their type for each interview. A table in the supplementary material shows the code frequencies from both coders for each interview. Coder 1 coded 75 segments and coder 2 coded 79 segments. Our main intention of the interviews, however, was to verify and potentially extend our set of patterns. Both coders identified many instances of outliers, clusters, categories, and range queries. Only one of the two coders identified two cases of multivariate optimization. However, both coders frequently identified correlation analysis, which we originally had not included in our set of patterns. Based on this process, we identified the following data patterns that match user intents when analyzing data in scatterplots.

Correlations. Correlations are associations between two or multiple dimensions. They were mentioned as a target pattern in five of our six interviews with domain experts. Frequently, analysts were looking to identify correlations in the overall datasets or parts of the data, but also find points that do not fit the correlations. They had the intent to **identify subsets of data that correlate**, but also **identify items that do not fit the correlation**. In several interviews, these were identified as "bad data". We found that participants did an approximate visual regression analysis, identifying both linear and non-linear trends.

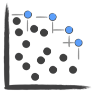


Outliers and Inliers. Outliers are data points that differ significantly from other items. They were brought up as a pattern of interest in all six interviews. Frequently, analysts wanted to understand what causes the data points to be outliers, relying on their background knowledge. Outliers are also related to, but distinct from the aforementioned points that do not fit a correlation: for example, an item can be an outlier in its magnitude but perfectly fit the correlation. Outliers were also mentioned as bad data that should be filtered out. We consider both outliers and "inliers", i.e., the set of points that are not outliers, as target patterns.



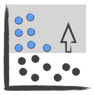
Clusters and Groups. Clusters or groups of data points are items that are similar to each other, but distinct from the rest of the dataset.

They were mentioned as a pattern that analysts look for in three out of six interviews. Clusters were frequently not well-defined in the data they analyzed.



Multivariate Optimization. One goal when analyzing data is to find data points that are dominant over multiple dimensions. A typical example is to find a hotel that is both, close to the city center and affordable. The set of such points is often called a skyline [3]. Hotels in the skyline are such that no other hotel is both cheaper and closer to the center. Skylines were brought up in two of our six interviews, and hence are the least frequently mentioned intent.

Categories. An observed pattern can sometimes be traced back to the items being of distinct categories. Four out of our six expert participants mentioned that they intend to select elements by category. For example, one expert wanted to separate the experiment from controls.



Ranges. Four of the six experts mentioned that they select data based on numerical ranges. Several experts mentioned that these ranges can be based on domain conventions for setting cut-offs. We observed range selections based on single or multiple dimensions, implying that ranges can be combined for more complicated queries.

Discussion. While we believe that the described patterns cover a broad range of use cases, we do not argue that or list of patterns is exhaustive. For example, there might be domain-specific patterns that are meaningful in certain contexts, which we did not cover. Also, analysts might use combinations of these patterns, for example, identifying the skyline of a category of values (e.g., the skyline of hotels, ignoring bed and breakfasts). Also, our pattern classification is valid only for tabular data in scatterplots. We expect that other patterns, such as rankings, would be common in different representations. Finally, we have sometimes included a pattern and its anti-pattern, such as outliers and non-outliers as separate patterns, but have not done so consistently for all patterns. We have included anti-patterns for those cases where they were explicitly mentioned in our interviews (outliers and correlation). However, anti-patterns could also be considered for other cases.

4 Mapping Patterns to Intents

Most patterns that we identified in our formative study are also commonly targeted in data mining. This implies that various algorithms can be used to identify them. We leverage this diversity to calculate a broad set of patterns using different algorithms, combinations of dimensions, and parameters. We then compare the computed patterns with user selections and rank them according to that match. While our initial step creates a large set of patterns, the subsequent ranking makes these predictions manageable. We explain the details of the algorithms used and our ranking approaches in this section.

Up to this point, we have implicitly assumed that the patterns we discussed appear in two-dimensional space. In practice, however, many datasets have much higher dimensionality. Hence, a key question we have to answer is: For which dimensions should we calculate predictions? We considered calculating patterns for:

1. all pairs of dimensions,
2. all dimensions that are actively brushed in the system,
3. all dimensions that are visible in the system,
4. all dimensions in the dataset, and
5. any combination of these options.

Calculating all possible options is computationally expensive, if not prohibitive, but also not necessary. As we aim to predict the intent of analysts interacting with (possibly multiple) 2D scatterplots, and not to reveal high-dimensional patterns, we decided to limit predictions to (1) pairwise dimensions and (2) the dimensions that are actively brushed. We believe that predicting patterns on pairs of dimensions is the most appropriate choice for 2D scatterplots, as these patterns match what is visible in the plot. This is also supported by the fact that our formative interviews did not surface an instance of higher-dimensional selections beyond ranges. However, we also do not want to exclude the possibility of analysts selecting higher-dimensional patterns. Hence we also calculate all patterns for all dimensions that are actively brushed, as the brushes indicate that an analyst is explicitly interested in a combination of these dimensions. Consequently, in a set of two 2D scatterplots visualizing dimensions A/B and C/D, and with active selections in both scatterplots, we would calculate and predict patterns in two dimensions for A-B and C-D; and patterns in 4D space for A-B-C-D.

4.1 Algorithms

Many algorithms can extract the patterns we describe. In our system, we deliberately rely on standard algorithms that are robust and simple, although more sophisticated versions might exist. One reason for this is generality: Many data mining algorithms require careful choices of hyperparameters, but choosing good parameters requires expertise and trial-and-error, which is not acceptable for our use case. Instead, we choose parameters for these simple algorithms by sampling the parameter space or rely on defaults. For example, we run k-means with a k of 2-7, but use defaults for all other parameters. We do not use evaluation approaches for the quality of the outputs; instead, we let our ranking approach reveal the most suitable results. We also assume that the visualization uses linear scales. However, an extension to logarithmic or power scales would be straightforward. We use algorithms provided by *Scikit-learn* [39] unless noted otherwise, and normalize the data before the analysis.

For **outliers**, we use the *local outlier factor*, in addition to the outliers that are a by-product of the clustering algorithm *DBSCAN*. We also treat inliers provided by the local outlier factory as a separate prediction.

Multivariate optimization is used to find values that are optimal across multiple dimensions. While a general optimization would require weighting the value of each dimension, **Skylines** are a generic approach that determines the items that are not dominated by other points. We implemented a version of the algorithm proposed by Borzsony et al. [3]. As a skyline requires a definition of what is considered "good" in each dimension (e.g., a *low* price, but a *high* customer rating is considered good for a hotel), we compute skylines for all high/low permutations of the 2D cases. We limit predictions to all-low or all-high for higher-dimensional cases, as calculating all possible permutations would be computationally expensive.

As **clusters** are frequently a matter of interpretation and different algorithms pick up on different types of clusters, we use two different formulations that have complementary strengths: *DBSCAN* and *k-means*. DBSCAN is based on a (parameterized) measure of density (clusters are clouds of dense points of arbitrary shape), whereas k-means assumes roughly spherical clusters, and requires the cluster number as a parameter. If no clusters are present, DBSCAN considers the whole dataset as one cluster (except for outliers), while k-means always provides a segmentation of the dataset. We solve each formulation multiple times with different parameterizations.

The patterns that we have described so far use the output of an algorithm as a reference against which we can compare an analyst's selection. The **range-based query** pattern differs from the other patterns in that we do not have such a reference. This is because the values used in range-selections are typically external to a dataset. Our interviews have shown that ranges can be the result of domain-specific knowledge, or can be used to select all the high or low values. We also found that analysts create complex queries by combining multiple simple brushes and selections. The traditional approach to storing range-based queries is to store the extent of brushes. However, this method is not general: it does not work for other selection types, like point or paintbrush selections, and it is only defined on spatial representations. To address this, we introduce a method that is based on learning a decision tree from the input of the user. We formulate the problem as a binary classification problem, where the decision tree is used to separate the selected from the not-selected points. As the decision three uses information-theoretic measures, it learns a compressed representation of the brushes of the user. For example, the red, rectangular brush in Figure 2 (a), could be stored with four coordinates identifying each side of the rectangle. A range query based on a decision tree, shown in Figure 2 (b), stores a generalized and simplified version with only two rules. We can also generalize a selection, similar to query relaxation [24], based on this idea. By pruning the decision tree by one level, we extract the most important components of the selections, as illustrated in Figure 2 (c). This can be useful to correct imprecise selections.

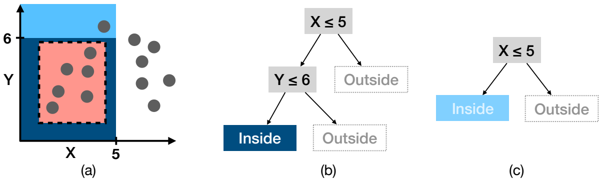


Figure 2: Using decision trees to capture range-based queries. (a) A brush is shown in red. The brush geometry can be described with four rules. (b) The decision tree simplifies the brush to two rules, illustrated in dark blue in (a). (c) A simplified decision tree, where one level has been removed. The result is a simple rule, that also includes a point that was not in the original selection, contained in the light-blue area in (a).

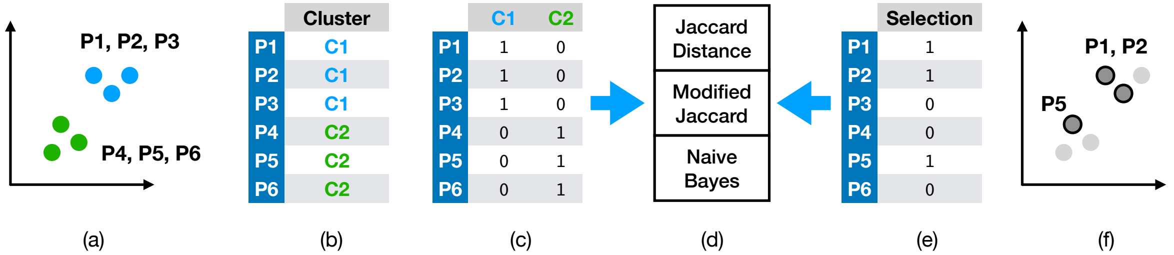
If a dataset contains categorical values, we treat each **category** as a separate pattern. While individual categories could conceivably be shown in the scatterplot, predicting an overlap between a selection and a category is especially important if a dataset has many categories that cannot be shown at the same time.

Finally, we use regression as a framework to analyze **correlations** in the data. To identify the sets of points that are part of a linear or quadratic correlation patterns, we run the following algorithm on linear and quadratic regression dataset:

1. First we assume all the points in the dataset X are inliers I and build a Scikit-learn regression model R on the I.
2. Then we calculate residuals r_i using R for all points x_i in X.
3. Next we define $bar(r) = median(r_i | x_i \in X)$.
4. Then we redefine I as all the points where $r_i < 2 \times bar(r)$.
5. We repeat this for a maximum 10 iterations, stopping early if inliers do not change between iterations.

4.2 Ranking Predictions

All the described patterns result in a classification for each item in the dataset. To rank the predictions in our system, we compare these patterns with a binary classification representing an analyst's selection. Figure 3 shows an overview of our method. Some algorithms, like clustering, produce a multi-class prediction, which we first transform into a set of binary classifications. We can then use a similarity metric to rank each of the predictions. We use a pre-processing step to remove identical predictions from the set of predictions to rank. Identical predictions occur



	Cluster
P1	C1
P2	C1
P3	C1
P4	C2
P5	C2
P6	C2

	C1	C2
P1	1	0
P2	1	0
P3	1	0
P4	0	1
P5	0	1
P6	0	1

	Selection
P1	1
P2	1
P3	0
P4	0
P5	1
P6	0

Figure 3: Overview of ranking patterns for the users' selection. First, we transform the output (a,b) of each of the algorithms into disjoint boolean vectors (c). Analogous, we can express the selection (f) as a boolean vector (e), too. Finally, these boolean vectors act as inputs to the Jaccard rankings, and the naive Bayesian classifier (d).

frequently if a pattern is robust to different parameterizations of the same algorithm. In the following, we discuss three different ways to rank the predicted patterns that are either optimized to infer intent for an existing selection, or to predict intent of a partial selection, plus a special case for ranking range queries.

Ranking for Inferring Intent. Our baseline metric is the Jaccard index, which is a measure of similarities between sets. We consider the set of selected items S, and the set of items in a candidate pattern C. The Jaccard index $J(S,C)$ between those two sets is then defined as

$$J(S,C) = \frac{|S \cap C|}{|S \cup C|} = \frac{|S \cap C|}{|S| + |C| - |S \cap C|}.$$

Here, a value of 1 corresponds to a perfect match, while a value of 0 indicates no overlap. The Jaccard index is well suited to infer the intent of an existing, complete selection.

Ranking for Auto-Complete. The tasks of auto-completing and inferring intent differ with respect to ranking a possible pattern: In the case of inferring intent for a completed selection, it is necessary to find the best match overall. In contrast, for auto-completion, the selection is partial, as the goal of the task is to complete the selection. Hence, we needed to develop a ranking approach that does not penalize incomplete selections. To do this, we rank the predictions using a modified Jaccard index J_m. We define the similarity between sets S and C as,

$$J_m(S,C) = \frac{|S \cap C|}{|S \cap C| + |C \setminus S| + w \times |S \setminus C| + r}$$

The modified similarity metric reduces the penalty for points in S that are not present in C by down-weighting $|S \setminus C|$ using a factor $w < 1$, reflecting the goal of a partial selection to be automatically completed. The metric also adds a regularization parameter of r to prevent boosting ranks in cases where few correct points are selected. Empirically, we found that $w = 0.2$ and $r = 3$ gives good results for datasets that are suitable to be visualized in scatterplots. Due to the regularization the metric never reaches 1, but 0 still indicates no overlap.

Ranking Ranges. Our range-based queries rely on a decision tree of arbitrary depth; hence the pattern captured by that decision tree is always a perfect match to the selection. Consequently, the range query would always rank at the top if we ranked it using the Jaccard index. However, this is inconsistent with what humans perceive as a good prediction of their intent: when analysts create complex selections, it is uncommon that they think of them as long lists of rules. Instead, they likely selected a pattern based on a higher-level relationship in the data. To address this, we assign a score R to the range-based query using a heuristic based on the depth d of the decision tree: $R = \frac{1}{d^2}$. Our heuristic relies on the assumption that simpler queries are more likely to match an analyst's intent than complex queries that require deep decision trees to represent them. The resulting score is on the same scale as the Jaccard index, and hence can be easily integrated.

Probabilistic Ranking. The Jaccard index considers each possible pattern independently. However, an analyst's intent is rarely independent, and some predicted patterns are more likely than others. To address this, we propose a probabilistic framework that models these effects. We denote predicted patterns with $C_i \in \mathbf{C}$ and the boolean vector representing the users' selection as S. Finding a probabilistic ranking of the predicted patterns is the same as determining the conditional probability $P(C|S)$ for each pattern. Framing the problem using probabilities also gives us more interpretability as it relates the different intents to one another: the probabilities for each intent add up to one:

$$\sum_{C_i \in \mathbf{C}} P(C_i|S) = P(C|S)$$

$$\sum_{C_i \in \mathbf{C}} P(C_i|S) = P(C_{\text{Outliers}}|S) + P(C_{\text{Clusters}}|S) + \ldots = 1$$

To compute $P(C_i|S)$ we can use Bayes theorem:

$$P(C_i|S) = \frac{P(S|C_i)P(C_i)}{P(S)}$$

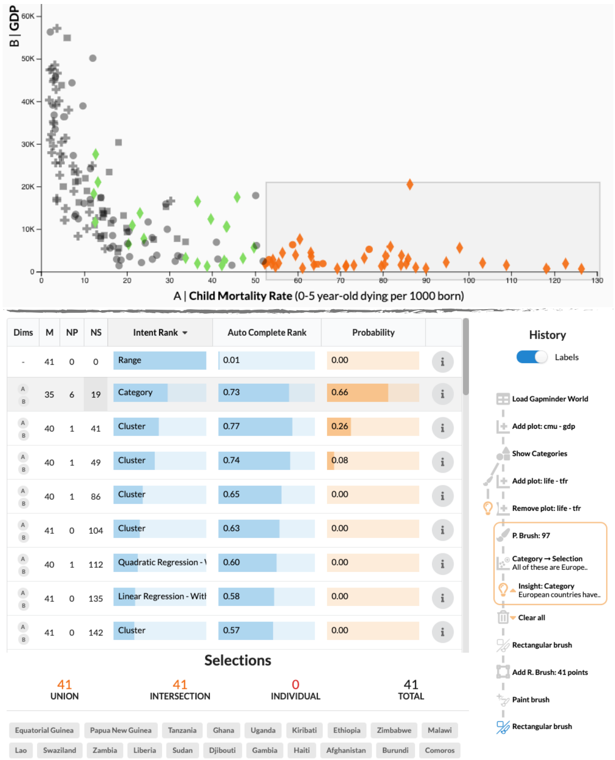
$P(S|C_i)$ models how a particular intent explains the current selection of the user. It is scaled by the term $P(C_i)$, which is called the *prior*. It describes the probability of each intent, without considering additional information.

Later, we will see how we can use this prior to capture domain knowledge about intents. Finally, $P(S)$ acts as a normalizing constant that ensures that the result is a probability. To make this equation computationally tractable, we make use of two observations. First, if we do not consider the order of selections, the problem that we are trying to solve is very similar to text classification. Our description of the users' selection is almost identical to a *bag-of-words* model, which is often used in this domain. The difference is that typically in text classification, the bag-of-words model describes the frequency of each word. In our method, this simplifies to a constant frequency of one if a point is part of the selection. Second, by assuming that each feature (selected point) is independent of another, we can compute $P(S|C_i)$ using the *naive Bayes* method. In particular, we use a *multinomial naive Bayes* classifier to compute the conditional probabilities: For each selection of the user, we train such a classifier on the output vector of each of the intents C_i. Given a selection S as an input, the classifier yields the corresponding probability. Our prediction is then the intent that maximizes this probability. Sometimes selected points are not part of any of the training samples, which leads to zero probabilities for the intents. This is a common problem when using naive Bayes classifiers. We use *Laplacian smoothing* to avoid this effect.

5 Visualization and Interaction Design

In this section, we describe how we implemented our methods in an interactive visualization system and explain our visualization design decisions. The interface allows analysts to add scatterplots as desired. Categories can be visualized using glyph types (see Figure 1). We provide a paint-brush feature with three different brush sizes, rectangular brushes, and individual, click-based selections. The items in multiple rectangular brushes can be treated as unions or as intersections within or between multiple plots. Points that are selected individually or using the paint-brush are always treated as part of the intersection. The labels of the items in a selection are shown in a separate view Figure 4, where we also break down the number of items in the union and intersection of multiple brushes.

We designed the **prediction interface**, shown in Figure 4, as a ranked table. Each predicted pattern is a row. Hovering over a prediction shows a preview; while clicking it replaces the selection with the prediction. The different scores are shown as bar charts in the columns as "Intent Rank" (the Jaccard index), "Auto-Complete Rank" (the Jaccard index modified to be sensitive to partial selections), and "Probability". The table can be sorted based on these scores. Other columns denote the "Matches (M)", i.e., the number of points that the prediction and selection share, the "Not Predicted (NP)" items, i.e., the number of items in the selection but not in the prediction, and the "Not Selected (NS)" items, i.e., the number of items in the prediction but not in the selection. Combined with the similarity scores, these numbers give analysts a sense of how well each prediction matches the selection. Hovering over each of the M, NP, or NS numbers, highlights the corresponding items in green (see Figure 4).



Dims	M	NP	NS	Intent Rank ▾	Auto Complete Rank	Probability	
-	41	0	0	Range	0.01	0.00	i
A B	35	6	19	Category	0.73	0.66	i
A B	40	1	41	Cluster	0.77	0.26	i
A B	40	1	49	Cluster	0.74	0.08	i
A B	40	1	86	Cluster	0.65	0.00	i
A B	41	0	104	Cluster	0.63	0.00	i
A B	40	1	112	Quadratic Regression - \	0.60	0.00	i
A B	41	0	135	Linear Regression - Witl	0.58	0.00	i
A B	41	0	142	Cluster	0.57	0.00	i

Selections

41 UNION
41 INTERSECTION
0 INDIVIDUAL
41 TOTAL

Equatorial Guinea · Papua New Guinea · Tanzania · Ghana · Uganda · Kiribati · Ethiopia · Zimbabwe · Malawi
Lao · Swaziland · Zambia · Liberia · Sudan · Djibouti · Gambia · Haiti · Afghanistan · Burundi · Comoros

History

Labels

- Load Gapminder World
- Add plot: cmu - gdp
- Show Categories
- Add plot: life - tfr
- Remove plot: life - tfr
- P. Brush: 97
- Category → Selection
  All of these are Europe..
- Insight: Category
  European countries have..
- Clear all
- Rectangular brush
- Add R. Brush: 41 points
- Paint brush
- Rectangular brush

Figure 4: The prediction interface shows ranked patterns based on the three different scores. The "Category" prediction for a selection (orange points, rectangle brush) is shown in green in the scatterplot. The M, NP, and NS columns show the number of matching items (M), not predicted items (NP), and predicted but not selected items (NS). Hovering over a cell highlights the corresponding items in the scatterplot in green. The "Dims" column displays the dimensions considered for calculating a pattern. The provenance visualization (bottom right) shows the steps which lead to current selection and prediction. Insights (orange) are used to group and aggregate steps that lead to them. [Interactive Figure]

Each prediction also shows on which dimension it was calculated (and their order) in the "Dims" column. We use short labels, which we replicate on the axes of the scatterplots to identify the dimensions. For range queries, we display the dimensions that are used in the decision tree.

When using **auto-complete**, analysts can sort by the auto-complete score. In addition, a pop-up appears right next to a selection in the scatterplot (Figure 5) showing the top-three predictions for the current selection according to the auto-complete score. This can be used as a short-cut to complete selections.

To enable **reproducibility and recall**, a provenance graph is visualized in the history view (Figure 4). Every persistent action, such as adding a plot or making a brush, is logged in the interface and can be retrieved at a later time. The provenance graph supports branching analysis histories. A prediction can be logged as a semantically meaningful insight, which can be supplemented with an annotation (see the annotation interface in Figure 1). Textual annotations are designed to connect the mid-level intent driven by the patterns in the data to the high-level, domain-specific goals. We use insights to group and aggregate the provenance graph: all actions that were in service of a particular insight are grouped together and can be collapsed. This allows us to show a concise and semantically meaningful analysis history, while still storing a complete history of interactions. The example in Figure 4 shows one expanded group, indicated with an orange frame, and one aggregated insight in the inactive branch on the left.

The provenance graph contains all the information that is necessary to reconstruct the semantics of a selection. That means that it is not just a list of IDs, but contains, for instance, the explicit range query, or the cluster centroid and the algorithm configuration that can be used to reproduce a specific pattern on updated data. In the future, we plan to export the intents into machine-readable form, so that an interactive analysis and filtering session can be used, for example, in computational notebooks.

6 Results

We have implemented our prediction approach in an open-source prototype, and also provide a variety of real and simulated datasets. A demo version of the tool is available at `http://vdl.sci.utah.edu/predicting-intent/`, the source code is available at `https://github.com/visdesignlab/intent-system`.

We demonstrate our results through examples of brushes and the matching prediction. Figure 1, for example, shows a partially selected cluster that is also predicted as a cluster. Figure 4 shows a brush that closely matches a category and a range. Figure 5 shows an example of how our system can be used to auto-complete complex brushes. Here, an analyst has selected four points in a dataset. The plot overall shows a strong linear correlation between X and Y. The selected points, however, don't follow that pattern. Our system recommended "Outlier", "Skyline", and a "Linear Regression — Outside" pattern among the top three patterns, with approximately equal probability (see the supplementary material for a full figure including predictions). The analyst had intended to select the outliers that deviate from that correlation. To do this, they can simply apply the outside linear regression prediction (previewed in green in Figure 5) as their selection. We provide further examples for all patterns in the supplementary material and refer to our prototype for an interactive demonstration.

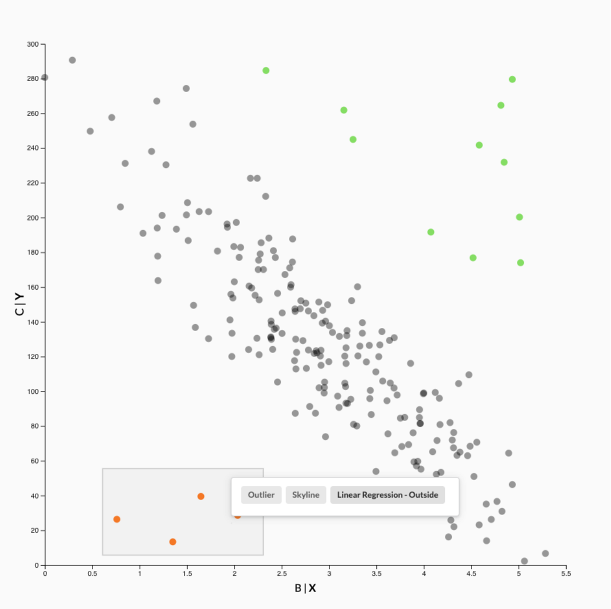


Figure 5: Using predictions to auto-complete complex brushes. By selecting just four points (orange) that deviate from the linear correlation between X and Y in this dataset, the "Linear Regression — Outside" prediction shown in the auto-complete pop-up can be used to reveal other points that deviate from that linear relationship (highlighted in green). Selecting that prediction applies that pattern as the selection. [Interactive Figure]

7 Evaluation

We evaluate our system using two methods: qualitative interviews with domain experts, and a quantitative, crowd-sourced study.

7.1 Qualitative Evaluation with Domain Experts

We demonstrated our system to two experts and interviewed them about their impressions of the tool. The two experts were selected from the participants of our formative interviews: E1 is a graduate student in the chemical engineering program and E2 is a faculty member in nursing. For both interviews, we obtained datasets that each expert works with and loaded them into our system. Both interviews were recorded with the consent of the participants. The study was reviewed by the institutional review board and classified as exempt from full review. For E1 the interview was conducted in person, while we used video conferencing for E2. The interviews began with questions about the experts' current workflow, followed by a demonstration of the features of our system. After the demonstration, each expert was asked to use the system to select interesting sets of points in their data and to describe why those points are meaningful, as well as to explore and comment on the intent predictions and logging mechanism.

Both E1 and E2 described their current workflows as using scatterplots to visually find data items of interest, which they then extract for downstream analysis. E2, in particular, commented on his preference to visually identify interesting patterns and sets of items: "I'm one of the statisticians that tends to be more visual. I can see the relationships easier than just running tens and tens of models and looking at tables of output to try and get an understanding of the variability within our data." Both experts approached their use and evaluation of the predictions within our system predominately as a mechanism for data item selection. From this perspective, they both found the range intent was a good match for their selections and speculated that the cluster intent would also work well for datasets that had clear clusters. E2 noted that the predictions are like "somebody sitting over my shoulder looking at how I lasso and select the [data items], and having a guess about it."

The other predictions did not closely match their manual selections. They noted, however, that these other predictions could be interesting to consider as alternative sets of items or patterns that they would not have otherwise considered. Both experts requested information about what the specific analysis methods were and how they were selecting data items, noting that before they would consider data items suggested by the predictions they would first want to verify the appropriateness of the underlying algorithms for their analysis. These comments suggest that the predicted intents, along with algorithmic descriptions, could be incorporated into other tools as a mechanism to support creativity and brainstorming during analysis.

As both experts focused on the tool and intent predictions as a means to select data items of interest, it was difficult to engage the experts in speculating about the efficacy of the predictions for tracking intent as part of a provenance workflow. Furthermore, neither expert currently uses automated mechanisms for capturing provenance of their exploratory visual analysis; E1, in particular, noted the difficulty she has in keeping track of her selections and analysis over multiple data sets, tools, and parameters. Both experts commented that the annotation mechanism coupled to selections would be useful for communication with colleagues. E2 also suggested that annotating with the predicted intents could be a useful way to get selections from scatterplots into a downstream analysis tool like R. This insight suggests that annotated intents could be an effective mechanism for chaining together analysis tools.

7.2 A Quantitative, Crowdsourced User study

We conducted a crowdsourced user study with 128 participants on Prolific to demonstrate that our methods for predicting patterns are valid, that they match the mental model of the participants, and that our system can be used to auto-complete selections. We focus on our application scenario of auto-complete for practical reasons: a quantitative study to measure the value of captured intent in a provenance graph is difficult to conceive, in particular in a crowd-sourced context. In contrast, it is possible to instruct participants to select a pattern, such as a cluster, and measure their performance in doing so with and without our system.

We chose a subset of our patterns: correlations (linear and quadratic), outliers, clusters, and multivariate optimization. We excluded ranges, as they are not useful for auto-completing selections; and categories, as selecting elements belonging to categories would be tedious in our system without auto-complete and yet an alternative user interface design that enables participants to explicitly select categories would solve that problem trivially.

Procedure. We used a within-subjects design for two conditions: **user-driven**, using only manual brushes, and **computer supported**, which adds a simplified version of our prediction interface. The interface was simplified to show the top-three predictions in the auto-complete popup, and a simple version of the prediction interface on the side, using only a single score (see supplementary material or visit the study interface). The names of the predicted patterns were not shown, to avoid biasing participants. To counter-balance any learning effects, the conditions were assigned in random order, the task order in each condition was randomized, and the dataset was randomized. Based on completion times of pilot experiments, each participant was paid $ 6.25 USD, for an estimated duration of 25 minutes, resulting in an hourly rate of about $ 15 USD. All participants viewed and agreed to an IRB-approved consent form. To be eligible, participants had to use a laptop or desktop device and either Chrome or Firefox browsers.

Our procedure consisted of five phases and follows guidelines on training participants for complex analysis tasks [35]: *passive training*, in the form of an 8-minute video introducing the types of patterns and the interface, *active training*, where they had to complete representative tasks, but could use a help-feature to reveal the answer, *trials* in the two conditions, and a short *post-study survey*. The tasks instructed participants to select one of the patterns they learned about during training. For clusters, a specific cluster was marked in the plot with a red cross. See the supplementary Figures S9 and S10 for the study interface and the study interface for specific task instructions.

Data and Tasks. We generate synthetic two-dimensional datasets with between 200 and 222 items for linear correlations, quadratic correlations, outliers combined with a linear correlation, outliers combined with a single cluster, clustered datasets with three or four cluster, and datasets for multivariate optimizations, each in three levels of difficulty: easy, medium, and hard. The levels of difficulty are driven by how apparent a pattern is. For example, an easy clustering dataset has fully separated clusters, whereas a hard dataset has clusters that significantly overlap. We generate two variations of each combination (to be used in the different experimental conditions), for a total of 36 datasets for the study and 6 datasets for training tasks.

For each dataset, we generate ground-truth through human labeling. Patterns such as clusters or outliers can be ambiguous and our goal is to match the human perception of those patterns. So, we chose to ask expert coders to label the datasets. Our coders were five doctoral students in visualization, with experience in analyzing these patterns. We instructed them to carefully label each dataset for a specific pattern, with no algorithmic support. We then treat all points that 4–5 of our coders selected as correct, the points that 2–3 coders selected as ambiguous (neither correct nor incorrect), and the points that only a single coder selected as incorrect. The supplementary material contains images of the datasets, the ground truth labels, and the code used to generate them.

Measures. We measure accuracy, time to completion, the type and rank of a predicted pattern chosen by a participant, and survey responses. After each question, we also elicit confidence and perceived difficulty on a five-point Likert scale, and ask for comments. We also log detailed interactions in a provenance graph. We calculate the accuracy of the participant's responses by using the Jaccard index of the response overlapping with the ground truth, where we first remove the ambiguous points (hence, selecting ambiguous points neither benefits nor penalizes a score). For our time measures, we subtract the times where the browser window showing the study was inactive. The final survey asks about the satisfaction with different features, and experience with visualization and statistics. Demographic data is provided through Prolific user profiles.

Pilots, Analysis, and Experiment Planning. We conducted several tests and pilots to evaluate tasks, system usability, data collection modalities, measures, and our procedure. We estimated the number of participants required to uncover effects based on a pilot run on Prolific with 10 participants. We used a power analysis to estimate the variance in our measures, which we combined with our observed means to estimate the number of trials required. Due to the limitations of null hypothesis significance testing, we base our analysis on best practices for fair statistical communication in HCI [11] by reporting confidence intervals and effect sizes. We compute 95% bootstrapped confidence intervals [7] and effect sizes using Cohen's d to indicate a standardized difference between two means. For the accuracy values of the medium/difficult tasks, we also supplement our analysis by including p-values from Wilcoxon signed-rank tests (given the non-normal distributions of our data and the within subjects design). We consider a Bonferroni-corrected threshold for significance of $p = 0.01$.

Expectations. We expected that accuracy will be higher using computer-supported mode for the medium and hard datasets, and that accuracy will be about the same and consistently very high with the easy datasets. We assume that the value of the prediction system will be greater on ambiguous patterns, and that obvious patterns will be easy to select manually, given the brushing tools we provide. We also expected that participants will perceive predictions as accurate, the interface as user-friendly, and prefer computer-supported mode. Finally, we initially also expected computer-supported mode to be faster, but we realized during testing and pilots that this would unlikely be the case.

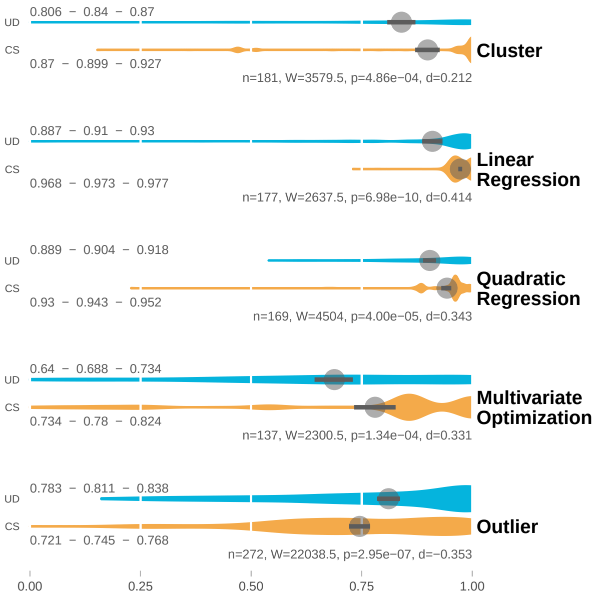


Figure 6: Tasks specific accuracy for combined medium and hard tasks shown as medians and 95% confidence intervals. Blue (UD) encodes the user-driven condition, orange (CS) the computer-supported condition. Violin plots visualize the underlying distribution. We also give the number of trials per condition for each task (n), Cohen's d for effect sizes (d), and p-values. All differences are significant.

Results. 128 participants completed the study. After reviewing the provenance data using a custom visualization (see `https://vdl.sci.utah.edu/intent-study-analysis/`) we realized that participants sometimes chose not to use predictions in the computer-supported condition. Since our goal was to measure the effects of using predictions, we removed trials that were not completed using predictions in computer-supported mode. To avoid biasing our data by removing low-effort results in one condition, we also always removed the equivalent trial in user-driven mode. We include data for all trials in our supplemental material. Based on these criteria, we retained 1381 of 2268 trials in each the computer-supported and user-driven. Hence, when auto-complete was available, participants chose to use it in 61% of cases. We analyze medium and hard tasks together (936 valid trials), and treat simple tasks as a baseline. For clusters, linear, and quadratic regression, the accuracy in easy tasks is $> 95\%$ in both conditions. For multivariate optimization and outliers, the effects, both in terms of accuracy and trends are similar to the medium and hard tasks, as shown in supplementary Figure S14. Accuracy and speed for every task individually are shown in supplementary Figure S16.

Figure 6 summarizes our main results. For the medium and hard tasks, accuracy was also fairly high in both conditions for clusters, linear regression, and quadratic regression (median of 84-97%), with a small to medium, significant effect showing higher accuracy in the computer-supported condition. The computer-supported clustering condition shows a small "bump" at an accuracy of around 0.5. Analysis of provenance data has revealed that this bump is due to one of the clustering predictions aggregating two ground-truth clusters into one.

Overall accuracy for the multivariate optimization task was lowest overall, with accuracy in the computer-supported condition being significantly higher, with a small to medium effect size. Interestingly, many of our expert coders omitted points that are contained in the formal definition of a skyline, resulting in a "bump" of accuracy scores at around 0.85, representing participants who have selected the formally correct skyline as recommended by the algorithm.

The accuracy for outliers was, to our surprise, significantly lower, with a small to medium effect size, in the computer-supported condition than in the user-driven condition. Our intervention lead to worse performance! Inspection of the provenance data revealed that in many cases, applying a prediction for outliers made user selections worse. We believe that this is due to a mismatch between how the experts coded an outlier, and what the algorithm and its parameterization underlying our prediction considered an outlier. Generally, we found that our coders and participants considered only clear outliers, while the algorithm also considered points at the margin of a pattern to be outliers.

Time to completion was generally slower by about 5 seconds (with completion times ranging from 21-31 seconds on average) for the computer-supported condition for the medium and hard tasks (see supplementary Figure S12). Again, the exceptions were outlier tasks that were slower by about 8 seconds on average. We speculate that the failure of our system to provide a prediction consistent with the participant's mental model caused that delay. Previous work by Saket et al. has shown that task completion in multi-paradigm interfaces can be slower, compared to a single paradigm interface [41]. They argued that optimizing efficiency is not a suitable goal in many contexts and that multi-paradigm tools can make analysts think more carefully.

Confidence was higher, and difficulty was perceived lower in the computer-supported condition, for all tasks except for outliers, where they were about the same (see supplemental Figure S13), suggesting that participants trusted the predictions when they matched their mental model. We also analyzed whether the type of predictions chosen by participants matched the patterns they were instructed to select, which is a useful metric to judge the quality of our predictions and rankings. Figure S18 shows the results. We see a strong overlap between prediction and target pattern. Notably, quadratic and linear regression were frequently substituted, and non-outliers were also frequently chosen for regression tasks.

Our exit survey revealed that participants generally found predictions accurate (average score of 3.6 on a 5-point Likert scale), found the prediction interface easy to use (average score of 4.2), found them helpful (average score of 3.8) but liked user-driven and computers-supported mode about equally well (average 3). The (default) paint-brush feature was widely appreciated (average score of 4.5), the rectangle brush tool was barely used, according to provenance data, and lowly rated (average score of 2.2).

Overall, our results are largely consistent with our expectations. The negative result for outliers shows that it is imperative that the mental model of a pattern matches with the predictions of an algorithm, and highlights the challenge in identifying suitable parameters. The otherwise higher accuracy for datasets of medium or hard difficulty leads us to conclude that our method for predicting intents works, except for detecting outliers. We were surprised by the evenly split preference between user-driven and computer-supported mode. The data is bimodal, suggesting that participants either liked or disliked computer-supported mode. We hope that a more detailed analysis of the provenance data will be able to shed light on this effect.

8 Discussion

In this work, we demonstrate a method for semi-automatically detecting and capturing analysts' mid-level intents. We argue that detecting intents is useful for two scenarios: to auto-complete selections, and to be able to semi-automatically record semantically rich insights in provenance data and therefore make visual analysis processes reproducible and justifiable. By capturing mid-level intents, we can, for example, more easily create curated analysis stories by leveraging ideas from our prior work on using provenance information to create interactive data stories [22]. The capability to capture mid-level intents opens up numerous other prospects as well.

Integration in Computational Workflows and Analysis Re-Use. Our interviews show that analysts frequently use scatterplots in combination with statistical modeling tools and computational notebooks, such as R-Markdown or Jupyter notebooks. Having semantically meaningful intents available means that we can generate robust analysis scripts based on interactive visualization, supporting more automatic computation workflows. For example, if an analyst uses our tool to select a specific cluster for downstream analysis, we will be able to generate code that will select this cluster even for updated data.

Learning from Interaction. Through large-scale capturing of intents, we can empirically learn patterns that analysts select to further improve our predictions. Such a system could dynamically "auto-correct" analysis and allow large-scale feedback on the usefulness and effectiveness of various features within complex tools. For instance, a software tool with a diverse set of users and skill levels would allow intent to be trained on experienced users so that novices are guided quickly towards effective strategies [5].

Other Visualization Techniques and Data Types. We chose to limit ourselves to scatterplots and tabular data because we believe that these are important cases that can be used to demonstrate the feasibility of our approach. There are numerous straightforward extensions and generalizations of our work. This ranges from implementing more brushing tools, such as lasso selections, to also allowing analysts to filter datasets. We argue that our framework could be extended to other visualization techniques, such as parallel coordinates, histograms, or tabular visualizations [17] with small adaptions. Other visualization techniques could also provide additional clues we could use for predicting intents. For example, in a tabular visualization, the action of sorting a table is likely important to understand the intent of a subsequent selection. Other data types, such as time-series or network data, are likely amenable to the same approach, but would require identifying appropriate patterns and the corresponding algorithms first.

Higher Dimensionality. While we allow users to explore multiple two-dimensional views, building a mental model of high-dimensional data can be difficult. A potential solution to this is dynamic dimensionality reduction. That is, given points already selected, the system could dynamically adjust a linear projection (e.g.P̃CA) to the best capture those datasets in a 1-, or 2-dimensional subspace. Alternatively, given more complex selections, like clusters of relevant points, the dimensionality reduction can use techniques such as Latent Discriminant Analysis to find the best linear projection towards separating the clusters. Another approach is to label pairs of points that should be close (or far). Using these pairs, a similarity learning method could provide the best linear projection that satisfies those constraints. An intent-driven tool could suggest the most informative point-pairs to label.

9 Limitations

Even in a simple scatterplot environment, there are numerous complexities our work has identified. When more than $2--4$ dimensions of a data set are relevant in the exploration, the combinatorial complexity of all the possible intents we model is significant. Complex actions would place an even larger burden on the prediction engine. One potential solution is the automatic filtering of entire classes of intents so that not all of them need to be explicitly explored.

Our tool currently does not handle missing data. When working with our collaborators, we frequently encountered datasets that were generally well suited to our approach but contained invalid or missing cells. On the front-end, we plan to provide separate views for items with missing data. On the back-end, appropriate interpolation and fitting strategies could be a solution.

Our current approach to parameter space exploration is naive. We could potentially improve our prediction by evaluating our classifications using methods such as silhouette analysis for clustering and varying the parameters accordingly.

10 Conclusion

In this paper, we introduce the first approach to predict, capture, and annotate mid-level intents of analysts as they interact with data in a scatterplot. We use a mixed-initiative approach, leveraging data mining methods to identify patterns in datasets, ranking potential matches based on user selections, and allowing analysts to specify which (if any) of the predicted intents fits their actual intents. We discuss two application scenarios: auto-completing selections, and increasing reproducibility. We believe that our work will form the foundation of many future projects. Immediate next steps are the application to different visualization techniques and data types. Other prospects include learning from interactions and integrating the output of interactions in visualizations into computational workflows.

11 Acknowledgements

We thank the domain experts we interviewed for their time and their willingness to provide datasets, and Lane Harrison and members of the Visualization Design Lab for feedback. We gratefully acknowledge funding by the National Science Foundation (IIS 1751238) and by the Deutsche Forschungsgemeinschaft (251654672-TRR 161).

References

[1] L. Bavoil, S. P. Callahan, C. Scheidegger, H. T. Vo, P. Crossno, C. T. Silva, and J. Freire. VisTrails: Enabling Interactive Multiple-View Visualizations. In *Proceedings of the IEEE Conference on Visualization (VIS '05)*, pp. 135–142. IEEE, 2005. doi: 10.1109/VISUAL.2005.1532788

[2] R. A. Becker and W. S. Cleveland. Brushing Scatterplots. *Technometrics*, 29(2):127–142, 1987. doi: 10.1080/00401706.1987.10488204

[3] S. Borzsony, D. Kossmann, and K. Stocker. The Skyline operator. In *Proceedings 17th International Conference on Data Engineering*, pp. 421–430, Apr. 2001. doi: 10.1109/ICDE.2001.914855

[4] E. T. Brown, A. Ottley, H. Zhao, Q. Lin, R. Souvenir, A. Endert, and R. Chang. Finding Waldo: Learning about Users from their Interactions. *IEEE Transactions on Visualization and Computer Graphics*,

20(12):1663–1672, Dec. 2014. doi: 10.1109/TVCG.2014.2346575

[5] D. Ceneda, T. Gschwandtner, and S. Miksch. A Review of Guidance Approaches in Visual Data Analysis: A Multifocal Perspective. *Computer Graphics Forum*, 38(3):861–879, 2019. doi: 10.1111/cgf.13730

[6] Y. Chen, S. Barlowe, and J. Yang. Click2Annotate: Automated Insight Externalization with rich semantics. In *2010 IEEE Symposium on Visual Analytics Science and Technology*, pp. 155–162, Oct. 2010. doi: 10.1109/VAST.2010.5652885

[7] G. Cumming. *Understanding the New Statistics: Effect Sizes, Confidence Intervals, and Meta-Analysis*. Routledge, 2013.

[8] E. Deelman, D. Gannon, M. Shields, and I. Taylor. Workflows and e-Science: An overview of workflow system features and capabilities. *Future Generation Computer Systems*, 25(5):528–540, May 2009. doi: 10.1016/j.future.2008.06.012

[9] M. Derthick, J. Kolojejchick, and S. F. Roth. An Interactive Visual Query Environment for Exploring Data. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, UIST '97, pp. 189–198. ACM, New York, NY, USA, 1997. doi: 10.1145/263407.263545

[10] W. Dou, D. H. Jeong, F. Stukes, W. Ribarsky, H. R. Lipford, and R. Chang. Recovering reasoning process from user interactions. *IEEE Computer Graphics & Applications*, 29(3):52–61, 2009.

[11] P. Dragicevic. Fair Statistical Communication in HCI. In *Modern Statistical Methods for HCI*, Human–Computer Interaction Series, pp. 291–330. Springer, 2016. doi: 10.1007/978-3-319-26633-6_13

[12] C. Dunne, N. Henry Riche, B. Lee, R. Metoyer, and G. Robertson. GraphTrail: Analyzing Large Multivariate, Heterogeneous Networks While Supporting Exploration History. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*, pp. 1663–1672. ACM, 2012. doi: 10.1145/2207676.2208293

[13] R. Eccles, T. Kapler, R. Harper, and W. Wright. Stories in GeoTime. *Information Visualization*, 7(1):3–17, Mar. 2008. doi: 10.1057/palgrave.ivs.9500173

[14] C. Fan and H. Hauser. Fast and Accurate CNN-based Brushing in Scatterplots. *Computer Graphics Forum*, 37(3), 2018. doi: 10.1111/cgf.13405

[15] J. Freire, D. Koop, E. Santos, and C. T. Silva. Provenance for computational tasks: A survey. *Computing in Science & Engineering*, 10(3):11–21, 2008.

[16] T. Fujiwara, T. Crnovrsanin, and K.-L. Ma. Concise provenance of interactive network analysis. *Visual Informatics*, Dec. 2018. doi: 10.1016/j.visinf.2018.12.002

[17] K. Furmanova, S. Gratzl, H. Stitz, T. Zichner, M. Jaresova, M. Ennemoser, A. Lex, and M. Streit. Taggle: Scalable Visualization of Tabular Data through Aggregation. *Information Visualization*, 2019. doi: 10.1177/1473871619878085

[18] M. J. Gingerich and C. Conati. Constructing Models of User and Task Characteristics from Eye Gaze Data for User-Adaptive Information Highlighting. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, Feb. 2015.

[19] J. Goecks, A. Nekrutenko, J. Taylor, and T. G. Team. Galaxy: A comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol*, 11(8):R86, 2010.

[20] D. Gotz and Z. Wen. Behavior-driven visualization recommendation. In *Proceedings of the Conference on Intelligent User Interfaces (IUI '09)*, pp. 315–324. ACM, 2009. doi: 10.1145/1502650.1502695

[21] D. Gotz and M. X. Zhou. Characterizing Users' Visual Analytic Activity for Insight Provenance. *Information Visualization*, 8(1):42–55, 2009. doi: 10.1057/ivs.2008.31

[22] S. Gratzl, A. Lex, N. Gehlenborg, N. Cosgrove, and M. Streit. From Visual Exploration to Storytelling and Back Again. *Computer Graphics Forum*, 35(3):491–500, 2016. doi: 10.1111/cgf.12925

[23] D. P. Groth and K. Streefkerk. Provenance and Annotation for Visual Exploration Systems. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1500–1510, 2006. doi: 10.1109/TVCG.2006.101

[24] J. Heer, M. Agrawala, and W. Willett. Generalized Selection via Interactive Query Relaxation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pp. 959–968. ACM, New York, NY, USA, 2008. doi: 10.1145/1357054.1357203

[25] J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala. Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '08)*, 14(6):1189–1196, 2008. doi: 10.1109/TVCG.2008.137

[26] J. Heer and B. Shneiderman. Interactive dynamics for visual analysis. *Communications of the ACM*, 55(4):45–54, 2012. doi: 10.1145/2133806.2133821

[27] N. Kadivar, V. Chen, D. Dunsmuir, E. Lee, C. Qian, J. Dill, C. Shaw, and R. Woodbury. Capturing and supporting the analysis process. In *2009 IEEE Symposium on Visual Analytics Science and Technology*, pp. 131–138, Oct. 2009. doi: 10.1109/VAST.2009.5333020

[28] M. Kreuseler, T. Nocke, and H. Schumann. A History Mechanism for Visual Data Mining. In *Proceedings*

of the IEEE Symposium on Information Visualization (InfoVis '04), pp. 49–56. IEEE, 2004. doi: 10.1109/INFVIS.2004.2

[29] B. C. Kwon, F. Stoffel, D. Jäckle, B. Lee, and D. Keim. VisJockey: Enriching Data Stories through Orchestrated Interactive Visualization. *In Poster Compendium of the Computation + Journalism Symposium*, 2014.

[30] J. Mackinlay, P. Hanrahan, and C. Stolte. Show Me: Automatic Presentation for Visual Analysis. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '07)*, 13(6):1137–1144, 2007. doi: 10.1109/TVCG.2007.70594

[31] A. R. Martin and M. O. Ward. High Dimensional Brushing for Interactive Exploration of Multivariate Data. In *Proceedings of the IEEE Conference on Visualization (Vis '95)*, pp. 271–278. IEEE Computer Society Press, 1995. doi: 10.1109/VISUAL.1995.485139

[32] A. Mathisen, T. Horak, C. N. Klokmose, K. Grø nbæk, and N. Elmqvist. InsideInsights: Integrating Data-Driven Reporting in Collaborative Visual Analytics. *Computer Graphics Forum*, 38(3):649–661, 2019. doi: 10.1111/cgf.13717

[33] B. A. Myers. *Creating User Interfaces by Demonstration*. Phd, 1987.

[34] P. H. Nguyen, K. Xu, A. Wheat, B. L. W. Wong, S. Attfield, and B. Fields. SensePath: Understanding the Sensemaking Process Through Analytic Provenance. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):41–50, Jan. 2016. doi: 10.1109/TVCG.2015.2467611

[35] C. Nobre, D. Wootton, L. Harrison, and A. Lex. Evaluating multivariate network visualization techniques using a validated design and crowdsourcing approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI), to Appear*. ACM, 2020.

[36] C. North, R. Chang, A. Endert, W. Dou, R. May, B. Pike, and G. Fink. Analytic Provenance: Process+Interaction+Insight. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, pp. 33–36. ACM, New York, NY, USA, 2011. doi: 10.1145/1979742.1979570

[37] A. Ottley, R. Garnett, and R. Wan. Follow The Clicks: Learning and Anticipating Mouse Interactions During Exploratory Data Analysis. *Computer Graphics Forum*, 38(3):41–52, 2019. doi: 10.1111/cgf.13670

[38] S. G. Parker and C. R. Johnson. SCIRun: A Scientific Programming Environment for Computational Steering. In *Proceedings of the ACM/IEEE Conference on Supercomputing (SC '95)*, p. 52. ACM, 1995.

[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[40] E. Ragan, A. Endert, J. Sanyal, and J. Chen. Characterizing Provenance in Visualization and Data Analysis: An Organizational Framework of Provenance Types and Purposes. *IEEE Transactions on Visualization and Computer Graphics (VAST '15)*, 22(1):31–40, 2016. doi: 10.1109/TVCG.2015.2467551

[41] B. Saket, L. Jiang, C. Perin, and A. Endert. Liger: Combining Interaction Paradigms for Visual Analysis. *arXiv:1907.08345 [cs]*, July 2019.

[42] B. Saket, H. Kim, E. T. Brown, and A. Endert. Visualization by Demonstration: An Interaction Paradigm for Visual Data Exploration. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):331–340, Jan. 2017. doi: 10.1109/TVCG.2016.2598839

[43] B. Shneiderman. Dynamic queries for visual information seeking. *IEEE Software*, 11(6):70–77, Nov. 1994. doi: 10.1109/52.329404

[44] Y. B. Shrinivasan and J. J. van Wijk. Supporting the analytical reasoning process in information visualization. In *Proc. CHI 2008*, pp. 1237–1246. ACM, 2008. doi: 10.1145/1357054.1357247

[45] B. Steichen, G. Carenini, and C. Conati. User-adaptive information visualization: Using eye gaze data to infer visualization tasks and user cognitive abilities. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, IUI '13, pp. 317–328. Association for Computing Machinery, Santa Monica, California, USA, Mar. 2013. doi: 10.1145/2449396.2449439

[46] H. Stitz, S. Gratzl, H. Piringer, T. Zichner, and M. Streit. KnowledgePearls: Provenance-Based Visualization Retrieval. *IEEE Transactions on Visualization and Computer Graphics (VAST '18)*, 25(1):120–130, 2019. doi: 10.1109/TVCG.2018.2865024

[47] M. Streit, H.-J. Schulz, A. Lex, D. Schmalstieg, and H. Schumann. Model-Driven Design for the Visual Analysis of Heterogeneous Data. *IEEE Transactions on Visualization and Computer Graphics*, 18(6):998–1010, 2012. doi: 10.1109/TVCG.2011.108

[48] S. L. Su, S. Paris, and F. Durand. QuickSelect: History-based Selection Expansion. In *Proceedings of Graphics Interface 2009*, GI '09, pp. 215–221. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 2009.

[49] M. Tory and V. Setlur. Do What I Mean, Not What I Say! Design Considerations for Supporting Intent and Context in Analytical Conversation. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 93–103, Oct. 2019. doi: 10.1109/VAST47406.2019.8986918

[50] M. Wohlfart and H. Hauser. Story Telling for Presentation in Volume Visualization. In *Proceedings of the*

Symposium on Visualization (EuroVis '07), pp. 91–98. Eurographics Association, 2007. doi: 10.2312/VisSym/EuroVis07/091-098

[51] L. Xiao, J. Gerth, and P. Hanrahan. Enhancing Visual Analysis of Network Traffic Using a Knowledge Representation. In *Visual Analytics Science And Technology, 2006 IEEE Symposium On*, pp. 107–114, 31 2006-nov. 2. doi: 10.1109/VAST.2006.261436

[52] B. Yu and C. T. Silva. VisFlow - Web-based Visualization Framework for Tabular Data with a Subset Flow Model. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '16)*, 23(1):251–260, 2017. doi: 10.1109/TVCG.2016.2598497