Finding Waldo: Learning about Users from their Interactions

Eli T Brown, Alvitta Ottley, Helen Zhao, Quan Lin, Richard Souvenir, Alex Endert, Remco Chang

Fig. 1. The interface from our user study in which participants found Waldo while we recorded their mouse interactions. Inset (a) shows Waldo himself, hidden among the trees near the top of the image. Distractors such as the ones shown in inset (b) and (c) help make the task difficult.

Abstract— Visual analytics is inherently a collaboration between human and computer. However, in current visual analytics systems, the computer has limited means of knowing about its users and their analysis processes. While existing research has shown that a user's interactions with a system reflect a large amount of the user's reasoning process, there has been limited advancement in developing automated, real-time techniques that mine interactions to learn about the user. In this paper, we demonstrate that we can accurately predict a user's task performance and infer some user personality traits by using machine learning techniques to analyze interaction data. Specifically, we conduct an experiment in which participants perform a visual search task, and apply well-known machine learning algorithms to three encodings of the users' interaction data. We achieve, depending on algorithm and encoding, between 62% and 83% accuracy at predicting whether each user will be fast or slow at completing the task. Beyond predicting performance, we demonstrate that using the same techniques, we can infer aspects of the user's personality factors, including locus of control, extraversion, and neuroticism. Further analyses show that strong results can be attained with limited observation time: in one case 95% of the final accuracy is gained after a quarter of the average task completion time. Overall, our findings show that interactions can provide information to the computer about its human collaborator, and establish a foundation for realizing mixed-initiative visual analytics systems.

Index Terms— User Interactions, Analytic Provenance, Visualization, Applied Machine Learning

---◆---

1 INTRODUCTION

Visual analytics systems integrate the ability of humans to intuit and reason with the analytical power of computers [24]. At its core, visual analytics is a collaboration between the human and the computer. Together, the two complement each other to produce a powerful tool for solving a wide range of challenging and ill-defined problems.

Since visual analytics fundamentally requires the close collaboration of human and computer, enabling communication between the two is critical for building useful systems [41]. While the computer can communicate large amounts of information on screen via visualization, the human's input to an analytic computer system is still largely limited to mouse and keyboard [28]. This human-to-computer connection provides limited bandwidth [22] and no means for the human to express analytical needs and intentions, other than to explicitly request the computer to perform specific operations.

Researchers have demonstrated that although the mouse and keyboard appear to be limiting, a great deal of a user's analytical intent and strategy, reasoning processes, and even personal identity can be recovered from this interaction data. Machine learning researchers have recovered identity for re-authenticating specific users in real time using statistics over raw mouse interactions [30, 36, 37, 45] and keyboard inputs [27], but classified only identity, no user traits or strategies. In visual analytics, Dou et al. [11] have shown that strategies can be extracted from interaction logs alone, but at the cost of many hours of tedious labor. Unfortunately these manual methods are not feasible for real-time systems to adapt to users. The techniques needed to learn about users and their strategies and traits in real time do not exist to our knowledge.

In this paper, we demonstrate on a small visual analytics subtask that it is indeed possible to automatically extract high-level semantic information about users and their analysis processes. Specifically, by using well-known machine learning techniques, we show that we can: (1) predict a user's task performance, and (2) infer some user personality traits. Further (3), we establish that these results can be achieved quickly enough that they could be applied to real-time systems.

Our conclusions draw from an online experiment we conducted to

- *Eli T. Brown is with Tufts U. E-mail: ebrown@cs.tufts.edu.*
- *Alvitta Ottley is with Tufts U. E-mail: alvitta.ottley@tufts.edu.*
- *Jieqiong Zhao was with Tufts U., now with Purdue U. E-mail: jieqiongzhao@purdue.edu.*
- *Quan Lin was with Tufts U. E-mail: linquan0201@gmail.com.*
- *Alex Endert is with Pacific Northwest National Lab. E-mail: alex.endert@pnnl.gov.*
- *Richard Souvenir is with U.N.C. Charlotte. E-mail: souvenir@uncc.edu.*
- *Remco Chang is with Tufts U. E-mail: remco@cs.tufts.edu.*

simulate a challenging visual search task that one might encounter as a component of a visual analytics application with the game *Where's Waldo* (see Figure 1). The participants were given a visualization enabling a set of interactions (panning and zooming) to explore the image and find the character Waldo. During the participants' search process, we collect a wide range of information about their interactions, including the state of the visualization, and the time and location of all mouse events.

Inspired partly by our visualization of the user paths through the search image, we used this low-level interaction data to create three encodings that capture three major aspects of visual analytics: data, user and interface. The encodings are: (1) *state-based*, which captures the total state of the software based on what data (portion of the image) is showing, (2) *event-based*, which captures the user's actions through statistics of the raw mouse activity, and (3) *sequence-based*, which encodes sequences of clicks on the interface's buttons. The encoded information is then analyzed using well-known machine learning techniques such as support vector machines (SVM) and decision trees to classify groups of users with performance outcomes and individual differences.

The results of our analyses demonstrate that we can indeed automatically extract users' task performance, and infer aspects of their personality traits from interaction data alone. Further, task performance can be estimated quickly enough to be used in a real-time system. Depending on which data encoding with its corresponding machine learning algorithm, we attain between 62% and 83% accuracy at differentiating participants who completed the task quickly versus slowly, with state-based yielding up to 83%, event-based up to 79% accuracy, and sequence-based 79%.

With the goal of uncovering more intrinsic user factors, we applied the same techniques to classify participants on personality traits, and found promising signal. In particular, we can classify users based on three of their personality traits: locus of control, extraversion, and neuroticism with 61% to 67% accuracy. The correlation between these three personality traits and the participants' performance are consistent with previous findings in the visual analytics community on individual differences [17, 33, 47].

Finally, on applying the techniques in real-time, we show that accurate prediction of the user's task performance and personality traits can be achieved after observing users for a limited time period. Using the same encoding and analysis techniques described above, we build classifiers based on a limited amount of the user's interaction logs. We demonstrate encouraging results for employing this technology in real-time systems, e.g. with only two minutes of observation on a task that requires an average of nearly eight minutes to complete, we can correctly classify the users with an average of 84% of the final accuracy.

Overall, our contributions to visual analytics are that we:

- Show that participants can be classified as fast or slow at the visual search task by applying machine learning to three encodings of participants' interaction data: (1) state-based, (2) event-based, and (3) sequence-based.

- Apply these same techniques to classify participants based on personality traits and demonstrate success for the traits locus of control, extraversion and neuroticism.

- Evaluate the plausibility of applying this work to real-time systems by providing results using shorter timespans of data collection.

2 RELATED WORK

The goal of inferring information about a user based on his or her interactions is generally referred to as "user modeling" and has been an active area of research in a variety of domains. Although the field is broad, our goal is narrower in that we specifically emphasize learning users' reasoning processes and characteristics as opposed to modeling users' tasks (as seen with the Microsoft Office Assistant, *Clippy*).

2.1 Analytic Provenance

Perhaps most related to our work is the research in the domain of *analytic provenance* in the visual analytics community. Similar to our goal, researchers in analytic provenance believe that the analysis process during an analytic task is just as important as the analysis product [32]. Through analyzing a user's interactions, researchers in analytic provenance seek to identify *how* a user discovers insight and how the same procedures can be stored and reapplied to automatically solve other similar problems [44, 23].

Many systems have been developed in the visual analytics community for logging, storing, and analyzing a user's interactions and activities. For example, the GlassBox system by Cowley et al. [9] records low-level events generated by the interface (such as copy, paste, window activation, etc.). At a higher level, VisTrails captures the user's steps in a scientific workflow [3]. Finally, at a model level, Endert et al. showed that user interactions can be analyzed systematically, and directly used to perform model steering operations [13]. Similarly, other researchers have demonstrated that a user's interactions can be used to infer parameters of analytical models, which can then be presented visually [4, 15, 44]. Our work shares synergistic activities with these prior works in that we also seek to extract higher-level information from low-level user interactions. However, our goal is to develop techniques that can automatically classify users on their characteristics.

2.2 Inferring Cognitive Traits and Strategies

Much of the existing work in the visual analytics community on connecting the ways users solve problems with their cognitive abilities has been based on eye tracker data [2, 29, 40]. For example, Lu et al. demonstrated how eye gaze data can be used to determine important or interesting areas of renderings and automatically select parameters to improve the usability of a visualization system [29]. Steichen et al. explored the use of eye tracking data to predict visualization and task type [42, 40]. With varying degrees of accuracy they were able to predict: (1) a user's cognitive traits: personality, perceptual speed and visual working memory, (2) the difficulty of the task, and (3) the visualization type. These findings are particularly important for visual analytics tasks as previous research has shown that users' cognitive traits can be used as predictors of speed and accuracy [17, 47]. Although researchers have demonstrated the utility of eye gaze data, its collection is often not suitable for dynamic or interactive systems where what the user sees is not static. Instead of using eye gaze data, in this work we forgo specialized sensors and analyze mouse interactions[1].

Cognitive traits can also be correlated with proficiency in certain domains. For instance, Ziemkiewicz et al. [47], Green and Fisher [17], and Ottley et al. [33] demonstrate a significant correlation between the personality trait locus of control (a measure of perceived control over external events) and speed and accuracy on complex visualization tasks. Though more subtle, they also found significant effects with the personality traits extraversion and neuroticism. Other cognitive traits such as perceptual speed [1, 6] and spatial ability [5, 46] have also been shown to affect performance on visual analytics task.

Other types of traits can be used to adapt systems as well. In the HCI community, Gajos et al. developed the SUPPLE system that can learn the type and degree of a user's disability by analyzing mouse interaction data and generate dynamic and personalized interfaces for each specific user [14]. Although the intended scenario is in the domain of accessibility, the approach and methods developed by Gajos et al. can be generalized to other interfaces as well.

In the web usage mining community, researchers have used click stream data for modeling and predicting users' web surfing patterns [12, 25, 26, 39]. Some of the techniques developed for these web mining applications could be adapted to extend work like ours. However, we focus on a more general but complex visual task, and on learning about the users themselves as they complete the task.

---

[1] Recent work suggests that mouse movements in some interfaces are strongly correlated with eye movements [7, 21]

3 EXPERIMENT

To investigate what interaction data encodes about users of a system, we sought a task that would simulate realistic tasks, and be difficult enough that people would have to think about how to solve it (engage strategies). Adopting a large visual search task satisfies our criteria: it is easy to explain to participants, but not easy to do, and it is a basic component of typical visual analytics tasks. Specifically, we chose *Where's Waldo* [19], a famous children's game consisting of illustration spreads in which children are asked to locate the character Waldo. Finding Waldo is not easy thanks to the size of the image, which is large enough to require panning and zooming, and the fact that it is craftily drawn to provide a challenge. However, the target is known and there is a guarantee that the task is possible.

We performed an online experiment, collecting interaction data as our participants searched for Waldo in a large image (for our interface, see Figure 1). While Waldo is attired in a distinct red and white striped pattern (see Figure 1: his full image appears in the panel on the right and his placement in the full spread is shown in inset (a)), he is sometimes hidden behind objects, and the illustrations are filled with distractors specifically designed to mislead the user (e.g., scene elements covered in red and white stripes or the characters shown in Figure 1 insets (b) and (c)). To locate Waldo, users have to visually filter unimportant data, making him sometimes difficult to find. This difficulty is also analogous to real-life applications of visual search, where the target item may be partly occluded or obscured by objects of similar color, shape or size.

3.1 Task

In the main task, participants were presented with a Where's Waldo poster and were asked to navigate the image by clicking the interface's control bar (Figure 1). The control bar was designed to resemble Google Maps' interface and afforded six interactions: *zoom in*, *zoom out*, *pan left*, *pan right*, *pan up* and *pan down*. However, unlike Google Maps, our interface does not allow dragging, rather all actions occur through mouse clicks only.

The zoom levels for the interface range from 1 to 7 (level 1 being no zoom and level 7 being the highest magnification possible). The full image has resolution 5646 by 3607 pixels. At zoom level 1, the full image is shown. At zoom level k, the user sees proportion $1/k$ of the image. Panning moves the display by increments of $1/2k$ pixels.

The interface also includes two buttons not used for navigation: *Found* and *Quit*. When the target is found, the participant is instructed to first click *Found* then click on the target. The user must then confirm the submission on a pop-up alert. We require multiple clicks to indicate Waldo has been found to encourage participants to actively search for the target instead of repeatedly testing many random guesses. If the participant clicks *Found* but does not click on the correct location of Waldo, the click is logged, but nothing happens visually. Unless the participant quits the application, the experiment does not terminate until Waldo is found correctly.

3.2 Data Collection

For our analysis, we recorded as much mouse activity as possible, including both mouse click and mouse move events. Mouse click events on interface buttons were logged with a record of the specific button pressed and a time stamp. Similarly, we recorded the interface coordinates of the mouse cursor and the timestamp for every mouse move event.

To establish labels for our machine learning analysis of performance outcomes and personality traits, we recorded both completion time and personality survey scores for each participant. Because researchers have shown [17, 33, 47] that the personality factors locus of control (LOC), a measure of perceived control over external events, and neuroticism and extraversion are correlated with performance on complex visualization tasks, the survey was chosen to collect those traits. Specifically, we use a twenty-seven-question survey which includes the Locus of Control (LOC) Inventory (five questions) [16] and the Big Five Personality Inventory (twenty questions) [10] intermingled. The survey also includes two attention checks which require

participants to give an obvious and precise response. These were used to filter participants who did not pay attention while completing the survey.

3.3 Participants

We recruited online unpaid volunteers, totaling 118 who successfully completed the task by finding Waldo, of whom 90 successfully completed a personality survey and passed an attention check. Women comprised 39 percent, and men 61 percent. Each participant used his or her own computer and completed the task via the Internet. They were required to have basic computer skills and to have never seen the poster in the experiment before. The participants had a median education level of a master's degree. Ages range from 18 to 45 ($\mu = 24$ and $\sigma = 2.8$). Average task completion time was 469.5 seconds ($\sigma = 351.9$).

3.4 Procedure

Participants were first asked to complete the personality surveys by rating a series of Likert scale questions on a scale of 1 (strongly disagree) to 5 (strongly agree). Next, participants read the instructions for the main portion of the experiment and were shown the main interface (Figure 1). They were instructed to manipulate the image by using the six buttons on the control bar to find Waldo using as much time as needed and told their completion time would be recorded. Once they had successfully found the target, they completed a basic demographic survey.

4 HYPOTHESES

We collected data at the lowest possible level to ensure that we captured as much information about the participants' analysis process as possible. Over the next four sections we discuss how we first visualize this data, then create encodings to capture different aspects of the participants' interactions based on three core aspects of visual analytics: data, user, and interface. Specifically we encode (1) the portion of the data being displayed, as high-level changes in program state, (2) low-level user interactions, in the form of complete mouse-event data, and (3) interface-level interactions, as sequences of button clicks on the interface's controls. We analyze our data with these encodings with machine learning to evaluate the following hypotheses. First, we hypothesize that participants who are quick at completing the task employ different search strategies from those who are slow, and that these differences are encoded in a recoverable way in the interactions; second, that we can analytically differentiate users' personality traits based on interactions; and third, that these differentiations can be detected without collecting data for the entire timespan of the task, but instead can be found using a fraction of the observation time.

5 VISUALIZING USER INTERACTIONS

To explore our hypothesis that we can detect strategies employed by different groups of participants, we first visualize their interactions. Figure 2 shows example visualizations of user movement around the Waldo image. The area of the visualization maps to the Waldo image. Each elbow-shaped line segment represents a transition from one user view of the image to another, i.e. from a view centered on one endpoint of the line to the other. Where these lines intersect with common end-points are viewpoints of the image experienced by the participant while panning and zooming. The lines are bowed (elbow shaped) to show the direction of movement from one viewpoint to the next. Lines curving below their endpoints indicate movement toward the left, and those curving above indicate movement to the right. Bowing to the right of the viewpoints indicates movement toward the bottom, and bowing left indicates movement toward the top.

Zoom levels of viewpoints are not explicitly encoded, but the set of possible center points is determined by the zoom level. High zoom levels mean center points are closer together, so shorter-length lines in the visualization indicate the user was exploring while zoomed in. Note that diagonal movement through the Waldo image is not possible directly with the participants' controls. Instead, diagonal lines in the

(a) Slow

(b) Fast
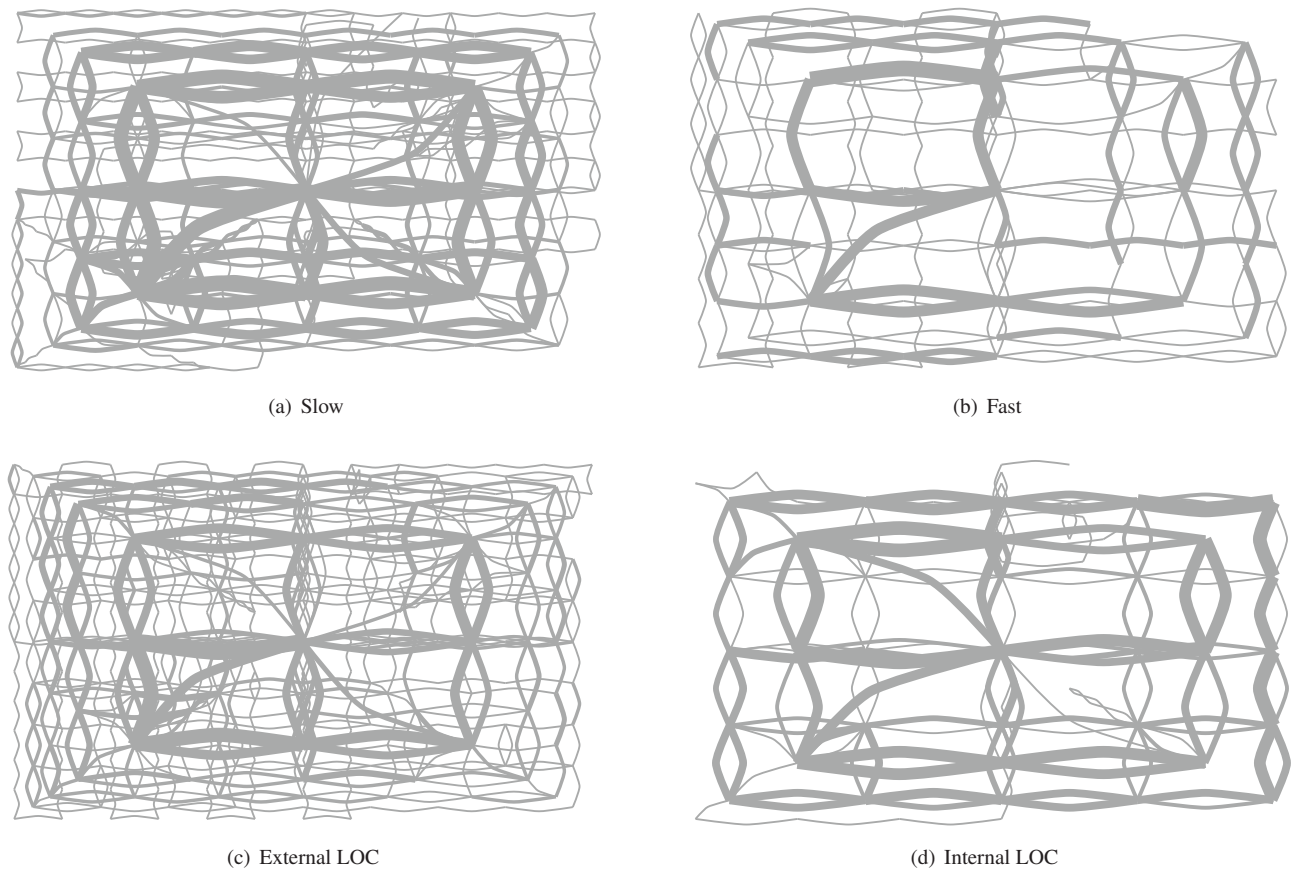
(c) External LOC

(d) Internal LOC

Fig. 2. Visualizations of transitions between viewpoints seen by participants during the study (see Section 5). Subfigures (a) and (b) show slow and fast users respectively, as determined by the mean_nomed splitting method (see Section 6). Subfigures (c) and (d) are split with the mean_nomed method based on locus of control, a personality measure of a person's perceived control over external events on a scale from externally controlled to internally controlled.

visualization are created because of zooming, i.e. when zooming out requires a shift in the center point.

This visualization can be used to show the movements made by a whole group of users by counting, for each flow line, the number of users who made the transition between the two corresponding viewpoints in the correct direction. In our examples, we are showing such aggregations for four different groups of users. In each case, the thickness of the lines encodes how many users in the group made that transition.

The top two sub-figures of Figure 2 compare users who were fast versus slow at completing the task. Users were considered fast if their completion time was more than one standard deviation lower than the mean completion time, and correspondingly considered slow with completion times more than one standard deviation above the mean (for further explanation see Section 6). Users who were slow produce a finer-grain set of lines, indicating they made more small movements through the image using a higher zoom level and saw more of the Waldo image in higher detail. Further, the extra lines in the lower left of Figure 2 (a) as compared to Figure 2 (b) suggest that these slower participants were led astray by the distractors in the image, e.g. the people wearing similar clothing to Waldo seen in Figure 1, insets (b) and (c).

Evidence of different strategies is also salient when visualizing results based on some personality factors. The personality trait locus of control (LOC) has been shown to affect interaction with visualization systems [17, 33, 47]. Figures 2 (c) and (d) visualize differences between participants with external (low) versus internal (high) LOC. In these subfigures, we see that the external group zoomed in much further on average, while the internal group performed more like the fast group and was able to find Waldo with a smaller set of viewpoints.

These observations are readily seen through these visualizations, but cannot be seen from inspection of the data, nor from machine learning results. Encouragingly, these visualizations hint that there are patterns to uncover in the data. The rest of this work explains our analytical results in extracting them automatically with machine learning.

6 COMPLETION TIME FINDINGS

In Section 5, we presented visual evidence that our collected interaction data encodes differences between groups of participants. However, being able to tell fast users from slow is more useful if it can be done automatically. In this section, we delve into the data with analytical methods, using machine learning to build predictors of task performance outcomes. In particular, we adopt two common machine learning algorithms, decision trees [31], which learn hierarchical sets of rules for differentiating data, and support vector machines (SVMs) [20], which learn hyperplanes that separate data points of different classes in the data space. We apply these, to three representations of the interaction data, created to capture different aspects of how users interacted with the system.

Specifically, we tested three categories of representations of the participants' interactions, corresponding to some core aspects of visual analytics (data, user, and interface): the views of the image data participants encountered during their task (state-based), their low-level mouse events (event-based), and their clicks on interface controls (sequence-based). In this section we briefly explain how we derive the

Table 1. Completion Time Classifiers - results for state space, edge space and mouse events were achieved using support vector machines. The n-gram space results use decision trees. These results were calculated using leave-one-out cross validation.

Data Representation	Class Split	Accuracy (%)
state space	mean_nomed	83
	mean	79
edge space	mean_nomed	83
	mean	63
mouse events	mean_nomed	79
	mean	62
n-gram space	mean_nomed	79
	mean	77

Table 2. Features calculated for SVM analysis of mouse movement and raw mouse click data. μ, σ, and μ'_3 refer to the mean, standard deviation, and third statistical moment. Pairwise indicates functions of pairs of consecutive events.

Click Event Features	Move Event Features
Clicks per second	Movements per second
Avg. time between clicks	Pairwise Euclidean distance (μ, σ, μ'_3)
% Left, %Right	Pairwise x distance (μ, σ, μ'_3)
% Up, % Down	Pairwise y distance (μ, σ, μ'_3)
% Zoom in, % Zoom out	Pairwise speed (μ, σ, μ'_3)
% Found, % Quit	Pairwise angle (μ, σ, μ'_3)
% Clicks on Image	

target participant groups used for our machine learning results, then show, for each representation of the data, our results at predicting if a given user would be fast or slow in completing the task.

We establish two different methods for labelling our participants based on the collected data. Our analyses aim to classify participants into discrete classes, fast and slow, but our recorded data includes only each participant's actual completion time. The first discretization method is to apply the mean completion time (469.5 seconds) as a splitting point: participants with a completion time lower than the mean are assigned to the 'fast' group, and higher to 'slow'. Participants with scores exactly equal to the mean are excluded from the data. In our results, this splitting method is indicated as *mean*. In the second method, we assume that participants whose scores are within one standard deviation of the mean have 'average' performance and we exclude them from the study, labelling the rest as above. We refer to this approach as the 'no-medium' splitting method, indicated in results tables as *mean_nomed*. The no-medium method allows us to see that stronger patterns emerge for participants with more extreme performance.

6.1 State-Based Analysis

In the visualization of participants' movement through the Waldo image (see Section 5), differences across groups of participants in how they examine the data become salient. This discovery would be more broadly applicable if the differences could be determined automatically. We create two data representations emulating these visual forms to search for patterns that differentiate users based on what parts of the image they chose to look at. In the "state space" encoding, we capture the portion of the data viewed as each participant navigated the Waldo image. In the "edge space" encoding, we capture transitions participants made between viewpoints of the image. Applying support vector machines (SVM) yields high-accuracy classifiers of completion time with both representations.

The state space encoding can be represented by a vector space. We consider the set $s \in S$ of all visual states (given by view position and zoom) that were observed by any user during completing the task. We create a set of vectors u_i, one representing each user, such that $u_i = (count_i(s_1), count_i(s_2), \ldots, count_i(s_{|S|}))$, where $count_i(s_j)$ indicates the number of times user i landed on state j. For the data from the Waldo task, this process yields a vector space in 364 dimensions.

A similar vector space expresses the transitions between viewpoints of the visualization, encoding how participants moved the viewpoint around the image in their search for Waldo. Their strategies may be encapsulated by how they directed the view during their search. In this vector space, the set $t \in T$ consists of all transitions made between any viewpoints by any participant while completing the task. If each viewpoint is represented by the location of its center, x, then $T = \{(k, m)\}$ where any participant made the transition $x_k \rightarrow x_m$ from position x_k to position x_m while searching for Waldo. Each individual user's vector is constructed as $v_i = (count_i(t_1), count_i(t_2), \ldots, count_i(t_{|T|}))$, where $count_i(t_j)$ indicates the number of times user i made transition t_j. The dimensionality of our derived transition-based vector space (edge space) is 1134. The zoom levels are not explicitly encoded, but the set

of possible center points is determined by the zoom level. This feature space is most closely related to the visualization described in Section 5 and seen in Figure 2.

The calculated vectors are used as a set of data features for input to an SVM [43], a widely-applied machine learning method that works on vector space data. SVMs are both powerful and generic, and work by discovering an optimal hyperplane to separate the data by class. For this work we focus on results from the default implementation in the machine learning software package Weka [18], which means a linear hyperplane, and slack parameter $c = 1$. This choice of an out-of-the-box classifier is intended to demonstrate that these results can be achieved in a straightforward manner.

Table 1 shows the accuracy of our completion time predictions, calculated via leave-one-out cross validation. Both state and edge space provide strong completion-time prediction results, with maximum accuracies of 83%. However, these classifiers can only take into account high-level changes in the software, as opposed to the lower-level physical actions that may characterize different participants, which leads us to investigate different encodings for further analyses.

6.2 Event-Based Analysis

Users move their mouse throughout the process of working with a visual analytic system. Sometimes they move the mouse purposefully, e.g. to click on a control, other times they hover over regions of interest, and sometimes they make idle movements. Where the state and edge space encodings fail to make use of this information, the event-based data encoding described in this section derives from the most raw interaction information available to capture innate behavioral differences.

Previous machine learning work has shown that mouse event data contains enough information to re-authenticate users for security purposes [30, 36, 37, 45]. We adapted the data representation of Pusara et al. [36] for our interaction data by calculating their set of statistics over event information. Because we are predicting completion time, we removed any statistics that we found to be correlated with completion time. Table 2 shows the set of functions we used to encapsulate the participants' mouse movements and raw clicks. This set includes statistics on click information (number of clicks and time between clicks), raw button click information (percentage of clicks on a particular button, e.g., "% Left" refers to the percentage of button clicks on the "Pan Left" button), and derived mouse movement information (such as the number of moves, and the mean, standard deviation and third statistical moment of the distance and angle between them). The set does not include total counts of clicks on different buttons or the total number of mouse movement events, because those were strongly correlated with the total completion time. In total, we use twenty-seven features, listed across the two columns of Table 2.

As with the state-space representations, we apply SVMs to the mouse-event data. Table 1 shows the accuracy achieved with the mouse-event data using SVM classifiers, calculated using leave-one-out cross-validation. This approach manages a maximum score of 79%, which shows that there is strong signal in this low-level mouse data. The input features may reflect subconscious mouse movement habits more than actual intended actions, so the results indicate that

the differences between populations may be driven by innate differences in approach or cognitive traits. Even though none of the features is individually correlated with the completion time, these low-level interaction statistics taken together are enough to analytically separate fast from slow users.

6.3 Sequence-Based Analysis

The most direct representation of a user's process may be the sequence of direct interactions with software. Clicks are conscious actions that represent the user's intentions, and thus building classifiers based only on these semantically relevant interactions may provide more insight into why and how participants' analytical strategies differ. For our sequence-based analysis, we examine the sequences of button clicks used by participants to achieve the task of finding Waldo. We connect n-grams, a method from information retrieval for extracting short subsequences of words from collections of documents, to decision trees, a class of machine learning algorithms that produces human-readable classifiers.

6.3.1 N-Grams and Decision Trees

The n-gram method from information retrieval is intended for text, so an n-gram feature space must start with a string representation of data. We assign a unique symbol to each of the seven buttons in the interface: 'L' for pan left, 'R' for right, 'U' for up, 'D' for down, 'I' for zoom in, 'O' for out, and 'F' for declaring Waldo found. Each participant's total interactions are thus given by an ordered string of symbols. We derive an n-gram vector space by considering each symbol a word, and each participant's sequence of words a document. Each dimension in the vector space then corresponds to one n-gram (i.e. one short sequence of user actions). Participants are represented by a vector of counts of the appearances of each n-gram in their interaction strings.

In our analyses we apply the J48 decision tree algorithm and NGramTokenizer from Weka [18] to classify participants based on task performance, and report accuracy scores from leave-one-out cross validation. The effectiveness of n-grams is sensitive to the choice of n. We empirically chose a combination of 2- and 3-grams as we found that to best balance accuracy and expressiveness of our eventual analytic output. Our results at classifying participants on completion time are shown in Table 1, revealing a top accuracy of 77% calculated with leave-one-out cross validation.

6.3.2 Decision Tree Interpretation

One advantage to using a decision tree with n-grams is that the resulting classifier is human-readable. Figure 3 shows the decision tree produced for the completion time data in n-gram space, using a mean split for classes. Each internal node shows a sequence of button clicks and the branches are labeled with the number of occurrences needed of that n-gram to take that branch. We can make several inferences about strategy from this tree. The root node indicates the strongest splitting criteria for the data. In this case, that node contains "L D", the n-gram corresponding to a participant clicking "Pan Left" then "Pan Down". If that sequence was clicked more than three times by anyone, that indicated the person would finish slowly. This makes sense because Waldo is in the upper right of the image. Moving in the wrong direction too many times can be expected to slow down progress at the task.

The "F U" and "D F R" nodes are also revealing. The "F" corresponds to telling the program that Waldo is found. These "F" button presses are not the last action, meaning they do not correspond to correctly finding Waldo. Instead, these sequences show participants' false guesses. Thus the tree suggests that participants who made several false guesses finished the task more slowly.

Finally, the "O O I" and "L O I" nodes correspond to behavior where the participant zoomed out and then back in again. The "O I" component could indicate participants zooming out to gain context before zooming in again. Alternatively, the same subsequence could indicate participants zooming out and immediately back in, wasting time.
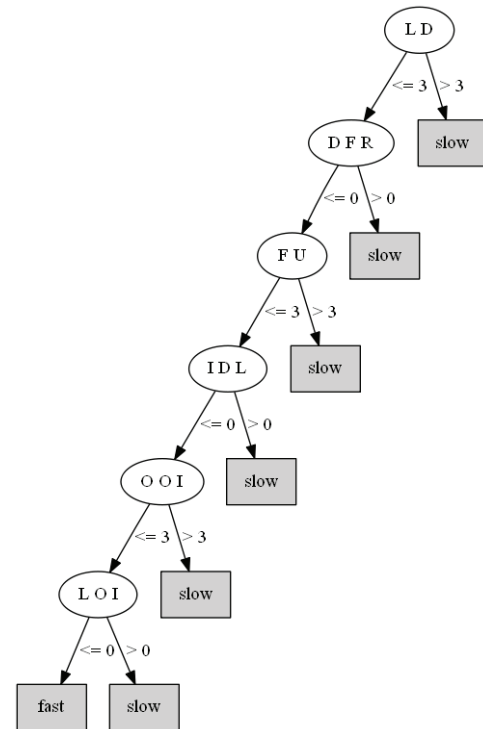

Fig. 3. This is the decision tree generated as a classifier for fast versus slow completion time with mean class splitting. Each internal node represents an individual decision to be made about a data point. The text within the node is the n-gram used to make the choice, and the labels on the out-edges indicate how to make the choice based on the count for a given data point of the n-gram specified. Leaf nodes indicate that a decision is made and are marked with the decided class.

The readability of this technique shows promise for identifying trends in strategies and behaviors. We cannot guarantee that these interpretations reflect the participants' actual intentions, but rather submit these as possible reasons for what is shown in the tree. The real power of using n-grams and decision trees on interaction sequences is that it makes this type of hypothesizing possible, leading to deeper investigation when it is beneficial to understand how people are solving a given task.

7 PERSONALITY FINDINGS

Prior research by Ziemkiewicz et al. [47] and Green and Fisher [17] suggests that users will use a visualization system differently based on their personality traits. Motivated by these findings, we explore the efficacy of extracting personality traits from interactions. Specifically, we apply the same data encodings and machine learning algorithms used for the completion time analyses to predict users based on their personality traits.

Instead of classes derived from completion time, we separate users into *low* and *high* groups based on their scores on each personality inventory: locus of control, extraversion, agreeableness, conscientiousness, neuroticism and openness to experience. Consistent with our completion time analysis, we test both mean and mean_nomed splits (see Section 6). Table 3 summarizes our analysis results.

Across several techniques, we successfully classified users based on their LOC, neuroticism, and extraversion scores. Of the personality traits, our techniques were best with LOC, yielding classification accuracies as high as 67%. This supports the findings of Ziemkiewicz et al. [47] that of the personality traits, LOC was the strongest predictor of users' performance on visualization search tasks. Consistent with our findings, prior work also found significant effects with neuroticism and extraversion [17, 47].

(a) State Based

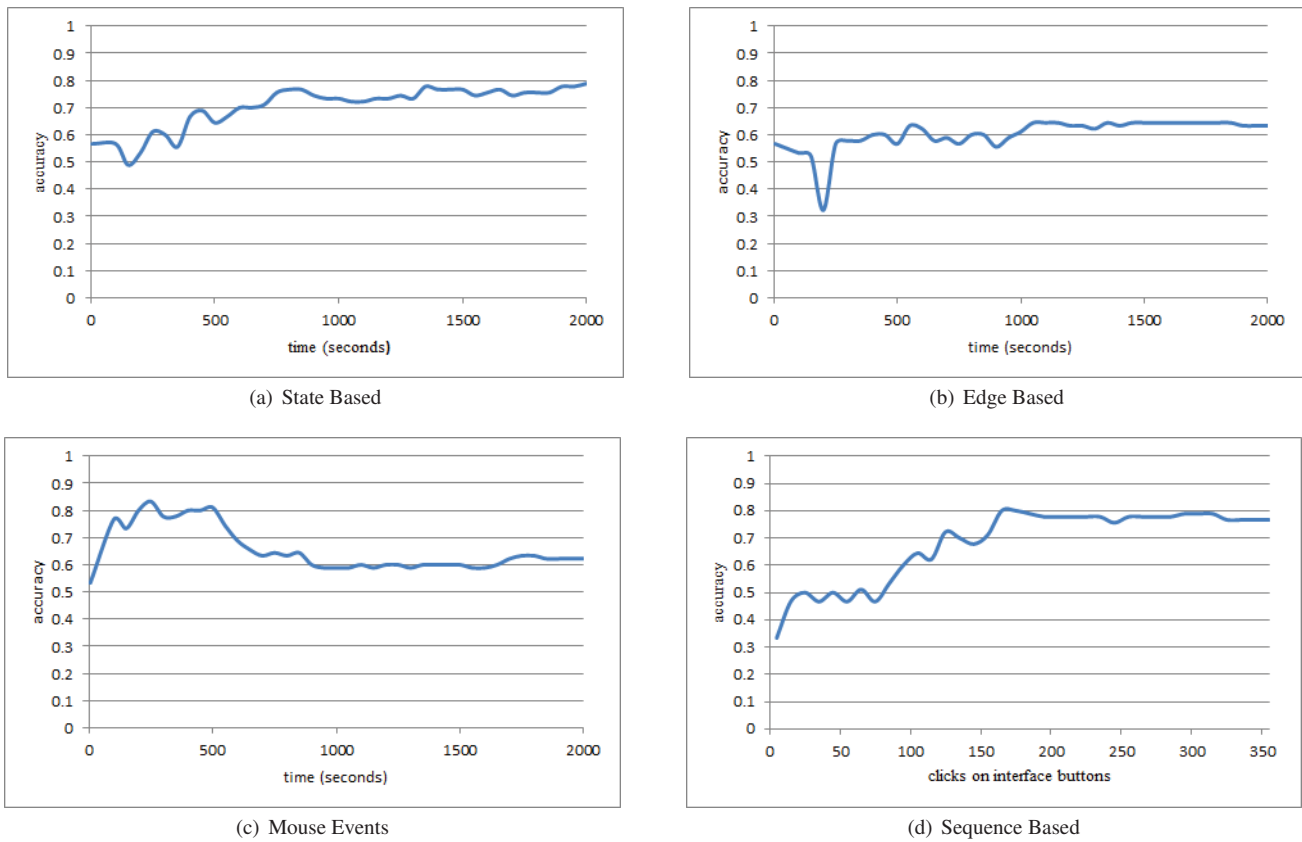(b) Edge Based

(c) Mouse Events

(d) Sequence Based

Fig. 4. Graphs showing the ability to classify participants' completion time as a function of the extent of data collected. The x-axis represents the number of seconds of observation, or number of clicks for the sequence based data. The y-axis is the accuracy achieved after that amount of observation. Accuracy values are calculated with leave-one-out cross validation, and use the mean splitting method (see Section 6).

Table 3. Personality Classifiers - all of these results are with SVM except when using n-grams, which we pair only with decision trees

Data Representation	Class Split	Accuracy (%)
LOC		
n-gram	mean	67
Neuroticism		
mouse events	mean_nomed	62
edge space	mean_nomed	64
Extraversion		
edge space	mean	61

8   LIMITED OBSERVATION TIME

The participants in our study were given as much time as they needed to complete the Waldo task. So far, the presented results have taken advantage of the full timespan of the collected data from their interactions to classify them. Investigating the minimal timespan required for this type of analysis is crucial for potential real-time applications, so we evaluated our classifiers' performance as a function of the data collection time.

Figure 4 shows, for each of the different data representations, graphs of how task performance classification improves (on trend) with more observation time, i.e. more information available. Figure 5 shows one example of this behavior from personality trait classifiers. The x-axis is the amount of data collected, and the y-axis is the accuracy achieved by training and classifying with that amount of data. For all but the sequence-based analysis, the x-axis represents time. For the button click sequences, the x-axis is based on the number of clicks instead. Leave-one-out cross validation (LOOCV) and the mean-based class definition are used for all these results.
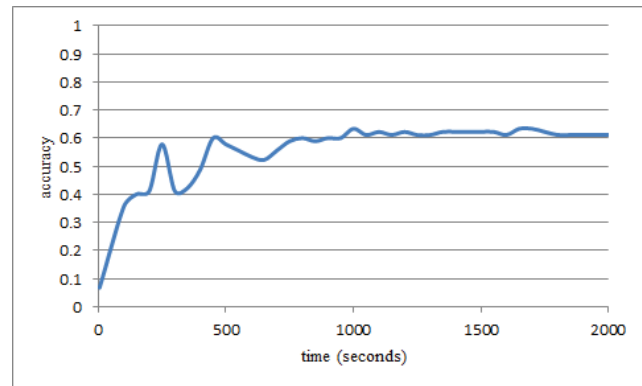
Fig. 5. This graph shows the dependence of the ability to classify the personality trait extraversion on the amount of time the participants are observed. The x-axis represents the number of seconds of observation. The y-axis is the accuracy achieved after that amount of time. This example uses the edge space encoding and the mean splitting method (see Section 6). Accuracy values are calculated with leave-one-out cross validation.

These graphs demonstrate two things. First, accuracy scores comparable to the final score can be achieved with much less than the maximum time. Note that close to the mean completion time, the encodings are achieving much of their eventual accuracy scores: state-based, 64% instead of its eventual 83%; edge-based, 60% compared to 63%; and sequence-based, 61% as opposed to 77%. These correspond to 77%, 95% and 79% of their final accuracy percentage scores, respec-

tively.

Second, as expected, in most cases using more data allows for better results. In the case of the mouse event data, the accuracy peaks before reaching the average participant's finishing time, about 470 seconds.

9 EXTENDED RESULTS

In this work, we focused on machine learning results produced with off-the-shelf algorithms to emphasize that they could be re-applied in a straightforward way. However, in the course of our investigation, we applied a number of additional customizations to find the most accurate classifiers possible with our data representations. These extended results can be found in Appendix A. In the appendix, we explain the additional methods we used and show the results we achieved by customizing the classifiers. We show cases in which our tuning produced higher-accuracy classifiers, and revealed signal with feature spaces or class splitting criteria that otherwise could not encode certain traits.

10 DISCUSSION AND FUTURE WORK

In this work, we have shown, via three encodings, that interaction data can be used to predict performance for real-time systems, and to infer personality traits. Our performance predictions ranged in accuracy from 62% to 83%. On personality traits, we were able to predict locus of control, extraversion, and neuroticism with 61% up to 67% accuracy. Further, we found that with only two minutes of observation, i.e. a quarter of the average task completion time, we can correctly classify participants on performance at up to 95% of the final accuracy.

Given the above results, there are some fascinating implications and opportunities for future work. In this section, we discuss the choice of task and data representations, and how they may be generalized, differences between the personality results versus those for completion time, and future work.

10.1 The Waldo Task and Our Encodings

The Where's Waldo task was chosen because it is a generic visual search task. It is a simple example of an elementary sub-task that comes up often in visual analytics: looking for a needle in a haystack. The user can manipulate the view, in this case with simple controls, and employ strategy to meet a specific task objective. In this section we address how this experiment and our analyses may scale to other systems. Because our set of encodings is based on three core aspects of visual analytics, data, user, and interface, we frame the extensibility of our approach in terms of data and interface.

The data in our experiment is the image in which participants search for Waldo. At a data scale of twenty-megapixels, our state-based interaction encodings, which are closely tied to the data because they capture what parts of the image a participant sees, reach hundreds of features to over 1000 features. As the size of the data (image) increases, the state space and edge space may not scale. However, the event-based and sequence-based encodings depend only on the interface, and thus could scale with larger image data.

Conversely, the interface in our experiment is a simple set of seven buttons. Increasing the complexity of the interface affects the event-based and sequence-based encodings. The mouse-event features include statistics about how often each button is pressed, and the sequence-based encoding requires a different symbol for each button. While these two encodings may not be able to scale to meet increased interface complexity, the state-based encoding is unaffected by the interface and thus could scale with the number of controls.

The three encodings we used in this paper can all be extracted from the same interaction logs. Each one of them provides enough information to recover task performance efficiently. Because of their complementary relationships with the core concepts of interface and data, their strength, as an ensemble, at learning from interaction data is not strictly constrained by either interface or data complexity.

The scalability of the ensemble of encodings raises the possibility that our approach could be generalized to other visual search tasks and to complex visual analytics tasks. In particular, since users' interactions in visual analytics tasks have been shown to encode higher-level reasoning [11], we envision that our technique could be applied

to other sub-tasks in visual analytics as well. Specifically, we consider the Waldo task as a form of the data search-and-filter task in the Pirolli and Card Sensemaking Loop [35]. We plan on extending our technique to analyzing user's interactions during other phases of the analytic process such as information foraging, evidence gathering and hypothesis generation.

10.2 Personality

Being able to demonstrate that there is signal in this interaction data that encodes personality factors is exciting. However, none of the results for personality factors are as strong as those for completion time. Not only are the overall accuracy scores lower, but we found that in examining the time-based scores (as in Section 8), for many personality factors, there was not a persistent trend that more data helped the machine learning (Figure 5 shows one of the stronger examples where there is a trend).

While the prediction accuracies are low, our results are consistent with prior findings [40] in the human-computer interaction community on individual differences research. Taken together, this suggests that although personality and cognitive traits can be recovered from users' interactions, the signals can be noisy and inconsistent. In order to better detect these signals, we plan to: (1) explore additional machine learning techniques, like boosting [38] for leveraging multiple learners together, and (2) apply our techniques to examine interactions from more complex visual analytics tasks. We expect the latter to amplify results as Ziemkiewicz et al. [47] have shown that personality trait effects are dampened when the task is simple. In their work, for complex inferential tasks, the effects were more pronounced and potentially easier to detect [47].

10.3 Future Work

This work is a first step in learning about users live from their interactions, and leaves many exciting questions to be answered with further research. The ability to classify users is interesting on its own, but an adaptive system could test the feasibility of applying this type of results in real time. Different cognitive traits may prove more fruitful for adaptation, but even completion time could be used to adapt, by giving new users advice if they start to follow strategies that would lead to their classification as slow.

Further, of the data representations we evaluated, only the mouse events, the lowest-level interactions, encode any information about time passing during the task. The other representations do not encode the time between states or button presses, but that information could be useful for a future study. For our sequence-based analysis, our approach was to pair n-grams with decision trees for readability, but there are plenty of existing treatments of sequence data that remain to be tried for this type of data classification on visual analytic tasks, including sequence alignment algorithms, and random process models, e.g. Markov models. Finally, in this work we focused on encoding one aspect of data or interface at a time, but combining feature spaces could be powerful. In fact, in experimenting with a feature space that leverages multiple types of encodings, we achieved 96% accuracy on completion time with mean_nomed splitting[2].

The experimental task was a simple version of a basic visual analytics sub-task. Our results could be strengthened by expanding the experiment to test Waldo in different locations, or different stimuli like maps with buildings and cars. The breadth of applicability could be evaluated by testing other elementary visual analytics tasks such as using tables to find data or comparing values through visual forms.

Our plans to extend this work expand on three fronts: (1) evaluating additional personal traits, like cognitive factors such as working memory, to our analyses, (2) trying further machine learning algorithms and encodings to learn from more of the information being collected, like the times of the interactions and (3) extending experiments with different tasks including deeper exploration of visual search. We believe

---

[2] Specifically, we tested a modified state space encoding where the zoom level information is replaced by an identifier of the button click that caused the state.

there are many opportunities to extend this work, both experimentally and analytically.

11 CONCLUSION

In this paper, we presented results of an online experiment we conducted where we recorded participants' mouse interactions as they played the game Where's Waldo. We broke the users into groups by how long it took them to find Waldo (completion time) and their personality traits. Visualizing the participants views of the data, we showed that there are differences in strategies across groups of users. We then applied machine learning techniques, and demonstrated that we can accurately classify the participants based on their completion time using multiple representations of their interactions: visualization states, low-level mouse events, and sequences of interface button clicks. By examining artifacts of our machine learning work with these sequences, we were able to identify short subsequences of interactions that identify groups of users. These human-readable classifier results hint at user strategies across groups. We were also able to detect and classify the participants based on some personality factors: locus of control, extraversion, and neuroticism. Finally, we showed the dependence of the machine learning results on the observation time of the participants.

A APPENDIX: EXTENDED RESULTS

Though we demonstrated that completion time and personality can be modeled from raw interactions can be done directly with off-the-shelf tools using default settings, further attention to detail can yield stronger classifiers. In this appendix, we discuss some additional results that we achieved by tuning the algorithms, including applying principal component analysis (PCA), and optimizing the parameters of support vector machines (SVMs).

Table 4. Additional SVM Results - all results are calculated using leave-one-out cross validation.

Data Representation	Class Split	Classifier	Accuracy (%)
Completion Time			
edge space	mean_nomed	SVM_{poly}	87
	mean	SVM_{poly}	72
mouse events	mean_nomed	SVM_{poly}	88
	mean	SVM_{poly}	82
LOC			
edge space	mean	SVM_{poly}	62
state space	mean_nomed	SVM_{poly}	63
state space$_{PCA}$	mean_nomed	SVM	63
Neuroticism			
edge space	mean_nomed	SVM_{poly}	68
state space$_{PCA}$	mean_nomed	SVM	68

The SVM algorithm is sensitive to a slack parameter [8] and to the choice of kernel. Common practice is to address this by using a parameter search to find the best parameter values [20]. In the context of deploying the best possible classifier for a given dataset, that entails simply trying different choices of the parameter (or sets of parameters) and evaluating the classifiers until the best can be reported. Since our goal is instead to evaluate the classifiers and encodings themselves for this type of data, we take the approach of validating the algorithm of classifier+param-search. As usual for cross validation, the data is split into k folds. Each fold takes a turn as the test data, while the other folds are used for training, providing k samples of accuracy to be averaged for a total score. In testing a classifier+param-search algorithm, the algorithm being evaluated on one fold is one that chooses a parameter by testing which value produces the best classification result. To evaluate "best classification result", another (nested) cross validation is needed. The original fold's training data is split into folds again and cross validation is used over those inner folds to pick the optimal parameter. Weka implements a more sophisticated version of this practice that allows optimizing two parameters at once (generally

referred to as grid search) and uses optimizing heuristics to limit the number of evaluations [34]. We have used this implementation to run a grid search that optimizes over (1) slack parameter and (2) degree of polynomial for kernel (1 or 2, i.e. linear or quadratic). In Table 4, this classifier is called SVM_{poly}. This table shows highlights of the results that we produced with this technique.

Another helpful factor in working with SVMs on high-dimensional data is principal component analysis. PCA projects the high-dimensional data into a lower-dimensional space defined by the eigenvectors of the original data. The number of eigenvectors is chosen to make sure that 95% of the variance in the data is accounted for in the low-dimensional approximation[3]. Applying PCA to the data space was particularly helpful in data representations like state space, which has a high degree of dimensionality. In Table 4, data representations with PCA applied are indicated by the subscript PCA.

Overall, the results in Table 4 show cases in which our tuning produced higher-accuracy classifiers, and revealed signal with feature spaces or class splitting criteria that otherwise could not encode certain traits. The completion time results for the edge space and mouse event feature spaces are improvements of up to 32%. Specifically with edge space encoding and mean split, SVM_{poly} offers 82% accuracy instead of 62% with off-the-shelf SVM. In our earlier analyses, we did not find sufficient signal to report on LOC with any state-based encodings, but using PCA or parameter search makes that possible. Through applying standard methods for tuning SVM, we gained higher accuracy over our existing results, and demonstrated connections between encodings and traits that were otherwise obscured.

REFERENCES

[1] B. Allen. Individual differences and the conundrums of user-centered design: Two experiments. *Journal of the american society for information science*, 51(6):508–520, 2000.

[2] G. Andrienko, N. Andrienko, M. Burch, and D. Weiskopf. Visual analytics methodology for eye movement studies. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 18(12):2889–2898, 2012.

[3] L. Bavoil, S. Callahan, P. Crossno, J. Freire, C. Scheidegger, C. Silva, and H. Vo. VisTrails: enabling interactive multiple-view visualizations. In *Proceedings of the Annual Conference on Visualization*, pages 135–142. IEEE, Oct 2005.

[4] E. T. Brown, J. Liu, C. E. Brodley, and R. Chang. Dis-function: Learning distance functions interactively. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 83–92. IEEE, 2012.

[5] C. Chen. Individual differences in a spatial-semantic virtual environment. *Journal of the American Society for Information Science*, 51(6):529–542, 2000.

[6] C. Conati and H. Maclaren. Exploring the role of individual differences in information visualization. In *Proceedings of the working conference on advanced visual interfaces*, pages 199–206. ACM, 2008.

[7] L. Cooke. Is the mouse a "poor man's eye tracker"? In *Annual Conference-Society for Technical Communication*, volume 53, page 252, 2006.

[8] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[9] P. Cowley, L. Nowell, and J. Scholtz. Glass box: An instrumented infrastructure for supporting human interaction with information. In *Proceedings of the Thirty-Eighth Annual Hawaii International Conference on System Sciences*, pages 296c–296c. IEEE, 2005.

[10] M. B. Donnellan, F. L. Oswald, B. M. Baird, and R. E. Lucas. The mini-ipip scales: tiny-yet-effective measures of the big five factors of personality. *Psychological assessment*, 18(2):192, 2006.

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

[12] M. Eirinaki and M. Vazirgiannis. Web mining for web personalization. *Transactions on Internet Technology (TOIT)*, 3(1):1–27, 2003.

[13] A. Endert, P. Fiaux, and C. North. Semantic interaction for visual text analytics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 473–482. ACM, 2012.

---

[3]We used the Weka filter implementation with this option enabled.

[14] K. Gajos and D. S. Weld. Supple: automatically generating user interfaces. In *Proceedings of the Ninth International Conference on Intelligent User Interfaces*, pages 93–100. ACM, 2004.

[15] S. Garg, J. E. Nam, I. Ramakrishnan, and K. Mueller. Model-driven visual analytics. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 19–26. IEEE, 2008.

[16] L. R. Goldberg, J. A. Johnson, H. W. Eber, R. Hogan, M. C. Ashton, C. R. Cloninger, and H. G. Gough. The international personality item pool and the future of public-domain personality measures. *Journal of Research in Personality*, 40(1):84–96, 2006.

[17] T. M. Green and B. Fisher. Towards the personal equation of interaction: The impact of personality factors on visual analytics interface interaction. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 203–210. IEEE, 2010.

[18] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[19] M. Handford. *Where's Waldo?* Little, Brown Boston, 1987.

[20] C.-W. Hsu, C.-C. Chang, and C.-J. Lin. A practical guide to support vector classification. Technical report, National Taiwan University, Taipei 106, Taiwan, April 2010.

[21] J. Huang, R. White, and G. Buscher. User see, user point: gaze and cursor alignment in web search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 1341–1350, New York, NY, USA, 2012. ACM.

[22] R. J. Jacob, J. J. Leggett, B. A. Myers, and R. Pausch. Interaction styles and input/output devices. *Behaviour & Information Technology*, 12(2):69–79, 1993.

[23] N. Kadivar, V. Chen, D. Dunsmuir, E. Lee, C. Qian, J. Dill, C. Shaw, and R. Woodbury. Capturing and supporting the analysis process. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 131–138. IEEE, 2009.

[24] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. Visual analytics: Definition, process, and challenges. In *Proceedings of the IEEE Conference on Information Visualization*, pages 154–175. Springer Berlin Heidelberg, 2008.

[25] P. Kolari and A. Joshi. Web mining: Research and practice. *Computing in science & engineering*, 6(4):49–53, 2004.

[26] R. Kosala and H. Blockeel. Web mining research: A survey. *ACM SIGKDD Explorations Newsletter*, 2(1):1–15, 2000.

[27] T. Lane and C. E. Brodley. Temporal sequence learning and data reduction for anomaly detection. *ACM Transactions on Information and System Security*, 2(3):295–331, Aug. 1999.

[28] B. Lee, P. Isenberg, N. Riche, and S. Carpendale. Beyond mouse and keyboard: Expanding design considerations for information visualization interactions. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 18(12):2689–2698, Dec 2012.

[29] A. Lu, R. Maciejewski, and D. S. Ebert. Volume composition and evaluation using eye-tracking data. *ACM Transactions on Applied Perception (TAP)*, 7(1):4, 2010.

[30] R. Maxion. Masquerade detection using enriched command lines. In *Proceedings of the IEEE International Conference on Dependable Systems and Networks*, pages 5–14. IEEE, 2003.

[31] T. M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[32] C. North, R. Chang, A. Endert, W. Dou, R. May, B. Pike, and G. Fink. Analytic provenance: process+ interaction+ insight. In *Proceedings of the 2011 Annual Conference Extended Abstracts on Human Factors in Computing Systems*, pages 33–36. ACM, 2011.

[33] A. Ottley, R. J. Crouser, C. Ziemkiewicz, and R. Chang. Manipulating and controlling for personality effects on visualization tasks. *SAGE Publications Information Visualization*, 2013.

[34] B. Pfahringer, G. Holmes, and fracpete. Class gridsearch, revision 9733. http://weka.sourceforge.net/doc.stable/weka/classifiers/meta/GridSearch.html, 2014.

[35] P. Pirolli and S. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of the International Conference on Intelligence Analysis*, volume 5, pages 2–4, 2005.

[36] M. Pusara and C. Brodley. User re-authentication via mouse movements. In *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, pages 1–8. ACM, 2004.

[37] R. Reeder and R. Maxion. User interface defect detection by hesitation

[38] R. E. Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

[39] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan. Web usage mining: Discovery and applications of usage patterns from web data. *ACM SIGKDD Explorations Newsletter*, 1(2):12–23, 2000.

[40] 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*, pages 317–328. ACM, 2013.

[41] J. J. Thomas and K. A. Cook. *Illuminating the path: The research and development agenda for visual analytics*. IEEE Computer Society Press, 2005.

[42] D. Toker, C. Conati, B. Steichen, and G. Carenini. Individual user characteristics and information visualization: Connecting the dots through eye tracking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 295–304. ACM, 2013.

[43] V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[44] L. Xiao, J. Gerth, and P. Hanrahan. Enhancing visual analysis of network traffic using a knowledge representation. In *Proceedings of the IEEE Symposium On Visual Analytics Science And Technology (VAST)*, pages 107–114. IEEE, 2006.

[45] Y. Yu. Anomaly detection of masqueraders based upon typing biometrics and probabilistic neural network. *Journal of Computing Sciences in Colleges*, 25(5):147–153, 2010.

[46] C. Ziemkiewicz and R. Kosara. Preconceptions and individual differences in understanding visual metaphors. In *Computer Graphics Forum*, volume 28, pages 911–918. Wiley Online Library, 2009.

[47] C. Ziemkiewicz, A. Ottley, R. J. Crouser, A. R. Yauilla, S. L. Su, W. Ribarsky, and R. Chang. How visualization layout relates to locus of control and other personality factors. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 19(7):1109–1121, 2013.

analysis. In *International Conference on Dependable Systems and Networks (DSN)*, pages 61–72. IEEE, 2006.