Visual Exploration of Large Scatter Plot Matrices by Pattern Recommendation based on Eye Tracking

Lin Shao
Graz University of Technology
Graz, Austria
lin.shao@cgv.tugraz.at

Nelson Silva
Graz University of Technology
Graz, Austria
nsilva@cgv.tugraz.at

Eva Eggeling
Fraunhofer Austria
Graz, Austria
eva.eggeling@vc.fraunhofer.at

Tobias Schreck
Graz University of Technology
Graz, Austria
tobias.schreck@cgv.tugraz.at

ABSTRACT
The Scatter Plot Matrix (SPLOM) is a well-known technique for visual analysis of high-dimensional data. However, one problem of large SPLOMs is that typically not all views are potentially relevant to a given analysis task or user. The matrix itself may contain structured patterns across the dimensions, which could interfere with the investigation for unexplored views. We introduce a new concept and prototype implementation for an interactive recommender system supporting the exploration of large SPLOMs based on indirectly obtained user feedback from user eye tracking. Our system records the patterns that are currently under exploration based on gaze times, recommending areas of the SPLOM containing potentially new, unseen patterns for successive exploration. We use an image-based dissimilarity measure to recommend patterns that are visually dissimilar to previously seen ones, to guide the exploration in large SPLOMs. The dynamic exploration process is visualized by an analysis provenance heatmap, which captures the duration on explored and recommended SPLOM areas. We demonstrate our exploration process by a user experiment, showing the indirectly controlled recommender system achieves higher pattern recall as compared to fully interactive navigation using mouse operations.

ACM Classification Keywords
H.1.2 Models and Principles: User/Machine Systems–Human information processing; H.3.3 Information Storage and Retrieval: Information Search and Retrieval–Relevance feedback

Author Keywords
Visual exploration; scatter plots; eye tracking

INTRODUCTION
A current problem of data analysts is the exploration of large amounts of data in a short period of time. The data may contain a high number of dimensions, which increases the number of potentially interesting views. One way to explore high-dimensional data more efficiently is to use Scatter Plot Matrices (SPLOMs), which visualize all pairwise combinations of dimension views in tabular form. However, the exploration in large SPLOMs is a challenging task, as the view space can be very large and interesting views may be overlooked in exploration. The resulting question is how to explore SPLOMs more efficiently to investigate a large variety of interesting views in less time? Quality metrics [4] can be used, which help with the visual exploration of patterns in large spaces of alternative visualizations, like projection views. A drawback of these approaches is that these methods are objective numerical measures of qualities based on specific tasks, e.g., clustering or outlier detection and thus, may fail to reflect the given user interest. To cater for the specific needs of the user, recommender systems including classification may be used to improve search and filter tasks. However, most classification methods require explicit user interactions to work.

Recently, research in eye movement analysis has made advances, in the technology to track eye movement, as well as in the analysis and visualization of eye movement data. Eye tracking so far has been mainly used to evaluate user gazing in conjunction with user interfaces, or to do user interaction. It is a promising research direction to include eye tracking for Visual Analytics approaches as an indirect means to monitor the user and to adapt the interactive analysis process. We believe that the integration of eye tracking into Visual Analytics approaches is reasonable and opens up new possibilities.

We present a novel concept to recommend interesting scatter plot views based on user attention measured by an eye tracker. This idea is inspired by information retrieval approaches and recommender systems, which help to find previously unseen information by explicit user relevance feedback. Therefore, we use an area of interest (AOI)-based metric that identifies scatter plots on the stimulus as AOI, and capture the transitions between AOIs and the gaze times for each AOI. A classifier learns the visual characteristics from previously seen plots and uses a dissimilarity measure to recommend the most visually dissimilar scatter plots for further exploration. Moreover, we conduct a user experiment to demonstrate the usability of our system and show efficiency increase during explorations.

ACM ISBN 978-1-4503-4903-1/17/03 ...$15.00.
http://dx.doi.org/10.1145/3038462.3038463

RELATED WORK

In many areas eye tracking devices are used for analyzing user behavior, e.g., in market research, human computer interaction and visualization research. Our approach combines user attention analysis with interactive learning systems to support exploratory analysis tasks.

Eye Tracking in Visual Analytics

In Human-Computer Interaction, eye movement tracking is commonly used to study usability issues [18]. An introduction to the basics of eye movement is given by Pool and Ball [20]. They report about key aspects of practical guidance in usability-evaluation studies, and give several statistical metrics that can be derived from eye tracking data and their possible interpretations. Attention heatmaps [6] are often used to show the distribution of the users' attention over the display space when performing tasks. These heatmaps are useful to visualize fixation counts and gaze times on a point-based level. Our work follows an AOI-based metric, where each scatter plot is an AOI. The usage of eye tracking in typical analysis tasks like detecting clusters or correlations is addressed by Etemadpour et al. [10]. Holmqvist et al. [16] have investigated different methods and measures for analyzing user attention patterns. An overview of visualization techniques and methods for analyzing eye movement data are given in [2, 5]. The use of eye movement as an input mechanism to steer a system is considered in [17, 20]. The choice of using eye movement tracking instead of point-and-click interaction allows a non-invasive and continuous tracking of the users. By using eye movement analysis, a fast and continuous tracking of the user interests in real time is possible. This allows for example, the detection of moments of confusion, indecision and high interest regions [11]. A previous study [12] discusses important links between cognitive processes and eye movements. Here, the potential of using eye movements as a performance measure is debated, and several examples of possible inferences that can be made using eye tracking are given. Finally, there are guidelines for adjusting the user interface design to improve the accuracy of eye tracking.

In our work, we use eye movement tracking to identify the user's interests in exploratory analysis tasks by extracting the eye gaze path on the user interface.

Active Learning and Recommending

In information retrieval, learning methods are often used to classify text documents automatically and have shown to be effective [22]. These methods derive models from a set of pre-classified documents and make use of the characteristics of the categories to label previously unseen data. For instance, in [7], classifiers are used for filtering and monitoring text data streams from Twitter. Heimerl et al. [14] compared three different approaches to interactively train a SVM text classifier – basic learning method, visual method and user-driven method. Furthermore, Visual Analytics tools are available, which help to train improved classifiers. Höferlin et al. [15] extended their active learning system with a Visual Analytics process to define filters by ad-hoc training classifiers in the domain of Video Visual Analytics. In [8], visual summaries of misclassified documents are presented to improve the feature ideation and creation of the classifier. Besides text documents, learning systems can also be applied to other research areas such as content-based recommender systems. Behrisch et al. [3] presented a recommender system for scatter plot exploration, which uses a classifier based on Scagnostics [24] features. Moreover, surveys of existing recommender systems and possible extensions concerning recommendation capabilities are given in [1, 19]. However, user interactions like labeling, selecting or rating a test data set are required to train the classifier. Our recommendation system makes use of the advantage of eye movement analysis and transfers the information about the user attention directly to a k-nearest neighbor recommender.

OVERVIEW OF OUR APPROACH

In this section, we present our recommendation concept and indicate how we indirectly integrate guided visual dissimilarity search to support the exploration in large SPLOMs.

Recommendation Concept

Our approach aims to support exploratory analysis tasks in SPLOMs by suggesting interesting and previously unseen scatter plots. We define interestingness in terms of unseen and visually dissimilar to previously explored patterns. The basic idea is to help the user explore large view spaces as efficiently as possible, avoiding to get lost in details and miss relevant patterns. Thus, the user may focus on underexplored areas and explore the SPLOM in a different way. To accomplish this, we use an eye tracker that automatically records user attentions during the exploration, and supply a k-nearest neighbor (k-NN) recommender with the characteristics of explored scatter plots. In our concept, the eye tracker acts as a bridge between data exploration and k-NN recommender, as depicted in Figure 1.

An advantage over previous recommender systems is that we use an eye tracker as intermediary to supply the k-NN recommender with the users' interests. Hence, users can fully concentrate on the exploration task and do not have to provide any kind of relevance feedback to the system. The eye tracker captures the actual users' interest in terms of total gaze duration and number of fixations per scatter plot. Consequently, we determine which scatter plots have been explored for the longest time and were fixated the most. These serve as a basis to determine the most dissimilar plots, for recommending. For instance, a scatter plot with a high number of fixations could indicate that it was used as reference for comparison. We also assume that scatter plots, which attracted the user's attention for a longer time, were of more interest to the user.

Based on a history of previously explored views, we perform a dissimilarity search to the top explored scatter plots –most fixations or longest gaze times– and recommend the most dissimilar ones for exploration. To provide a larger variety of different scatter plots as recommendations, we perform multiple queries from the scatter plots, which called the user's attention the most. Since we are recommending visually dissimilar and unseen scatter plots, our recommendation approach still works for exploratory analysis tasks even if scatter plots with high durations were considered as uninteresting. Our first priority is to support the exploration for undiscovered patterns.

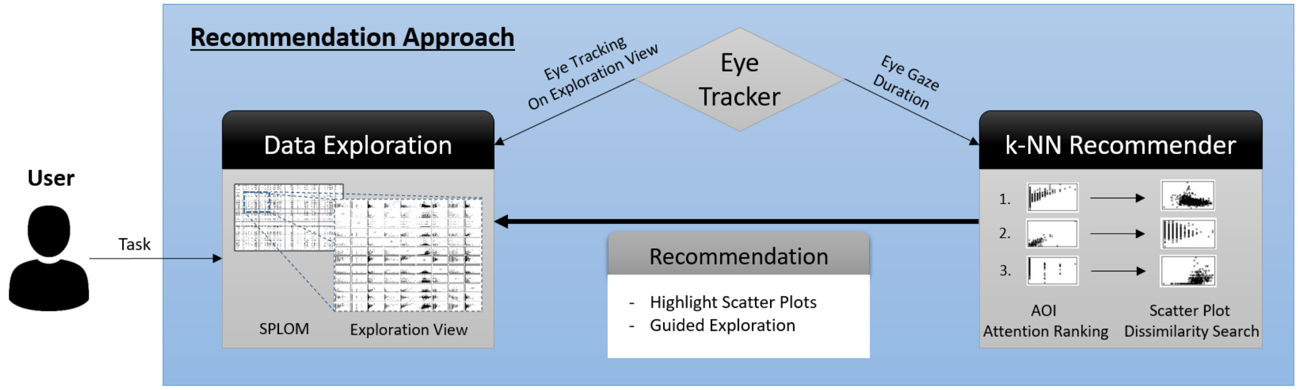


Figure 1. We use an eye tracker as independent intermediary to connect the k-NN recommender with the data exploration interface. Thus, users can focus on the data exploration task while the recommender automatically receives the information from the eye tracker for supporting the exploration.

Finally, the k-NN recommender returns the recommendations in terms of visual feedback or by guiding the exploration to the suggested areas. It is a unidirectional connection between k-NN recommender and data exploration interface, in which the recommender supplies computed information to the exploration interface in real time. Thus, the exploration interface has to wait for input information and transmit the recommendations to the user. To inspect the recommended scatter plots, users can manually change the SPLOM viewport to certain areas by zooming and panning interactions. Alternatively, users can let the system fully control the viewport by using the *Guided Exploration* function. By using a single keystroke, the viewport is changed to the SPLOM area with the most recommendations. The major advantage of this function is that users do not have to decide about when and where to navigate within the matrix. The system automatically detects the areas in the SPLOM with the highest computed interest to the user. Optionally, the system may be set to decide to change the current viewport, once the main patterns visible in a current viewport have been attended by the user (see section below). In this manner, we create a recommender system including navigation guidance, which constantly learns from the user's eye movement and recommends undiscovered scatter plots.

Visual Cluster Analysis

In SPLOMs, zooming and panning functions are needed to investigate individual scatter plots in larger view spaces. However, one problem of these interaction functions is that the user may get lost in the detail of a large view space. Another drawback is that these zoomed-in sections, depending on the data characteristics or matrix ordering employed, may contain visually similar scatter plot patterns. An example is shown in Figure 2 (a) - row 4 and 5. Such areas containing similar plots may expend user attention, which is especially important during early exploration stages. At this point, questions arise concerning when to leave the current exploration viewport and follow the recommendations provided by the system.

To this end, we integrate visual cluster analysis into our system to estimate the number of different scatter plot classes given in a SPLOM, and provide recommendations and navigation guidance based on the exploration degree of the current viewport. In other words, we notify the user to explore further sections as soon as the current viewport is fully explored, i.e., at least one scatter plot of each class is explored. To do this, we cluster scatter plots by their visual similarity and determine the number of various patterns in the user's viewport.

Since image-based descriptors performed quite well for detecting visual similarities in scatter plots [21, 23], we rely on image features based on the density of points and the distribution of edge orientations [25] as implemented in the edge histogram descriptor. Due to performance reasons, we extract the image feature vectors of all scatter plots and apply the DBSCAN clustering algorithm [9] in advance to achieve all classes of similar scatter plot patterns. Finally, we compute the number of distinct scatter plot classes of the current SPLOM viewport and send notifications when all different classes were explored for at least a minimum exploration time of 200 ms. The limit of 200 ms is an important threshold in human visual attention as it is the time needed to initiate an eye movement and therefore to begin a serial attentive mechanism. Usually, the responses obtained in less than 200 ms are considered as preattentive perceptions [13].

EXPLORATION RECOMMENDATIONS VIA EYE TRACKER

Now, we provide implementation details of our prototype system and show how the major components interact together, as shown in Figure 1. Users start their exploratory analysis task on the *Data Exploration* interface, while an eye tracker is recording users attention and feeding the k-NN recommender.

Eye Tracking Integration

We have incorporated a remote eye tracker[1] in our recommender system through the integration of an open source message broker called ActiveMQ[2]. The message broker allows us to exchange messages between more than one client or server application. In this way, all the messages coming from the eye tracker are queued and available for multiple client applications. Both, the k-NN recommender and the user interface are able to consume eye tracking messages in real time.

The eye gaze coordinates are calculated with respect to the screen that the person is looking at, and are represented by a pair of x and y coordinates given on the screen coordinate system. When the system is calibrated, the eye tracker calculates the user's eye gaze coordinates with an average accuracy.

[1] The Eye Tribe: https://theeyetribe.com/.
[2] Apache Software Foundation: http://activemq.apache.org/.

Assuming the user sits approximately 60 cm away from the screen and tracker, this accuracy corresponds to an on-screen average error of 0.5 to 1 cm. The EyeTribe SDK returns both raw and smoothed data coordinates. The smoothed data set contains the coordinates of the estimated on-screen gaze position. To remove high variability in the raw data, the SDK incorporates a gaze data validation algorithm that maintains and analyzes a frame history of gaze frames at run time. Since users typically do not remain stationary for a long period of time and a high number of fast eye gaze jumps can occur, smoothing updates are necessary. In this way, unstable gaze coordinates can be converted to more stable and accurate gaze coordinates. In this work, we use smoothed gaze coordinates.

We use the eye tracker to collect gaze information on the entire screen space and also to detect off screen times. However, we implemented a series of AOI-hit checks that allows us to record the exploration times for when a user is exploring specific scatter plots in the SPLOM. Each scatter plot corresponds to a AOI-hit region that will start a stopwatch timer and add up the fixation counter when it is activated. With this information we are able to distinguish the different AOIs and to calculate statistics on all explored AOIs.

K-Nearest Neighbor Recommender

Our recommender includes a message listener that receives information from the eye tracker in real time. Consequently, we continuously sum up the gaze duration and fixation counts for each scatter plot (AOI), and create a ranked scatter plot list by these values. By default, we rank the scatter plots by their total gaze duration and initiate the recommendation process after a minimum threshold of scatter plots and scatter plot classes (see Section *Visual Cluster Analysis*) have been investigated. Since our recommendation approach is based on the top N explored scatter plots, these minimum thresholds are required to guarantee that the subsequent recommendation process performs well. Firstly, the threshold for the number of minimum scatter plots (τ_{SP}) guarantees that there are at least N queries to perform, and secondly, the other threshold for scatter plot classes (τ_{SPC}) is needed to include class distinction within the queries. For instance, if the first N explored scatter plots are from the same class, we do not initiate the recommendation process, since it would perform equal queries and result in similar recommendations. This is an iterative recommendation process where $N = \tau_{SP}$ and τ_{SP} is for initiating the recommendation process. To not overwhelm the users by giving them too many recommendations, the threshold τ_{SPC} must be fulfilled anew for each iteration to execute the recommendations. In our configuration we set $\tau_{SP} = 10$ and $\tau_{SPC} = 5$. In this case, our recommendation system starts at the earliest after 10 scatter plots and 5 scatter plot classes have been explored. It repeats the recommendation process and will recommend every time unseen scatter plots from various classes when the user explored 10 scatter plots from 5 different classes. Thereby, we create a recommendation loop that constantly recommends visually dissimilar scatter plots and learns about explored scatter plot features.

To recommend dissimilar scatter plots regarding previously seen ones, we use image descriptors based on gradient and density features. We extract edge orientation [25] and density features of the top N explored scatter plots and use these characteristics as feature vector for the dissimilarity measure. For computing density and edge features, we adapt state-of-the-art techniques from previous work [23]. Finally, we use the Euclidean distance to determine the distance between the top N explored scatter plots to the unseen scatter plots and choose scatter plots with the highest distance for each query as recommendation.

Recommendations and Guided Exploration

We provide the users two different ways for inspecting the recommendations, either by manual or automatic navigation to suggested areas in the SPLOM.

Manual Exploration: The system indicates recommendations by highlighting the suggested scatter plots in the SPLOM and by displaying the plots in a ranked view (see Section *System Design*). The exploration interface is waiting for new recommendations and will highlight certain SPLOM indices in the overview representation (*analysis provenance heatmap*) and update the ranked view respectively. By this way of recommendation, we leave the users unrestricted freedom of choice in the further exploration process and just notify them about possibly interesting scatter plots. Thus, users have the option to follow the recommendation system or ignore some suggestions and explore the SPLOM by themselves. A typical mode of operation is that the user may compare the patterns in the current SPLOM viewport with the ranked list of recommended scatter plots, and decide when or if a navigation away from the current viewport is desired. Whatever the user decides, the eye tracker is still recording the exploration process and the system will continue suggesting unseen scatter plots based on the actual user attention.

Guided Exploration: If the user activates the *Guided Exploration* function, the system will automatically guide the user to the most interesting areas of the SPLOM that contains recommended scatter plots. We first compute the degree of interest of the current viewport of the user by taking the visual similarity of scatter plot patterns into account. If the current viewport is fully explored, i.e., all different scatter plot patterns are investigated, the system will guide the user to unexplored areas of the SPLOM. The only requirement is that the threshold τ_{SPC} for new recommendations must be adjusted to the grid of the user's viewport, to guarantee a continuous cycle of recommendations. We use a viewport with a 6×6 grid and a τ_{SPC} threshold of 5. To detect the most interesting area within the SPLOM, we iterate a search over the entire matrix using a sliding window approach to find an interesting viewport, which includes the most suggested AOIs. One suggested AOI increases the interestingness of a viewport to 10, whereas explored areas decrease the interest by 1. Thereby, the navigation decisions will be balanced between a high number of recommended AOIs and less explored AOIs. Consequently, users do not have to care about navigating within the SPLOM, i.e., zooming, panning and selecting interesting areas, and have the highest probability to explore undiscovered scatter plot patterns.

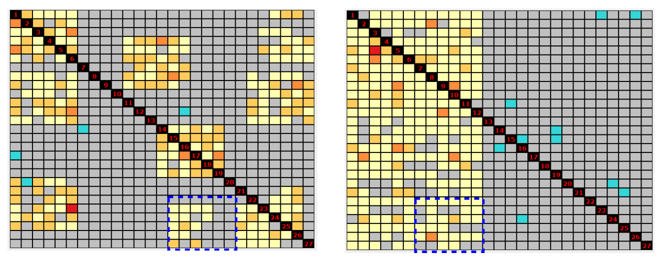
Figure 2. Our scatter plot recommender system consists of (a) the main exploration view (a given viewport of a larger SPLOM), which is monitored by the eye tracker; (b) a overview panel that shows the entire SPLOM including already explored and recommended unseen scatter plots; (c) an exploration detail view with statistics about the user analysis process; (d) ranked history view of explored scatter plots; and (e) recommendation view with suggested new scatter plots for further investigation.

SYSTEM DESIGN

In this section, we introduce the design of our system and explain the usabilities of the individual components. Figure 2 shows the interface of the recommender system.

To identify the sequence of explored scatter plots, we make use of the eye tracker that measures the user's eye gazes on the exploration view - Figure 2 (a). The exploration view displays an enlarged subset of the SPLOM (6×6), which is linked to an AOI-based configuration of the eye tracker. This allows to investigate plots in detail, e.g., for local trends or patterns, and improves the tracking mechanism of explored scatter plots.

To control the exploration view, users can either use the keyboard, apply scroll functions or directly jump to a certain area by using the navigation view - Figure 2 (b). Moreover, the navigation view provides a general overview of the SPLOM by showing the current viewport and visualizes already explored scatter plots and highlights recommendations. The current viewport is visualized by a blue bounding box with dashed lines and is synchronized with the exploration view. A heatmap representation with a sequential colorscale shows the history of explored scatter plots, whereas darker colors indicate longer gaze durations and lighter colors indicate shorter durations (*analysis provenance heatmap*). New recommendations for exploration are emphasized as turquoise rectangles –a complementary color to the heatmap– and allow the user an efficient navigation to the interesting areas of the SPLOM. By this overview representation, we additionally provide a new way of creating data provenance for SPLOM explorations. Specifically, it summarizes all user attentions during the exploration period in one raster image and can be ideally used as reference to other exploration results. By means of the color-coding, users are able to quickly identify areas with highest user attention and relate these across the SPLOM. Furthermore, it generates visual exploration patterns of how users explored the projection space, as shown in Figure 3. One can clearly observe different exploration approaches –guided exploration vs. sequential exploration– by comparing the navigation views.



(a) Guided exploration. (b) Sequential exploration.

Figure 3. Illustration of two different visual patterns on how users explored a SPLOM. (a) the user explored the SPLOM by using our recommender system, one can clearly see the block patterns, which results by jumping to the suggested areas. In contrast, (b) the user explored the SPLOM sequentially with the top-down principle and from left to right.

The exploration detail view, shown in Figure 2 (c), displays additional metadata about the exploration and is also the basis to evaluate our user experiments. More precisely, it shows how many individual scatter plots and scatter plot classes were explored and presents calculated mean values based on the exploration time. By these values, we evaluate the performance of the users and compare the performance under different conditions – with recommendations, without recommendations and with guided explorations.

In our approach, the history data of explored scatter plots plays an important role in creating recommendations. In contrast to feedback-driven recommendation systems, we use quantitative information directly extracted from the eye tracker, such as exploration duration or fixation count of a scatter plot, instead of using binary decisions like yes or no. Hence, we can individually adapt the recommendations and create weighted queries based on the actual user attention. For this reason, we provide a history view that shows all previously explored scatter plots and simultaneously serves as data provenance of our recommendations - Figure 2 (d). Just like the navigation view, all exploration information will be updated in real time. By default, the history view is sorted by exploration duration, and we use the top ranked scatter plots as reference for searching dissimilar recommendations. Based on the exploration task, the ranking priority can be switched to e.g., fixation counts, and thus, changes the references for recommendations.

Finally, after computing the recommendations, all results will be displayed in the recommendation view - Figure 2 (e). This view is automatically updated when the constraints are fulfilled (see Section *Recommendation Concept*) or on user demand (via button click). This view is also linked to both the navigation view and the exploration view. By clicking on a recommended plot, the bounding box of the navigation view as well as the viewport will update to the appropriate place.

For comprehensibility and monitoring reasons, a eye cursor can be enabled, which is displayed as a small red rectangle, as shown in Figure 2 (a) - row 4 and 5. For instance, users can enable the eye cursor for verifying the eye tracking calibration and disabling the cursor if they feel that it would distract them from the exploration task. Another important usage is to let third parties (e.g., analysts) follow eye movements of the user and analyze their exploration processes. However, one major feature of the eye cursor is that it gives exploration information about the current viewport and notifies the user if all current scatter plot classes were discovered. Immediately after all classes have been explored, the color of the eye cursor will change from red to green and thus informs the user to explore further areas of the SPLOM.

EXPERIMENTAL EVALUATION

In our preliminary evaluation, we test if our system allows users to discover and explore scatter plot patterns faster (more efficient) than fully interactive exploration. Therefore, we did an experimental comparison of user performance with and without recommendations.

Data Set: For our experiments, we use a data set from the *eurostat* data repository[3] that contains approximately 5500 data sets each with information about an EU-related statistic on topics including economy, population and industry. We tested our approach on a subset containing 27 statistical attributes including population density, duration of work, electricity consumption, etc. from 28 EU countries that show temporal changes over the last decades. From these 27 dimensions, we created a SPLOM resulting in 702 scatter plots ($n \times (n-1)$) in which each data instance (point) represents one EU country at a specific year.

As a ground truth to the experiment, we clustered all 702 plots into a number of classes (see Section *Visual Cluster Analysis*), representing visually similar plots. The set of clusters represents the data set and our analysis goal is to overview as many clusters as possible in a given amount of time. The clustering procedure returned acceptable results, nevertheless

[3]European Commission: http://ec.europa.eu/eurostat.

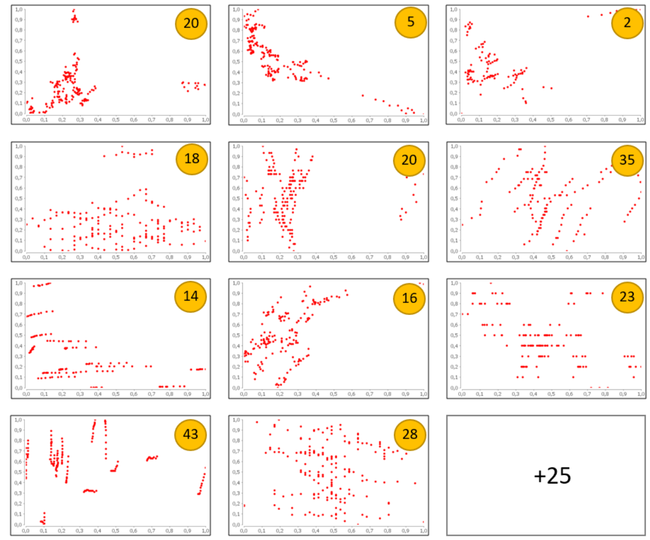


Figure 4. A subset of different scatter plot classes and their frequency of occurrence in the data. In total, we defined 36 scatter plot classes from 702 scatter plot views.

we manually adjusted the clusters by their visual similarities (i.e., similar pattern, similar trend, similar density, etc.). Finally, we obtained 36 different scatter plot classes as ground truth for our experiment. Figure 4 shows a subset of the scatter plot clusters thereof. On average, a cluster consists of 20 scatter plots, while the largest cluster consists of 43 similar scatter plots (Figure 4 - bottom left) and the smallest of 2 (Figure 4 - top right). This ensures that the experimental task is challenging and not too easy to accomplish.

Definition of Experimental Analysis Tasks: One common problem in exploratory analysis tasks is that users have to explore a large amount of plots as well as similar plots in order to find interesting patterns in the data. To show how our recommender system can help, we had users perform the following task: *Discover as many different global scatter plot patterns as possible within a given amount of time*. Each participant had three trials:
(1) Fully interactive SPLOM navigation; (2) Highlighting recommendations, but no navigation guidance; (3) Recommendations and navigation guidance.

We set the allowed exploration time to 2 minutes. We define as the success rate the number of unique clusters observed by the user during the exploration, and compare this across the three different system setups. For analysis of the exploration process, we captured screenshots with activated eye cursor and created a log file including mouse and keyboard events. Users were given a brief introduction to the system design and interaction techniques, and then allowed to try out the system for a first dry run. This introduction was followed by an interview session to gather initial feedback and to clarify possible issues. Afterwards, the users performed the actual experiment task, while their response time and exploration success rate were recorded. Finally, a questionnaire survey on system usability was conducted.

Results: The study included 12 participants (8 male, 4 female) all of whom were graduate students and had casual to

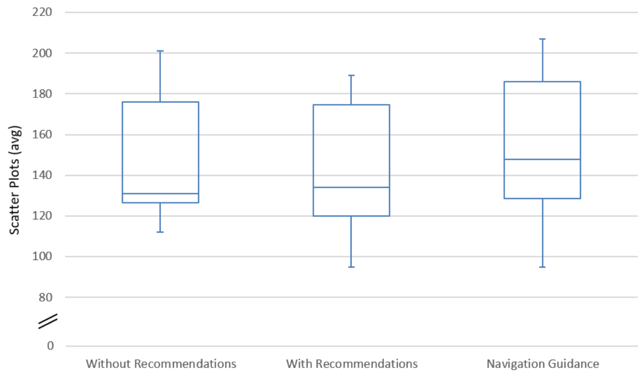


Figure 5. The box plot shows that participants explored less scatter plots in the 2. experiment (with recommendations) compared to the 1. experiment (without recommendations). On average the highest score was achieved in the 3. experiment.

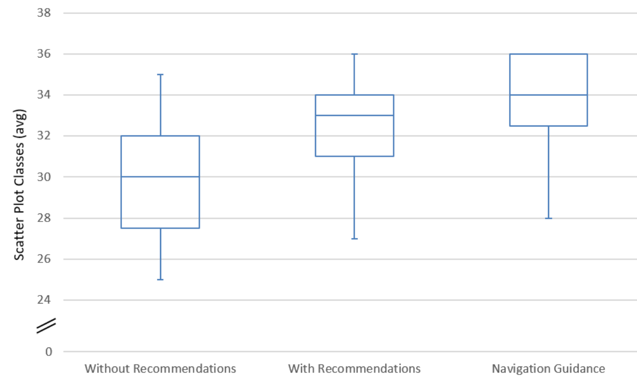


Figure 6. The results clearly show that the performance of the exploration of different scatter plot classes improved in the 2. and 3. experiment, which included the proposed visual recommender system.

moderate expertise in exploratory data analysis. To assess our approach, we observed for the three different experiment trials both the total number of explored scatter plots, as well as the number of unique scatter plot clusters. We count a scatter plot as explored, if it has been fixated for a minimum amount of time. We count a cluster to be explored if one of its member scatter plots has been fixated for 200 ms. The experiment results are shown in Figure 5 and 6. Figure 5 shows the average number of explored scatter plots including deviation for each experiment in a box plot. On average, the participants explored 147 scatter plots in the 1. experiment, 143 in the 2. experiment and 153 scatter plots in the 3. experiment. One can see from the box plot that the participants explored less scatter plots in the 2. experiment as compared to the 1. experiment, and achieved the highest score in the 3. experiment. We observed in the 2. experiment that some participants were somewhat distracted by the recommendations and lost time while looking at the navigation view. Our questionnaire survey indicated that some participants had difficulties at navigating and finding the suggested plots in the SPLOM. On the basis of our observations during the experiments and the survey results, we draw the conclusion that users may needed more time to process the information given by the recommender, and consequently, lost performance in exploring scatter plots and scatter plot classes respectively. This is confirmed by the fact that users explored more scatter plots by using our guided exploration function (3. experiment) where they do not have to take care of the navigation in the SPLOM.

However, if we now consider the results of explored scatter plot classes in Figure 6, one can clearly see the performance improvements of our participants in the second and third experiment, which include the proposed recommender approach. It is interesting to note that when considering the number of unique clusters explored, also the performance in the 2. experiment improved, although the total number of scatter plots explored was low in that experiment. In this case, the participants explored on average 30 scatter plot classes in the 1. experiment, 32 classes in the 2. experiment and 34 classes in the 3. experiment. Furthermore, it shows that the positive trend remains for all three boxes (median, upper and lower quartiles) and their whiskers (minimum and maximum values). By comparing only the top results of the first two experiments, one can see that the best participant explored 35 scatter plot classes in the 1. experiment and 36 classes in the 2. experiment. Even better were the results in the 3. experiment where the upper quartile –splits off the highest 25% of data from the lowest 75%– achieved the best experiment results of 36 explored scatter plot classes.

The study also showed that our approach supports the exploration of various patterns in the data and helps to obtain a better overview in SPLOMs. We were able to improve the exploration rate of scatter plot classes in both experiment trials using our visual recommender system, and achieved the highest scores by using our guided exploration function.

DISCUSSION AND EXTENSIONS

We also made several observations during the study and from the questionnaire survey. It turned out that most participants preferred to freely navigate within the SPLOM, but getting visual recommendations from the system (2. experiment) and following these interactively. One participant said he enjoyed following the recommendations within the SPLOM to see the visual changes of scatter plot views along the dimensions. However, some users stated they felt lost when the guided exploration function automatically changed their exploration view. This might be improved by motion animation to show the navigation direction within the SPLOM space, helping keep up a mental map of the overall SPLOM. Also, we observed different exploration strategies users followed in the first and second experiments. This was enabled by our heatmap view. We discovered different approaches like random exploration, column-wise, row-wise or clockwise navigation. While it would be interesting to relate these navigation patterns with the exploration success rates, here we could not find sufficient evidence in our study to do such correlation analysis. However, we believe that future experiments can be done relating exploration success with eye gaze paths.

One way to improve the precision and quality of the recommendations might be to upgrade the hardware of the system, e.g., by using eye tracking glasses or larger displays. We used an affordable eye tracker with a sampling rate of 30Hz to 60Hz and ran our system on a 25" screen. To achieve good calibrations and precise eye tracking results, we created a self-made chin rest to stabilize the participants' heads. We saw difficulties to calibrate two of our initially 14 invited participants and had to cancel their experiments due to poor calibration results, probably based on reflections given by glasses.

Our approach is a first concept and implementation for eye-tracking based SPLOM exploration and can be extended in many directions. User profiles could be generated to improve the recommendations based on individual user preferences and tasks, respectively. Thus, users could give explicit interest feedback to the system and train the k-NN recommender according to their preferences. As a result, the system could especially provide weighted recommendations based on different sets of geometric or graph-theoretic features, e.g., connectivity, density or outlying. Another interesting extension would be the integration of local scatter plot pattern analysis. Thus, the system could identify locally relevant areas of scatter plots, e.g., clusters or multiple correlations, and provide recommendations based on these.

Finally, we note that the user experiment is a first step to assess the effectiveness of our novel recommendation concept. Additional measurements can be defined, including qualitative user measurements and other exploration tasks might be tested. It remains a difficult problem to measure the value of analytical insight gained from interactive systems, and evaluation approaches for eye-tracking based analytics systems can be researched in future work.

CONCLUSION

We introduced a novel recommendation concept for exploratory analysis tasks, in which we utilize eye tracking information to suggest interesting and unseen views. We extracted image features of previously explored samples and used them to retrieve and recommend visually dissimilar views. Thus, users can focus on their exploration task while the system automatically captures explored features and constantly supports the user with exploration notifications and guidance. We implemented a prototype system for exploration, which also includes visualization of exploration provenance data. In a user experiment, we have compared the number of scatter plots and scatter plot classes that can be explored in a given amount of time, with and without our recommendation approach. The results show that our recommender system can increase the number of scatter plot patterns during exploration, which can improve the overall analysis success. This is a first concept and we we believe eye tracking has much potential to indirectly capture user information on an ongoing analysis process which can be useful to steer the analysis.

REFERENCES

1. G. Adomavicius and A. Tuzhilin. 2005. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17, 6 (June 2005), 734–749.
2. G. L. Andrienko, N. V. Andrienko, M. Burch, and D. Weiskopf. 2012. Visual Analytics Methodology for Eye Movement Studies. *IEEE Trans. Vis. Comput. Graph.* 18, 12 (2012), 2889–2898.
3. M. Behrisch, F. Korkmaz, L. Shao, and T. Schreck. 2014. Feedback-driven interactive exploration of large multidimensional data supported by visual classifier. In *Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on*. 43–52.
4. E. Bertini, A. Tatu, and D. Keim. 2011. Quality Metrics in High-Dimensional Data Visualization: An Overview and Systematization. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (Dec 2011), 2203–2212.
5. T. Blascheck, K. Kurzhals, M. Raschke, M. Burch, D. Weiskopf, and T. Ertl. 2014. State-of-the-Art of Visualization for Eye Tracking Data. In *EuroVis - STARs*.
6. A. Bojko. 2009. Informative or Misleading? Heatmaps Deconstructed. In *Human-Computer Interaction. New Trends: 13th Int. Conference*.
7. H. Bosch, D. Thom, F. Heimerl, E. Püttmann, S. Koch, R. Krüger, M. Wörner, and T. Ertl. 2013. ScatterBlogs2: Real-Time Monitoring of Microblog Messages Through User-Guided Filtering. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2022–2031.
8. M. Brooks, S. Amershi, B. Lee, S. M. Drucker, A. Kapoor, and P. Simard. 2015. FeatureInsight: Visual support for error-driven feature ideation in text classification. In *2015 IEEE Conference on Visual Analytics Science and Technology*. 105–112.
9. M. Ester, H. Kriegel, J. Sander, and X. Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proc. of the Second International Conference on Knowledge Discovery and Data Mining* (1996).
10. R. Etemadpour, B. Olk, and L. Linsen. 2014. Eye-tracking investigation during visual analysis of projected multidimensional data with 2D scatterplots. In *International Conference on Information Visualization Theory and Applications*. 233–246.
11. J. H. Goldberg and A. M. Wichansky. 2003. Eye Tracking in Usability Evaluation: A Practitioner's Guide. In *The Mind's Eye*. 493 – 516.
12. Mary M. Hayhoe. 2004. Advances in Relating Eye Movements and Cognition. *Infancy* 6, 2 (2004), 267–274.
13. C. Healey and J. Enns. 2012. Attention and Visual Memory in Visualization and Computer Graphics. *IEEE Transactions on Visualization and Computer Graphics* 18, 7 (2012).
14. F. Heimerl, S. Koch, H. Bosch, and T. Ertl. 2012. Visual Classifier Training for Text Document Retrieval. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012).
15. B: Höferlin, R. Netzel, M. Höferlin, D. Weiskopf, and G. Heidemann. 2012. Inter-Active Learning of Ad-Hoc Classifiers for Video Visual Analytics. In *IEEE Conference on Visual Analytics Science and Technology*. 23–32.
16. K. Holmqvist, M. Nystrom, R. Andersson, R. Dewhurst, H. Jarodzka, and J. van de Weijer. 2011. *Eye Tracking. A comprehensive guide to methods and measures*.
17. R. J. Jacob. 1991. The Use of Eye Movements in Human-computer Interaction Techniques: What You Look at is What You Get. *ACM Trans. Inf. Syst.* 9, 2 (1991), 152–169.
18. R. J. Jacob and K. S. Karn. 2003. Eye Tracking in Human-Computer Interaction and Usability Research: Ready to Deliver the Promises. In *The Mind's Eye*. 573 – 605.
19. P. Lops, M. de Gemmis, and G. Semeraro. 2011. *Content-based Recommender Systems: State of the Art and Trends*.
20. A. Poole and L. J. Ball. 2005. Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future Prospects. In *C. Ghaoui (Ed.): Encyclopedia of Human-Computer Interaction.*
21. M. Scherer, T. von Landesberger, and T. Schreck. 2012. A Benchmark for Content-Based Retrieval in Bivariate Data Collections. In *Proc. Int. Conference on Theory and Practice of Digital Libraries*.
22. F. Sebastiani. 2002. Machine Learning in Automated Text Categorization. *ACM Comput. Surv.* 34, 1 (2002), 1–47.
23. L. Shao, T. Schleicher, M. Behrisch, T. Schreck, I. Sipiran, and D. A. Keim. 2016. Guiding the exploration of scatter plot data using motif-based interest measures. *Journal of Visual Languages & Computing* 36 (2016), 1 – 12.
24. L. Wilkinson, A. Anand, and R. Grossman. 2005. Graph-Theoretic Scagnostics. In *Proceedings of the 2005 IEEE Symposium on Information Visualization*.
25. Chee Sun Won, Dong Kwon Park, and Soo-Jun Park. 2002. Efficient use of MPEG-7 edge histogram descriptor. *Etri Journal* 24, 1 (2002).