芸術科学会論文誌 Vol. 13, No. 3, pp. 185 – 197

属性推薦と特徴ベースフィルタリングを用いたシステムログ分析のための可視化手法

林 亜紀[1)](非会員)　伊藤 貴之[1)](正会員)　中村 聡史[2)](非会員)

1) お茶の水女子大学大学院 人間文化創成科学研究科
2) 明治大学 総合数理学部 先端メディアサイエンス学科

A Visual Analytics Tool for System Logs Adopting Variable Recommendation and Feature-Based Filtering

Aki Hayashi[1)]　Takayuki Itoh[1)]　Satoshi Nakamura[2)]

1) Graduate School of Humanities and Sciences, Ochanomizu University
2) Department of Frontier Media Science, School of Interdisciplinary Mathematical Sciences, Meiji University
{aki, itot} @ itolab.is.ocha.ac.jp, satoshi @ snakamura.org

概要

クレジットカードの決済情報などの購買ログや，ウェブのアクセスログといったシステムログの観察・分析は，ログの傾向の発見や，更新，セールなどのアクションを起こすのにふさわしい時期の特定，また不正なログの検出など多くの目的において有用である．しかしながら，システムログは多くの場合，非常に規模が大きく，属性数も多いため，有意義な分析結果を短時間で得ることは容易であるとはいえない．本稿では，多次元時系列データとしてのシステムログの効率的な分析を可能にする Visual Analytics Tool を提案する．本ツールでは，ディスプレイ空間をグリッド状に分割し，時系列属性を X 軸，それ以外の属性のうち 1 つを Y 軸に割り当て，各グリッドに該当するログの集計値を色で表示する．結果として本ツールでは，システムログの統計情報をヒートマップ形式で表示する．また，本ツールでは，有意性の高い可視化結果を提供する属性の推薦機能と，有用な情報だけを切り取った可視化結果を提示することで，可視化結果の可読性を高める特徴ベースフィルタリングを実現する．本稿では提案するツールの実行結果と非専門家によるユーザテストの結果により、提案手法の有効性を示す.
キーワード：ビジュアルアナリティクス、可視化、システムログ、ヒートマップ

Abstract

Analysis and monitoring of system logs such as transaction logs and access logs are important for various objectives including trend discovery, update or campaign effort determination, and malicious behavior monitoring. However, these logs may be massive, consisting of millions of records containing tens of variables, and therefore it may be difficult or time-consuming to discover significant knowledge. This paper presents a visual analytics tool which enables us to effectively observe system logs as time series. The presented tool divides the display into a grid assigning time-series to X-axis, one of other attributes to Y-axis and colors to the distribution of the logs. The tool visualizes statistical information of system logs using heatmap. The tool recommends variables that can reveal interesting discoveries and provides feature-based filtering that selects meaningful items from the visualization results. This paper presents the use cases and the experimental results performed by non-professional users.
Keywords: Visual analytics, visualization, system log, heatmap

1 Introduction

Time-varying and multi-variate data visualization are active research topics in information visualization and visual analytics. These two topics mostly developed independently in their early years; however, the visualization of time-varying multi-variate datasets has become an active research topic as of late.

This paper focuses on visualization of system logs. Many system log files have enormous numbers of records that contain various attributes. For example, records of credit card transaction logs contain attributes such as card ID, shop ID, item name, date and time, and pricing. Web access logs contain IP addresses , URLs of the accessed pages, dates and times, status codes, and referrer URLs. We can treat the logs containing various attributes as time-varying multi-variate datasets, and expect to obtain fruitful knowledge by observing the time-varying changes of the attributes. Visual analytics techniques for time-varying multi-variate datasets are useful for this purpose.

During the initial trial of visualization of system logs, we discussed the following issues:

- Necessity of variable recommendation:
  As mentioned above, system logs usually contain tens of attributes. It is often messy to look at the visualization results of all attributes at once, and therefore it is useful if visual analytics tools automatically suggest attributes valuable to visualize.
- Necessity of feature-based filtering:
  In our experience, large portions of system log datasets are noisy or trivial, and therefore we would like to selectively visualize meaningful or fruitful portions of the datasets. It is useful if noisy, meaningless, or trivial portions are interactively filtered from the visualization results.

This paper presents a visual analytics tool for system logs. The tool does not visualize all attributes (also called "variables" in this paper) in a single display space, but displays the time-varying change of statistics of a user-selected attribute. The tool features a heatmap-based display of statistics of system logs, by dividing the drawing space into a grid, assigning a temporal variable to the X-axis, and another variable to the Y-axis, and coloring the grid-subspaces according to the numbers of corresponding records. We preferred to implement a heatmap rather than polyline-based representation, because a heatmap does not cause cluttering problems even with large-scale data.

The presented tool features "variable recommendation" and "feature-based filtering". The variable recommendation technique calculates multiple scores which denote the importance of the variables. When a user selects one of the scores, the tool colors the button for interactive selection of variables according to the scores. Users can visually recognize the recommendation levels of the variables. The feature-based filtering technique displays smaller numbers of data items to improve the visual comprehensiveness. The technique provides two types of feature-based filtering: sorting and clustering. The sorting-based filtering technique calculates scores of each data item according to their importance, and sorts the data items by the scores. Users can interactively control the number of data items to be displayed, so that only important data items are visualized. The clustering-based filtering technique divides the data items according to their temporal patterns, and displays representative data items in the clusters. Users can visually discover interesting temporal patterns from the feature-based filtering result, and can interactively select one of the representative data items to display all the data items of the interested cluster.

This paper presents examples of visual analytics with real system logs of credit card transactions and Web accesses, and demonstrates what kinds of interesting trends can be interactively discovered. This paper also presents the result of experiment for non-professional users to examine the efficiency of the proposed visual analytics tool.

2 Related Work

The visual analytics tool presented in this paper is technically similar to multi-variate and time-varying data visualization techniques in the literature. Also, there have been several systems on visualization of transactions and Web access logs. This section introduces related work in the area of visualization techniques and systems.

2.1 Multi-variate Data Visualization

There have been variety of multi-variate visualization techniques. Non-time-varying multi-variate datasets can be effectively visualized in a single display space, by applying Parallel Coordinates [8] or Scatterplot-Matrices. Dimension analysis techniques have been applied to these visualization techniques, to determine meaningful orders of dimensions for Parallel Coordinates, or to select meaningful pairs of dimensions for Scatter Plots. Dimension reduction techniques have also been applied to obtain effective Scatter Plots. Migut et al. [14] presented a technique for visualization of dimensional dissimilarities, which is useful for dimensional analysis assisting decision makings.

Various dimension analysis and selection techniques for multi-variate data visualization have also been presented. Schneidewind et al. [19] presented a coordinated-view of Jigsaw maps and Pixel Bar Chart for dimension analysis and selection. Albuquerque et al. [1] presented a coordinated-view of RadViz, Pixel-Oriented Displays, and Table Lens for data clustering and dimension analysis on the top of Class/Cluster Density Measure. Ferdosi

et al. [4] presented a morphological-operator-based technique for selection of dimensions used for clustering and other processes. Tatu et al. [20] presented a technique for correlation- or distribution-selection of dimensions so that users can obtain fruitful visualization results by Parallel Coordinates or Scatter Plots. May et al. [13] presented a technique for feature-based selection of dimensions for Scatter Plots. VisSTAMP [5] also presented a technique for similarity-based dimension layout applying SOM (Self Organization Map) to obtain effective visualization results by Parallel Coordinates.

Compared with the above techniques, the presented visual analytics tool focuses on temporal change of multi-dimensional datasets, by featuring dimension recommendation and feature-based filtering with various criteria.

2.2 Time-varying Data Visualization

2.2.1 Representation of Time-varying Data

Polyline is the most common representation for time-varying data in our daily life. However, a polyline-based representation has two drawbacks:

- cluttering among large number of polylines in a single display space
- utilizing less display space when the numeric distribution is unbalanced

While several polyline-based techniques [21] were improved to solve the above problems, other representations including heatmaps have been applied to many time-varying data visualization techniques. Also, clustering of time-varying data is also effective to sample large-scale datasets. Other representations for time-varying data visualization include 3D histogram [11], spiral representation for periodicity analysis [22], piles of painted polylines such as ThemeRiver [6], and Two-Tone Pseudo coloring [18]. Heatmap-based representation has been also applied to various time-varying data visualization techniques, as well as the technique presented in this paper. We think heatmap has advantages over other representations from the standpoint of cluttering reduction and display space utilization for overviews. Imoto et al. presented a technique that extracts interesting portions of time-varying data on a heatmap [7]. Ziegler et al. [24] also presented a heatmap-based technique applying Pixel Bar Charts[10].

2.2.2 Summarization of Time-varying Data

We think there are two approaches for summarization in information visualization: numeric approaches such as clustering and sampling, and visual approaches such as focus+context techniques, including fisheye views. Clustering is useful for various goals in time-varying data visualization. Uchida et al. [21] presented a level-of-detail control technique for polyline-based time-varying data visualization, which selectively displays the preferable number of representative polylines when applying a clustering algorithm. Ziegler et al. [24] presented a technique for comparative visualization of multiple time-varying datasets by applying the common clustering algorithm to multiple datasets. Focus+context techniques are also useful for time-varying data visualization. For example, CloudLines [12] represents real-time time-varying datasets while shrinking older portions of the datasets.

The visual analytics tool presented in this paper realizes feature-based filtering by applying sorting and clustering. We believe these mechanisms are sufficient for the purpose of system log visualization, but we are interested in implementing focus+context techniques in a future work.

2.3 System Log Visualization

Credit card transactions are one of the applications of the visual analytics tool presented in this paper. This has been an active research topic in the data mining field [3], which aims to discover interesting trends or malicious behaviors. Several visualization techniques for transactions have been also presented. WireVis [2] is a coordinated-view system of heatmap, polyline charts, pie charts, and search results, which are applied to visualize transaction data. The heatmap featured by WireVis is one of the most relevant techniques to ours. However, it does not support variable recommendation, while ours supports variable recommendation as introduced in Section 3.3.

Since general transaction datasets contain large numbers of attributes, several techniques featured recommendation of meaningful sets of attributes to be visualized [15, 17]. Compared with such techniques for visualization of credit card fraud detection results, the presented visual analytics tool fixes the variable assigned to the X-axis to time, which makes attribute recommendation simpler. The paper presents a coordinated-view with the scatter plot based visualization [17]; WireVis has similar contributions.

Visualization of Web access logs is an active research topic since it is useful for improving the design and link structure of Web sites. Statistics of Web accesses [23] or access patterns [9, 16] have been mapped onto the link or hierarchy structure of Web sites. The technique presented in this paper is applied to visualize time-varying change of Web accesses, in contrast to these other techniques that do not focus on time.

3 Visual Analytics Tool

This section presents a visual analytics tool which realizes overview and feature-based filtering of system logs. Figure 1 is a screenshot of the GUI of the presented tool. We targeted support for credit card transaction logs and Web access logs as system logs while developing the tool. The tool provides interactive mechanisms to aggregate records of the system logs, displaying the results on demand. It

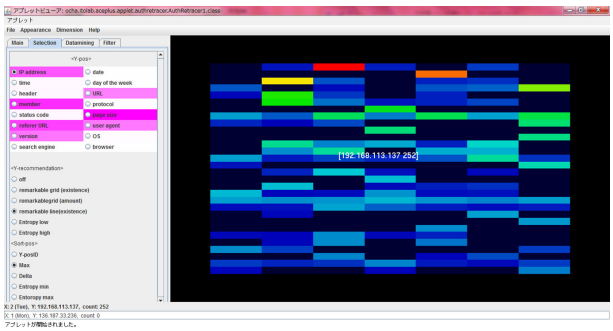

Figure 1: GUI of the presented visual analytics tool.

provides the three features:

1. overview of time-varying numeric features
2. recommendation of variables to be assigned to the Y-axis of the display space
3. feature-based filtering of visualization results, as the interactive data exploration mechanisms

These three features are freely combined. For example, users gain an overview of the data after assigning the recommended variables to the Y-axis, then summarize the visualization result, and filter unnecessary portions of the datasets. Users can select effective variables to assign to the Y-axis for the filtered datasets, then filter outlier portions of the datasets, and finally summarize the result again. The flexible combination of the features of this tool enables users to quickly discover interesting or important trends in the datasets.

3.1 Supposed System Logs

This paper defines a system log as a file containing a large number of records corresponding to the lines, and a record as a set of predefined variables. We applied the following two system logs to the presented visual analytics tool. The credit card transaction logs we used in this work contain 40 variables including date, day of the week, time, card ID, item code, shop ID, nation ID, price, fraud type, and so on. Fraud type refers to the falsification, theft, loss, or plagiarism of a card ID. The web access logs we used in this work are based on the Apache format, which contains variables including IP address of the browser, user name, date, time, URL, protocol version, status code, transferred bytes, referrer URL, and user agent. Our implementation removes records of accesses to content files (mainly images and sounds) as a preprocess.

3.2 Overview of the Time-varying Features

The presented tool represents system logs as a heatmap, by assigning a temporal variable to the X-axis, and another variable to the Y-axis. It displays the datasets avoiding cluttering and effectively utilizing the display space because it uses a heatmap. The tool does not represent every variable in a single display space, but just time-varying features of one variable. We suppose that users analyze the system logs by manually switching the variables to be assigned to X- and Y-axes. Also, our implementation supports a coordinate view mechanism with another multi-variate visualization technique that extracts user-specified portions of datasets and passes them to the other visualization component. This technique assists users in understanding multi-variate trends and phenomena of the datasets. Figure 7(c) is an example of the coordinate view mechanism, in which a specified portion of the dataset is delivered to a scatter plot component.

The variables of the system logs may contain both ordinal and nominal values. Ordinal variables (including temporal variables, prices, transferred bytes, and so on) are divided into predefined intervals. Nominal values are enumerated when system log files are opened.

Figure 2 denotes the processing flow and term definition of the presented visual analytics tool. The tool supposes that input system logs consist of records containing predefined numbers of variables. It divides the X- and Y-axes according to the number of intervals of ordinal values or the number of enumerated nominal values, generating a lattice there. This section calls the intervals of ordinal values or enumerated nominal values “items”. Each rectangular subspace of the lattice is colored according to the number of corresponding records. This section calls the number of the records the “amount”.

Figure 3 shows an example of the overview of a system log file. Warmer colors (e.g. red or orange) are assigned when the amount is relatively large, otherwise cooler colors (e.g. blue or green) are assigned. The tool also provides a GUI panel featuring buttons for the selection of variables to be assigned to the Y-axis. Colors of labels denote the recommendation level of the variables: Deeply colored variables are highly recommended to be assigned to the Y-axis.

The tool features a filtering mechanism based on particular values of the variables. When a user specifies interesting values in the user interface, the tool aggregates the records corresponding to the specified values. The tool provides two kinds of user interactions for filtering: mouse click and keyboard input operation. The tool allows users to specify values of any variables, not limited to the variables which are assigned to X- and Y-axes.

The tool also features an outlier detection mechanism. It can selectively display only outlier items or non-outlier items while recalculating the colors. Also, it can display

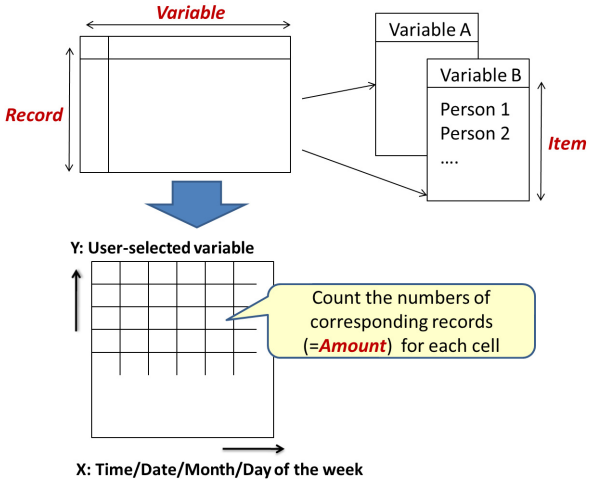

Figure 2: Processing flow and term definition. The presented tool counts the number of records (amount) corresponding to the value of selected variable (item) and time interval.

both outlier and non-outlier items while painting outlier portions by a particular color (pink in our implementation). The mechanism applies an outlier test that calculates τ_l by dividing the deviation of a sample x_1 by the unbiased standard deviation σ, as the following equation:

$$\tau_1 = (x_1 - \mu)/\sigma \tag{1}$$

where μ is an average and τ_1 is a user-defined constant value which can be set as 2.0, 4.0, or 6.0 by a user interface of our implementation.

In addition to the above filtering mechanisms, the tool features zooming, details on demand, and color adjustment functions. Users can freely zoom or shift the heatmap. They can obtain detailed information, including the value of a variable, and the amount, by clicking the rectangular subspace of the heatmap. Also, they can adjust the brightness of the heatmap by operating a slider, so that they can obtain a preferable distribution of colors.

3.3 Variable Recommendation

As above-mentioned, the presented tool visualizes system logs by assigning an arbitrary variable to the Y-axis. The effectiveness of the visualization results therefore strongly depends on the variable selection. This tool features a variable recommendation mechanism which suggests variables that contain meaningful numeric features.

Our implementation of the variable recommendation mechanism applies the following five criteria for the variable recommendation:

(r1) maximum amount in the whole dataset

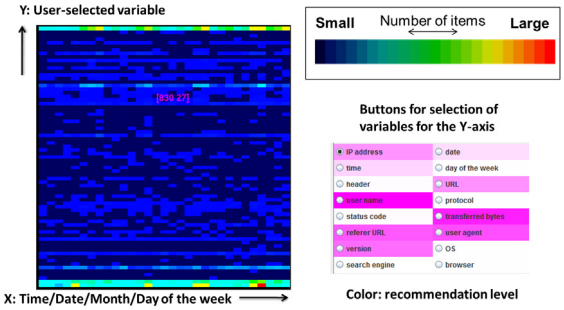

Figure 3: (left) The example of the overview of a system log. The color of each cell shows the relative amount as shown in the color map (upper-right). The tool also provides the GUI panel (lower-right) which shows the recommendation level of each variables by the intensity of the button color.

(r2) average of maximum amount in the items

(r3) maximum of total amount in the items

(r4) lowness of entropy of amount

(r5) highness of entropy of amount

Selecting one of the above five criteria, our implementation evaluates all the variables and displays the evaluation result by the brightness of the colors of the buttons corresponding to the variables, as shown in Figure 3 (right). This user interface is intuitive for users because they can subjectively select deeply colored variables. The following describes how to evaluate variables based on the five criteria.

(r1) maximum amount in the whole dataset:

Our implementation calculates the score e_1 by the following equation:

$$e_1 = (max_{whole} - ave_{whole})/ave_{whole} \tag{2}$$

where max_{whole} is the maximum amount in the whole dataset, and ave_{whole} is the average amount in the whole dataset. This criterion is effective to specify variables which contain extremely large amounts.

(r2) average of maximum amount in the items:

Our implementation calculates the score e_2 by the following equation:

$$e_2 = ave\{(max_{item} - ave_{item})/ave_{item}\} \tag{3}$$

where max_{item} is the maximum amount in an arbitrary item, and ave_{item} is the average amount in the item. Here $ave\{x\}$ means the average of x. This criterion is effective to specify variables which contain many items with extremely large amounts.

(r3) maximum of total amount in the items:

Our implementation calculates the score e_3 by the following equation:

$$e_3 = (tmax_{item} - tave_{item})/tave_{item} \tag{4}$$

where $tmax_{item}$ is the maximum of the total amount in an arbitrary item, and $tave_{item}$ is the average of the total amount in the item. This criterion is effective to specify variables which contain items with constantly large amounts.

(r4) lowness of entropy of amount:

Our implementation calculates the entropy H by the following equation:

$$H = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{5}$$

where n is the number of items, and p_i is the possibility of the i-th item. The variable contains more randomness if H is lower.

(r5) highness of entropy of amount:

Our implementation uses the entropy H as mentioned above. The variable is more uniform if H is higher.

3.4 Feature-Based Filtering

There may be an enormous number of items in a variable. For example, we have seen tens of thousands of URLs or referrer URLs in Web access logs. It is very difficult to visually recognize all the items if we display all of them as an overview. To solve this problem, this section presents feature-based filtering mechanisms that display fewer numbers of meaningful items, applying sort and clustering algorithms.

3.4.1 Sorting-based Filtering

Sorting-based filtering sorts the items in a variable assigned to the Y-axis, according to one of the following four criteria:

(s1) the maximum amount in each of the items

(s2) the maximum increase of amount in each of the items

(s3) lowness of entropy of amount in each of the items

(s4) highness of entropy of amount in each of the items

Figure 4 shows examples of feature-based filtering, which display less than 1,000 of meaningful items selected from over 6,000 items.

(s1) the maximum amount in each of the items:

Figure 4(s1) shows an example that displays 88 items. The tool selected not only items that have consistently high amounts, but also items that have high amounts for short periods of time.

(s2) the maximum increase of amount in each of the items:

Figure 4(s2) shows an example that displays 186 items. The tool preferentially selected items whose amounts are drastically increased.

(s3) lowness of entropy of amount in each of the items:

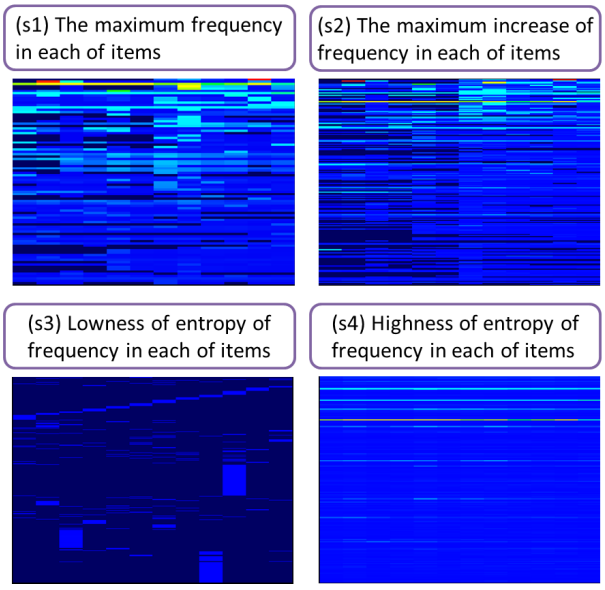

Figure 4: The results applying sorting-based filtering.

Figure 4(s3) shows an example that displays 882 items. The tool preferentially selected items whose amounts are randomly varied (not constant).

(s4) highness of entropy of amount in each of the items:

Figure 4(s4) shows an example that displays 189 items. The tool preferentially selected items whose amounts are relatively constant.

3.4.2 Clustering-based Filtering

Sorting-based filtering may sometimes fail to retain meaningful items. For example, meaningful items include those with cyclic peaks, but their amounts are not always preferentially retained by sorting-based filtering. Our tool provides clustering-based filtering as another solution. It divides the items in a variable into a predefined number of clusters based on the similarity of the items. Our current implementation applies the k-means clustering method to divide the items into 30 clusters.

Figure 5 shows an example of clustering-based filtering.

Representative items are selected in each of the clusters and displayed as shown in Figure 5(1). This example displays various patterns of temporal amount variations, where some of them may be missed to remain by sorting-based filtering. When a user selects one of the clusters by pressing a button corresponding to a particular cluster, the tool displays all the items belonging to the selected cluster. Figure 5(2) is an example of 326 items belonging to the user-selected cluster indicated by a pink rectangle in Figure 5(1). This example shows that many of items had extremely higher amounts in a short period. Our

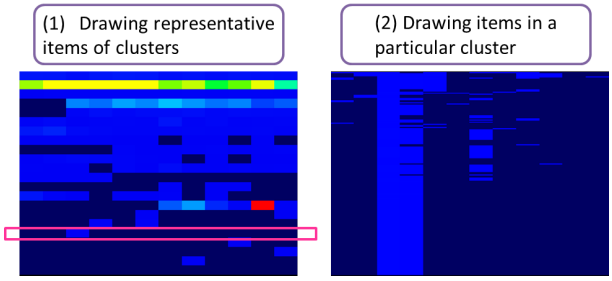

Figure 5: The results applying clustering-based filtering.

implementation indicates the number of items belonging to each cluster by the brightness of the button labels.

We preferred the k-means clustering method compared with hierarchical clustering techniques, because we wanted to fix the number of clusters due to user interface design matters, and we estimated that the k-means clustering method requires much less computation time than hierarchical clustering techniques for our datasets.

4 Use Cases

This section shows use cases of the visual analytics tool presented in the previous section, and demonstrates the effectiveness of the tool. The section firstly shows the visual analytics results of credit card transaction logs, including trend and fraud analysis. It then shows the visual analytics results of Web access logs, including analysis of overall trends, errors, and influences of updates. We implemented the presented visual analytics tool with JDK (Java Development Kit) 1.6.0, and executed it on Windows 7.

4.1 Trend and Fraud Analysis of Transactions

This section introduces examples of visualization results using a credit card transaction log.

4.1.1 Day-by-day Trend Analysis

Figure 6(a) shows the transactions during a month (November 2007), where the X-axis is divided by days, and the Y-axis denotes all shop IDs, which is the recommended variable by the criterion "(r3): maximum of total amount in the items". This example shows that there are consistently many transactions in several shops as indicated by two red circles in Figure 6(a). The example also denotes that transactions in several shops are periodically increased in weekends or holidays as indicated by red arrows in Figure 6(a). We then applied clustering-based filtering and visualized the contents of a particular cluster, as shown in Figure 6(b). The figure indicates that the shops in this cluster had relatively much more transaction activity in the final weekend, as indicated by red arrows. This kind of trend can be easily discovered by using the variable recommendation and feature-based filtering of this tool.

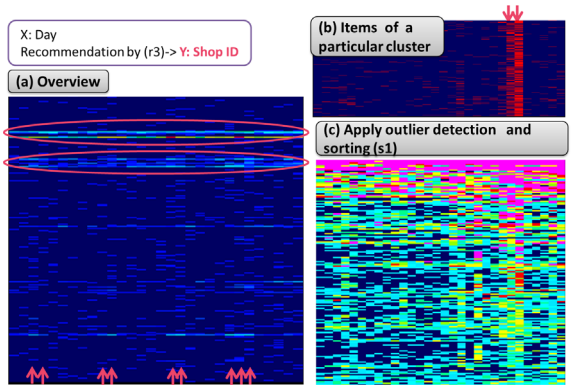

Figure 6: Day-by-day trend analysis of transactions. (a) overview, (b) applying clustering, and (c) applying outlier detection and sorting.

Figure 6(c) draws both outlier and non-outlier items (all shop IDs); the outlier items are in bright pink. The result denotes that shops which have larger maximum amounts have consistently large numbers of transactions, while other shops exhibit an extreme increase in transactions on particular days or weekends in the latter part of a month in this case. It is not easy to discover such trends from Figure 6(a), but we could easily discover them thanks to outlier detection and color recalculation.

4.1.2 Fraud Analysis by Variable Recommendation

The transaction logs we used in this work contain "fraud type", which are recorded after frauds are proven. We extracted 740 transactions which have fraud types from the original dataset during six months (from July 2007 to December 2012) and visualized them for the fraud analysis.

Figure 7(a) (left) shows a visualization result which applied sorting-based filtering and focused on the top four item codes. Here the X-axis is divided according to the day of the week, and the Y-axis is divided according to the item code, which is recommended by the criterion "(r4): lowness of entropy of amount". Figure 7(a) (right) draws only non-outlier items while recalculating colors for the improvement of readability. We found several trends of fraud transactions from this visualization: foreign items (row 1) have constant transactions, appliance (row 2) and train tickets (row 3) are active on weekends, and mail-order (row 4) is active on weekdays. Figure 7(b) shows a visualization result after only transactions of appliance

(item code = 650) are extracted, where sorting-based filtering is then applied. Here, the X-axis is divided according to days, and Y-axis is divided according to shop IDs, recommended by the criterion "(r3): maximum of total amount in the items". It draws all extracted shop IDs. This result denotes that many frauds are detected at a particular shop as indicated by a red thin rectangle in Figure 7(b). Moreover, we observed the shop indicated in Figure 7(b) by applying a scatter plot, as shown in Figure 7(c). We discovered there that falsified credit cards were often used on Saturdays(red), where amounts of most of transactions were around 100,000 Japanese yen. The result suggests that this particular shop was targeted or collaborated with malicious people who manufactured or used the falsified credit cards, and therefore the credit card fraud detection system may need to carefully observe this shop especially on Saturdays.

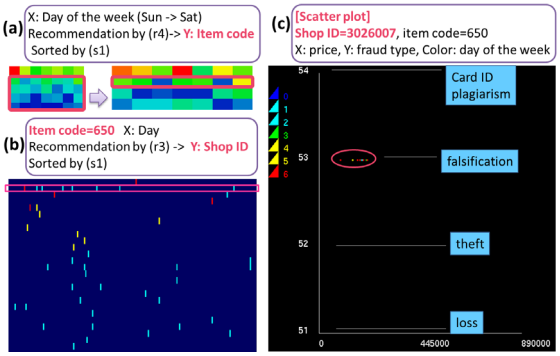

Figure 7: (a)(b) The time trend of fraud transactions using recommendation and sorting. (c) The more specified characteristic of malicious transactions using scatterplot.

4.2 Trend and Error Analysis of Access Logs

This section introduces examples of visualization results of Web access logs of two Web servers. One of them (called "Web A") is an access log of a Web server of a university laboratory, which contains Web pages of a professor and students. They introduce their research publications on the Web. The professor also provides various class materials with ID-password authentication. The other (called "Web B") is an access log of a Web server of a computer science researcher, which provides free software, BBS for the support of the software, introduction and review of restaurants, and Weblogs writing on computer science. Web B is relatively large and famous, and therefore Web B has more accesses.

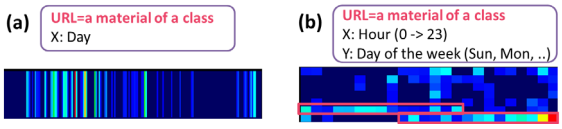

Figure 8: Analysis of appropriate update timing. (a) tendency of access through 6 month, and (b) time trend of each day of the week.

4.2.1 Analysis of Appropriate Timing of Updates

Figure 8 is a visualization results of Web A during 6 month (April 2011 to September 2011 which correspond to a semester in this university). We extracted the URLs of one of the materials for the professor's class held in this semester. Figure 8(a) is a visualization result assigning days to the X-axis. From this result we found big peaks of the accesses in the beginning of semester, and weekly small peaks after the big peaks, while accesses were fewer during vacations in August. We also visualized by assigning time to the X-axis (where the left end denotes AM0:00), and day of the week to the Y-axis (where the lower end denotes Sunday, and the upper end denotes Saturday), as shown in Figure 8(b). We found that the material had relatively more accesses from Sunday evening to Monday morning, especially 23:00 to 24:00 on Sunday as drawn in red in Figure 8(b). The class was opened in the morning on Monday, and therefore accesses were concentrated from Sunday evening to Monday morning. We concluded that the materials of this class should be updated until Sunday afternoon.yo

4.2.2 Error Analysis with Status Code 405

This section introduces accesses which had status code 405 (method not allowed), issued by servers to protect the systems from suspicious accesses.

Figure 9(a)-(d) shows visualization results of accesses of Web A which had status code 405 during about two years (November 2009 to September 2011). Figure 9(a) shows a result, where the X-axis is divided according to month, and the Y-axis is divided according to the accessed URLs. It draws all extracted URLs. We repeatedly observed a refused URL "/indonesia.htm" as indicated by a red rectangle (1). We could also observe three times of scan accesses to many URLs, which contained the extension ".asp", in a short term as indicated by another red rectangle (2). We estimated these accesses were suspicious, because all the above URLs did not exist.

Figure 9(b) is another example, where the X-axis is divided according to month, and the Y-axis is divided according to header, recommended with the criterion "(r4): lowness of entropy of amount". This result just displayed two headers, PUT, and CONNECT, while other major

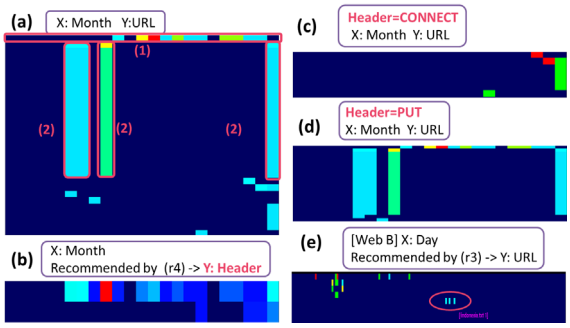

Figure 9: Error analysis with status code 405. (a) Overview of the access tendency. (b) Applying variable recommendation. (c)(d) The user can distinguish two kind of malicious accesses. (e) Comparison of the similar attack for [Web B].

headers such as GET or POST were not observed. We also visualized the same accesses by dividing the Y-axis according to versions and browsers, which were also recommended with the same criterion. Here we observed many of the accesses were from Microsoft Internet Explorer, which may be an attack targeting the Windows-specific weakness.

We extracted accesses which have status code 405 and header "CONNECT", and visualized all extracted URLs by dividing the X-axis according to month, and the Y-axis according to accessed URLs, as shown in Figure 9(c). We selected particular URLs from this result, and visualized the corresponding accesses by dividing the Y-axis according to IP addresses of browsers. Consequently we found three IP addresses which are famous as crackers [25]. We concluded we might be better to block access from these IP addresses.

Figure 9(d) is a visualization of accesses which have header "PUT", where the X-axis is divided according to month, and the Y-axis is divided according to all extracted URLs. The results looked similar to Figure 9(a), and again, the IP addresses of the browsers were famous attackers [25]. These results denote that accesses which had status code 405 contained suspicious accesses blocked by the Web server.

Figure 9(e) shows a visualization result of accesses of Web B during 6 months (from April to September, 2010), which have status code 405, where the X-axis is divided according to date, and the Y-axis is divided according to all accessed URLs, recommended with the criterion "(r3): maximum of total amount in the items". We found that less number of accesses had status code 405 on Web B, even though the total number of accesses of Web B was much more than that of Web A. Also, we found that most of URLs in Figure 9(e) were in existence, except the URL as indicated by a red circle in Figure 9(e). The exceptional URL was "/indonedia.txt", which was very similar to "/indonesia.htm" observed in the access log of Web A and these accesses started at almost the same time; however, the number of accesses in Web B was much fewer and occurred during a shorter term compared with Web A.

To explore this fact, we extracted the accesses inside the red circle, assigning IP addresses to the Y-axis. Then we found that there were three IP addresses, and none of them were already reported attackers. Web B was running on a secure commercial Web server while Web A was just on a server of a university laboratory, and therefore it seemed that Web B might strongly block malicious accesses rather than Web A. That might bring the difference of visualization results between Web A and B.

5 Evaluation

To evaluate the effectiveness of presented visual analytics tool, the authors performed user experiments with 27 student participants. The participants did not have any special knowledge of system log analysis. Before starting the tests, we explained all functionalities to the participants. After that we asked the participants to use presented tool visualizing the web access logs of Web A during one month (April, 2010).

We tested the effectiveness of sorting- and clustering-based filtering functions of the visual analytics tool. In this test, we asked the participants to play with the presented tool, and answer the questions how they could find specific behaviors. At the same time as the above test, we also asked them subjective evaluations regarding individual functions of the visual analytics tool, or the tool itself as a whole.

5.1 Quantitative evaluation

We tested how two types of presented feature-based filtering help the users to obtain the meaningful analysis from the multi-variate time-series data.

5.1.1 Sorting-based Filtering

	not apply	apply
a percentage of correct answer	20/21	21/21
average of required time (sec)	79.05	26.60

Table 1: Effectiveness of sorting-based filtering

We asked the participants to assign days to the X-axis and URLs to the Y-axis, and find the URL which has the maximum day accesses in a whole month. Table 1 compares the required time and the number of participants who found the correct URL with or without applying the sorting-based filtering. The number of participants who could get correct answers was increased. One participant

might made mistake without applying filtering because she answered the URL which was depicted next to the correct one. However all participants could find correct URLs using the presented filtering function. Most of them mentioned that they could easily find the correct URLs, because they could eliminate many URLs to observe and just focused on URLs depicted at the top of the window. 20 participants mentioned that they found it easier to discover such URLs when they applied the sorting-based filtering, and just 1 participant mentioned that it might take a long time to get used to understand how to use the sorting-based clustering.

Note that we regard only 21 participants gave valid responses because other 6 participants might have changed X-axis or Y-axis by mistakes during the experiment. They would not repeat such mistakes, if we gave more time to play with the visual analytics tool and then they got used to it.

We calculated the average time among 20 participants who could get correct URLs for both applying and not applying sorting-based filtering. The average of required time applying sorting-based filtering was about a third of the time without applying sorting-based filtering. This result demonstrates that the presented tool can shorten the time to find the items which have specific behavior.

5.1.2 Clustering-based Filtering

	not apply	apply
a percentage of correct answers	16/26	20/26
average required time (sec)	390.70	72.80

Table 2: Effectiveness of clustering-based filtering

Next, we asked the participants to assign days to the X-axis and IP addresses to the Y-axis, and find the IP addresses which have more than 80 times of sudden increase of accesses on 23rd, April. We treated three IP addresses which have such behaviors as correct answers. Table 2 compares the number of the participants who got correct IP addresses and the required time. This time we treated the results of 26 participants as available results because one participant misread the requirements and chose wrong attribute for Y-axis.

The presented tool could increase the number of correctly answered participants. Many participants failed to find all three correct IP addresses even though they could find one or two, without applying the filtering. On the other hand, 23 participants could find the correct cluster, and 20 participants could find all the correct IP addresses, while applying the filtering. 3 participants had mistakes because they might forget or misread the requirement which says "more than 80 times", though they could find the correct cluster.

The other 3 participants found the wrong cluster whose representative IP address had the maximum access (74 accesses) on that day among all 30 representatives. On the other hand, the representative of the right cluster had of course the peak on that day but that was only 45 accesses. In fact the IP addresses contained in the wrong cluster had a peak on another day; however, the participants did not understand the feature of clustering. This evaluation result suggests that we might need to improve the choice of representative of each cluster. One of the participants suggested that she wanted to know the common features of each cluster by a explicit way. 25 participants mentioned that clustering-based filtering made it easier to answer the question, because they could firstly observe the overall tendency, and then scrutinize the small number of items which have specific behavior. Although just one participant was skeptical because she thought that there could be the case that more than one clusters contained the items which had similar behavior. Our current implementation fixed the number of clusters due to the computation cost. We might need to adopt more advanced clustering algorithm which can adjust the number of clusters appropriate for each axis selections. Moreover, several participants mentioned that it was favorable if users could select more than one clusters to be drawn.

We also calculated the average time of experiments among the participants who found correct IP addresses for both applying and not applying the filtering. As a result, we found that the clustering-based filtering drastically speeded up the answering time.

5.2 Subjective evaluation

We conducted a subjective evaluation of both each functionalities and the tool as a whole. We asked the participants to answer the helpfulness on a scale of one to five. Table 3 shows the average rates of all participants. We also asked to comment the reasons of the evaluation, and then summarized their comments and suggestions.

functionalities	average rate
sorting-based filtering	4.25
clustering-based filtering	4.35
variable recommendation	3.42
outlier detection	4.44
overall	4.30

Table 3: Subjective evaluation of each functionalities

- The tool features a lot of functionalities which are useful to observe multi-variate time-series from various perspectives.
- It was difficult to master how to use this tool.
- Variable recommendation is useful, however, sometimes I could not understand the reason of recommendation.

- It would be more useful if the items which had similar tendency of outliers occurrence would be drawn nearby each others.
- It would be more useful if the tool indicated statistical information for example max, min and distribution of the statistical value.
- Sometimes I wanted to observe multiple visualization results selecting various attributes or applying different filtering.

We could get substantially high rate except variable recommendation. For variable recommendation, the user mentioned the criteria (r1) maximum amount in the whole dataset, (r2) average of maximum amount in the items and (r3) maximum of total amount in the items got relatively highly acclaimed, but others: (r4) lowness of entropy and (r5) highness of entropy got bad grade. The participants claimed that sometimes same attribute was recommended both (r4) and (r5); we may need to adjust the threshold for recommendation using entropies. Also, development according to the last three suggestions will be our future work.

5.3 Discussion

We obtained many ideas of our future work from the user experiments. From the evaluation by the user experiments, we could find the effectiveness in an aspect of time analysis with two types of feature-based filtering. The presented visual analytics tool supposes the users to practice their analysis skill using the tool for their daily work; in other words, it does not suppose them to understand the usage of tools in such a short time spent during the user experiments introduced in this paper. We assume that several mistakes the participants made this time will not occur when they get used to this tools after a certain time of training.

We obtained relatively good evaluation for feature-based filtering which we prepared several fixed tasks through the user experiments, and on the other hand we obtained relatively lower grade for evaluation of variable recommendation and outlier detection which we did not prepare fixed tasks. The result might be not the same as this time, if we demonstrated the example of effectiveness of variable recommendation and outlier detection. As a future work, we would like to conduct more detailed user tasks including demonstration of the sequence of tendency discoveries.

6 Conclusion

This paper presented a heatmap-based visual analytics tool for system logs including credit card transaction logs and Web access logs. The tool adopts a feature-based filtering technique to display fewer numbers of data items for improving user comprehension. It also features a variable recommendation technique, which allows the user to select variables that bring fruitful visualization results easily. The paper presented examples of visual analytics with real credit card transaction logs and Web access logs, and demonstrated what kinds of knowledge could be visually and interactively discovered. The paper also introduced the results of user experiments as evaluation.

In the future, we would like to add several features to the presented tool, such as logical operations for filtering unnecessary data items, and more sophisticated clustering and pattern recognition schemes for time-varying data. Also, we think it is important to automatically select meaningful criteria for variable recommendation (r1 to r5 in Section 3.3) and sorting-based filtering (s1 to s4 in Section 3.4). We then would like to experiment with larger datasets, and other kinds of system log datasets in addition to transactions and Web accesses.

Acknowledgments

We would like to express our gratitude to Intelligent Wave Inc. for providing a test dataset of credit card transactions.

References

[1] G. Albuquerque, M. Eisemann, D. Lehmann, H. Theisel, M. Magnor, Improving the Visual Analysis of High-dimensional Datasets Using Quality Measures, IEEE Symposium on Visual Analytics Science and Technology, 19-26, 2010.

[2] R. Chang, M. Ghoniem, R. Kosara, W. Ribarsky, J. Yang, E. Suma, C. Ziemkiewicz, D. Keim, A. Sudjianto, WireVis: Visualization of Categorical, Time-Varying Data from Financial Transactions, IEEE Symposium on Visual Analytic Science and Technology, 155-162, 2007.

[3] J. Dorronsoro, F. Ginel, C. Sanchez, C. Cruz, Neural Fraud Detection in Credit Card Operations, IEEE Trans. on Neural Networks, 8(4), 827-834, 1997.

[4] B. Ferdosi, H. Buddelmeijer, S. Trager, M. Wilkinson, J. Roerdink, Finding and Visualizing Relevant Subspaces for Clustering High-Dimensional Astronomical Data Using Connected Morphological Operators, IEEE Symp. on Visual Analytics Science and Technology, 35-42, 2010.

[5] D. Guo, J. Chen, A.M. MacEachren, K. Liao, A Visualization System for Space-Time and Multivariate Patterns (VIS-STAMP), IEEE Trans. on Visualization and Computer Graphics, 12 (6), 1461-1474, 2006.

[6] S. Havre, B. Hetzler, L. Nowell, ThemeRiver: Visualizing Theme Changes over Time, IEEE Symp. on Information Visualization, 115-123, 2000.

[7] M. Imoto, T. Itoh, A 3D Visualization Technique for Large Scale Time-Varying Data, 14th Intl. Conf. on Information Visualisation, 17-22, 2010.

[8] A. Inselberg, M. Reif, and T. Chomut, Convexity algorithms in parallel coordinates, Journal of ACM, 34(4), 765-801, 1987.

[9] M. Kawamoto, T. Itoh, A Visualization Technique for Access Patterns and Link Structures of Web Sites, 14th Intl. Conf. on Information Visualisation, 11-16, 2010.

[10] D.A. Keim, M. Hao, U. Dayal, M. Hsu, J. Ladisch, Pixel Bar Charts: A New Technique for Visualizing Large Multi-Attribute Data Sets without Aggregation, IEEE Symp. on Information Visualization, 113-126, 2001.

[11] R. Kosara, F. Bendix, H. Hauser, Timehistograms for Large, Time-Dependent Data, Eurographics/IEEE TVCG Symp. on Visualization, 45-54, 2004.

[12] M. Krstajic, E. Bertini, D. Keim, CloudLines: Compact Display of Event Episodes in Multiple Time-Series, IEEE Trans. on Visualization and Computer Graphics, 17(12), 2011.

[13] T. May, A. Bannach, J. Davey, T. Ruppert, J. Kohlhammer, Guiding Feature Subset Selection with an Interactive Visualization, IEEE Symp. on Visual Analytics Science and Technology, 109-118, 2011.

[14] M. Migut, J. Gemert, M.Worring, Interactive Decision making using Dissimilarity to visually represented Prototypes, IEEE Symp. on Visual Analytic Science and Technology, 139-147, 2011.

[15] A. Nagasaki, T. Itoh, M. Ise, K. Miyashita, A Correlation-based Hierarchical Data Visualization Technique and Its Application to Credit Card Fraud Data, 1st Intl. Workshop on Super Visualization, 2008.

[16] O. Nasraoui, H. Frigui, A. Joshi, R. Krishnapuram, Mining Web Access Logs Using Relational Competitive Fuzzy Clustering, Eight International Fuzzy Systems Association World Congress, 1999.

[17] C. Sakoda, A. Nagasaki, T. Itoh, M. Ise, K. Miyashita, Visualization for Assisting Rule Definition Tasks of Credit Card Fraud Detection Systems, IIEEJ Image Electronics and Visual Computing Workshop, 2010.

[18] T. Saito, H. Miyamura, M. Yamamoto, H. Saito, Y. Hoshiya, T. Kaseda, Two-Tone Pseudo Coloring: Compact Visualization for One-Dimensional Data, IEEE Symp. on Information Visualization, 173-180, 2005.

[19] J. Schneidewind, M. Sips, D. Keim, Pixnostics: Towards Measuring the Value of Visualization, IEEE Symp. on Visual Analytics Science and Technology, 199-206, 2006.

[20] A. Tatu, G. Albuquerque, M. Eisemann, J. Schneidewind, H, Theisel, M. Magnor, D. Keim, Combining Automated Analysis and Visualization Techniques for Effective Exploration of High-dimensional Data, IEEE Symp. on Visual Analytics Science and Technology, 59-66, 2009.

[21] Y. Uchida, T. Itoh, A Visualization and Level-Of-Detail Control Technique for Large Scale Time Series Data, 13th Intl. Conf. on Information Visualisation, 80-85, 2009.

[22] M. Weber, M. Alexa, W. Muller, Visualizing Time-Series on Spirals, IEEE Symp. on Information Visualization 2001, 7-14, 2001.

[23] Y. Yamaguchi, T. Itoh, Y. Ikehata, Y. Kajinaga, Interactive Poster: Web Site Visualization Using a Hierarchical Rectangle Packing Technique, IEEE Symp. on Information Visualization, 2002.

[24] H. Ziegler, M. Jenny, T. Gruse, D. A. Keim, Visual Market Sector Analysis for Financial Time Series Data, IEEE Symp. on Visual Analytics Science and Technology, 83-90, 2010.

[25] http://www.projecthoneypot.org/

林 亜紀

2010 年お茶の水女子大学理学部情報科学科卒業．2012 年お茶の水女子大学大学院人間文化創成科学研究科理学専攻博士前期課程修了．同年日本電信電話（株）入社．お茶の水女子大学大学院人間文化創成科学研究科理学専攻博士後期課程在学中．

伊藤 貴之

1990 年早稲田大学理工学部電子通信学科卒業．1992 年早稲田大学大学院理工学研究科電気工学専攻修士課程修了．同年日本アイ・ビー・エム (株) 入社．1997 年博士 (工学)．2000 年米国カーネギーメロン大学客員研究員．2003 年から 2005 年まで京都大学大学院情報学研究科 COE 研究員 (客員助教

授相当)．2005 年日本アイ・ビー・エム (株) 退職，2005 年お茶の水女子大学理学部情報科学科助教授 (准教授)．2011 年より同大学教授．ACM, IEEE Computer Society, 情報処理学会，芸術科学会，画像電子学会，可視化情報学会，他会員．

中村 聡史

1976 年生．2004 年大阪大学大学院工学研究科博士後期課程修了．同年，独立行政法人 情報通信研究機構 専攻研究員．2006 年京都大学大学院情報学研究科特任助手，2009 年同特定准教授，2013 年明治大学総合数理学部准教授，現在に至る．サーチとインタラクションや，情報曖昧化技術，ソーシャルアノテーション分析などの研究活動に従事．情報処理学会，ヒューマンインタフェース学会などの会員．博士（工学）．