

Research Article

Personalized Recommendation in Interactive Visual Analysis of Stacked Graphs

Alejandro Toledo,¹ Kingkarn Sookhanaphibarn,¹ Ruck Thawonmas,¹ and Frank Rinaldo²

¹DHCJAC, Ritsumeikan University, 1-1-1 Noji-higashi, Kusatsu, Shiga 525-877, Japan

²ISE, Ritsumeikan University, 1-1-1 Noji-higashi, Kusatsu, Shiga 525-877, Japan

Correspondence should be addressed to Alejandro Toledo, alex@ice.ci.ritsumei.ac.jp

Received 14 October 2011; Accepted 2 November 2011

Academic Editors: J. A. Hernandez, W. Lam, A. Salmerón, and L. S. Wang

Copyright © 2012 Alejandro Toledo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a system which combines interactive visual analysis and recommender systems to support insight generation for the user. Our approach combines a stacked graph visualization with a content-based recommender algorithm, where promising views can be revealed to the user for further investigation. By exploiting both the current user navigational data and view properties, the system allows the user to focus on visual space in which she or he is interested. After testing with more than 30 users, we analyze the results and show that accurate user profiles can be generated based on user behavior and view property data.

1. Introduction

Due to the exponential growth of information in virtually all industries, the need for computational tools that support the analysis process is increasing. As stated by Thomas and Cook [1], our ability to collect data is increasing at a faster rate than our ability to analyze it. In the past, recommender algorithms have been adapted to assist in this process. In the context of e-commerce systems, for example, we find methods that make item recommendations based on user profiles and item properties. Amazon and iTunes Store, for instance, are two popular e-commerce systems that have largely benefited from it. In the context of information visualization, however, we believe that recommender systems have not been sufficiently exploited.

We observe commonalities in e-commerce and information visualization systems. Both systems

- (1) endow information target elements, shaped as commercialized items in the former and visual hypotheses in the latter,
- (2) share a hierarchical structure upon which the exploratory analysis is carried on, and
- (3) can suffer from information overload.

To locate items, buyers often have to navigate from one page to another, observing items' properties until the target item is found. Likewise, to locate visual hypothesis, users have to explore different areas within the visualization space, observing views' properties, until the visual hypothesis is formed. When the burden of information is high, the person is faced with decisions such as determining how close the target element is with respect to his/her desirability. Furthermore, under extreme pressure, these kinds of decisions become more complicated to make.

In response to the information overload problem, researchers have devised solutions based on information-filtering techniques to prioritize the delivery of information for individual users. Recommender systems, for instance, have served as information-filtering techniques to recommend information items that are likely to be of interest to the user. Typically, a recommender system compares a user profile to some reference characteristics and seeks to predict the "rating" that the user would give to an item they had not yet considered. These characteristics may be from item information (the content-based approach) or the user's social environment (the collaborative filtering approach).

In this paper, we present a system which combines information visualization and recommender systems to support

information-seeking tasks in interactive visual analysis. In our system, a stacked graph visualization employs a content-based recommendation algorithm which infers user preferences from view dwell times and view properties. We selected the content-based approach because we wanted to avoid explicit ratings and a large usage history—commonly used in collaborative filtering. We tested our system using a data set of reported occupations in the United States labor force from year 1850 to 2000. The visual representation of this data set leads to a view space of nearly 10,000 unique views, which users can produce using the interaction controls provided by the system. After a usability study with 32 subjects, we show that effective guidance in visual search space can be produced by analyzing the time users spent on views, the properties of those views, and the properties of a collection of views obtained in advance.

The contributions of this work are (a) the adaptation of a content-based recommender algorithm to the stacked graph visualization, (b) the application of dwell times to inferring user preferences in visual exploration, and (c) the evaluation on the effectiveness of (a) and (b).

2. Related Work

Our work is similar to a large number of projects whose effort has led to a collection of visual analytic tools that help users explore, analyze, and synthesize data [2–6]. However, much of the work therein has focused primarily on helping users visualize and interact with data sets. In our work, we aim to support visual exploration of existing data together with an approach for automatically guiding users in the task of exploring time-series data sets. To this end, a recommender approach selects candidate views of a stacked graph for proposing to the users, aiming to match their analytical needs.

The notion of whether a visualization system can suggest useful views is an intriguing idea that has not been explored much. Koop et al. [7] proposed VisComplete, a system that aids users in the process of creating visualizations by using previously created visualization pipelines. The system learns common paths used in existing pipelines and predicts a set of likely module sequences that can be presented to the user as suggestions during the design process. This method is similar to the predictions made by unix command lines and can be considered as using the collaborative filtering approach of recommender systems. Our work uses the content-based approach.

Gotz et al. [8] proposed interactive tools to manage both the existing information and the synthesis of new analytic knowledge for sense-making in visualization systems. This work so far has not paid much attention on how to consolidate the users discoveries. Yang et al. [9] present an analysis-guided exploration system that supports the user by automatically identifying important data nuggets based on the interests of the users.

The notion of implicit/explicit feedback has been used to support recommendation systems. Traditional relevant feedback methods require that users explicitly or implicitly give feedbacks about their usage behaviors. Claypool et al. [10] provided a categorization of different implicit interest

categories, such as, dwell times on a page, mouse clicks, and scrolling. Studies on dwell times, in particular, have gained a special interest in the research community [11]. Morita and Shinoda [12] studied dwell times in the form of reading times to explore how behaviors exhibited by users while reading articles could be used as an implicit feedback.

Parson et al. [13] evaluated dwell times as an indicator of preference for attributes of items. In his work, Parson identifies a variety of factors that potentially could affect the dwell time, particularly in an uncontrolled setting. We believe that dwell times, as an indicator of preference, can be reliably extracted during interactive visual analysis. This is because viewers are arguably more focused than, for instance, buyers purchasing goods online. The most notable difference between Parson’s method and our method is the application domain. Parson conducted experiments on an e-commerce setting; our method is based, and evaluated, on a visual analytics setting.

Related work on stacked graphs can be found in websites such as Sens.us [14], ManyEyes [3], and NameVoyager [15], which allow many users to create, share, and discuss visualizations. One key feature of these systems is that they leverage the knowledge of a large group of people to effectively understand disparate data. Havre et al. [16] used the stacked graph to depict thematic variations over time within a large collection of documents. There is also work aiming at improving the aesthetics of the stacked graph [17, 18]. In our work, we adapt a recommender engine to the stacked graph visualization.

3. Methodology

To produce recommendations in interactive visual analysis, we based our study on the integration of a stacked graph visualization and a content-based recommender algorithm. Below we describe these two technologies and address our reasons for selecting them as part of our study.

3.1. The Stacked Graph Visualization System. In this paper, we use and extend the stacked graph visualization proposed in [14]. The stacked graph is a practical method for visualizing changes in a set of time series, where the sum of their item values is as important as the individual values. Our stacked graph implementation has been designed to visualize different time series sets. In this paper, we use a collection of 255 occupations reported in the United States labor force from 1850 to 2000. For each occupation, both male and female genders were reported. Thus, the entire data set is composed of 510 time series.

The method used to visualize the data is straightforward, that is, given a set of occupation time series, a stacked graph visualization is produced, as shown in Figure 1. The x -axis corresponds to years and the y -axis to occurrence ratios, in percentage against all occupations, for the occupations currently in view. Each stripe represents an occupation name, and the width of the stripe is proportional to the ratio of that occupation in a given year. The stripes are colored blue and pink for male and female genders, respectively. The brightness of each stripe varies according to the number of

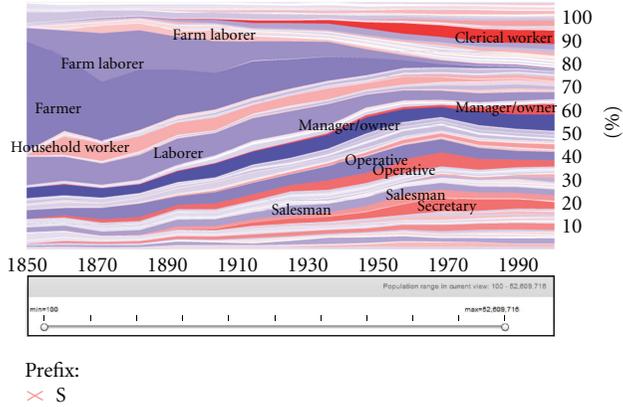


FIGURE 1: Stacked graph showing a collection of 255 occupations.

occurrences; the occupation with the largest population, for the whole period, is darkest and stands out the most.

As shown in Figure 1, when the system starts, the user sees a set of stripes representing all occupation names with the population range between 100 and 52,609,716. The former corresponds to the occupation with the lowest number of population (professor of statistics), and the latter to the occupation with the highest number (manager). Filtering of this data is achieved using two interaction controls. With the first one, filtering by a prefix, the user may type in letters, forming a prefix; our system will then display data of only those occupation names beginning with that prefix. The system reacts directly with each keystroke. Therefore, it is not necessary for the user to press return or to click a submit button. In addition, the system moves smoothly between visualization states. So when a letter is typed, an animated transition helps preserve the visualization context.

With the second interaction control, filtering by a population range, the user can change the data currently in use from the default values. The system provides a slider allowing the change using any population range between 100 and 52,609,716. The idea behind this interaction control is that we can restrict the view to certain data of interest, according to their population range, resulting in concise views of the data. Figure 2 shows a stacked graph filtered by both a prefix and a population range.

3.2. The View Data Set. As part of an offline computation, we obtained what we call the *view data set*, which is the collection of all unique views a user can produce from the system using the two aforementioned interaction controls. Considering the huge number of prefixes and population ranges and their possible combinations, it soon becomes clear that the total number of unique views would produce an extremely large view data set, thus impairing the system performance. One approach of producing a smaller view data set is considering prefixes of shorter length and a representative collection of population ranges. Using this approach, we obtained a view data set of nearly 10,000 unique views. For each, we collected the prefix, the population range, and the total number of occupations.

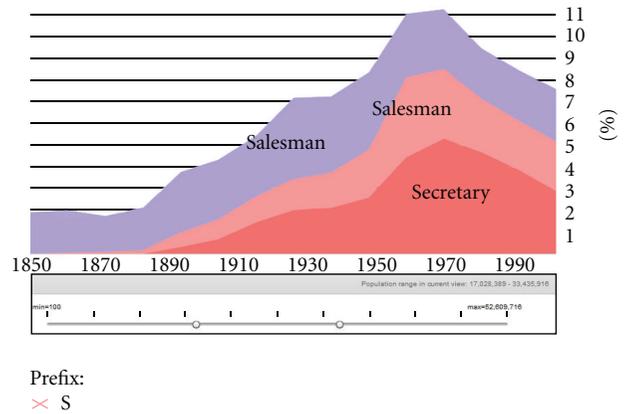


FIGURE 2: Stacked graph filtered by a prefix and a population range.

TABLE 1: Scheme structure of the view data set.

View ID	Textual attributes (prefix)					Numerical attributes (population range)			
	a	b	c	\dots	z	Min.	Max.	z (min.)	z (max.)
1	1	0	0	\dots	0	100	1000	-1.18	-0.77
2	0	0	1	\dots	0	1000	10000	1.27	1.41
\cdot	\cdot	\cdot	\cdot	\dots	\cdot	\cdot	\cdot	\cdot	\cdot
n	0	1	0	\dots	0	500	1500	-0.09	-0.65

3.3. Textual and Numerical Attributes. One can distinguish between textual and numerical attributes. Textual attributes correspond to view prefixes, whereas numerical attributes to population ranges. As shown in Table 1, each view is defined by a collection of binary-valued attributes, namely, a, b, c, \dots, z , where one of them is set to one—meaning that the view was produced using that prefix—and the rest to zero. Numerical attributes, on the other hand, are represented by the minimum and maximum numbers characterizing a population range. For the example below, we assume that the average and standard deviation for the attribute min. are 533.33 and 368.18, respectively, and for the attribute max. 4166.67 and 4129.84. Finally, we take advantage of this offline computation to obtain the z -score of each numerical attribute which is used in the online recommendation process described in the next section.

3.4. Content-Based Recommender Algorithm. As a means to expose potential views of interest, our system uses the seen views in the view data set to reveal views not seen yet by the user but still adhering to his/her user profile. This approach is comparable to the “discover the unexpected” as coined by Thomas and Cook [1]. Our method is based on the premise that preferences for a view is related to preferences for properties of that view. It uses only the current user’s navigational data in conjunction with view property data to

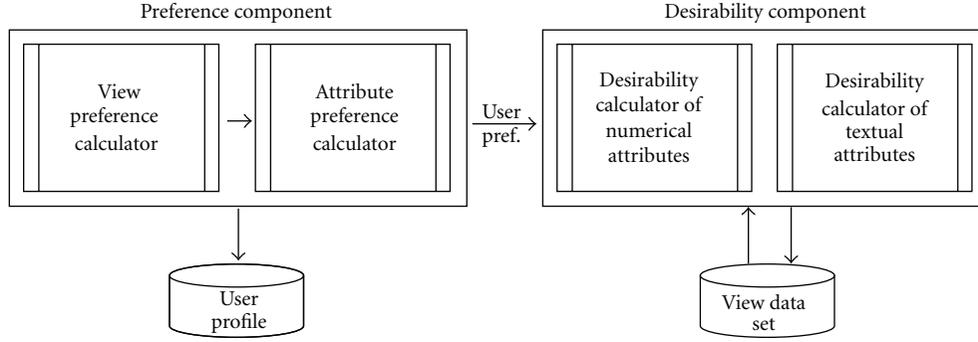


FIGURE 3: Schematic view of the DESIRE recommender algorithm.

make recommendations. We propose a solution based on the DESIRE algorithm proposed by Parson et al. [13] and present a demonstration of its use.

3.5. Definitions and Notations. In the following, A denotes the set of all possible attributes in a view. $\alpha \subseteq A$ denotes the textual-attribute subset, and $\beta \subseteq A$ denotes the numerical-attribute subset, where $\alpha \cap \beta \equiv \emptyset$ and $\alpha \cup \beta \equiv A$.

A view v is defined by a tuple of textual and numerical properties as $v = (\alpha(v), \beta(v)) \in V$, where V denotes the view data set of time series of interest, as described in the previous section. $\alpha(v)$ is the tuple of textual attribute values of view v and $\beta(v)$ the tuple of numerical attribute values of view v .

A user profile $P \subseteq V$ is defined as a collection of seen views, that is, views produced by the user while interacting with the system.

Definition 1. We defined the set of unseen views U as the complement of P with respect to V

$$U = V \setminus P = \{x \mid x \in V \wedge x \notin P\}. \quad (1)$$

Definition 2. The set of preferences for seen views v in P is defined as

$$\left\{ \text{PRF}_v = \frac{z(t_v)}{\epsilon} \mid v = 1, \dots, |P| \right\}, \quad (2)$$

where $z(t_v)$ is the trimmed z-score (cf. Algorithm 1) of the dwell time t of view v and ϵ a threshold for outliers account and normalization. Following the recipe in [13], we set ϵ to 3.

Definition 3. The set of preferences for textual attributes x in α is defined as the weighted mean of its values in all seen views:

$$\left\{ \text{PRF}_{\alpha_x} = \frac{\sum \alpha_x(v) \text{PRF}_v}{\sum \alpha_x(v)} \mid x = 1, \dots, |\alpha| \right\}, \quad (3)$$

where $\alpha_x(v)$ is the value of textual attribute x of view v in P .

Definition 4. The set of preferences for numerical attributes x in β is defined as the weighted mean of its values in all

```

procedure VIEWPREF( $P$ )
   $a \leftarrow$  average of all view dwell times in  $P$ 
   $s \leftarrow$  standard deviation of all view dwell times in  $P$ 
  for  $v = 1$  to  $|P|$  do
     $z(t_v) \leftarrow \frac{t_v - a}{s}$  // zscore of dwell time  $t$  of
    view  $v$ 
    if  $z(t_v) > \epsilon$  then // trimming zscore
       $z(t_v) \leftarrow \epsilon$ 
    else if  $z(t_v) < -\epsilon$  then // trimming zscore
       $z(t_v) \leftarrow -\epsilon$ 
    end if
     $\text{PRF}_v \leftarrow z(t_v)/\epsilon$  // preference for view  $v$ 
  end for
  return  $\{\text{PRF}_v\}$  // a set of view preferences
end procedure

```

ALGORITHM 1: Compute preferences for views.

positive samples, that is, all seen views with positive view preference values, as defined in (2):

$$\left\{ \text{PRF}_{\beta_x} = \frac{\sum_{\text{PRF}_v > 0} \beta_x(v) \text{PRF}_v}{\sum_{\text{PRF}_v > 0} \text{PRF}_v} \mid x = 1, \dots, |\beta| \right\}, \quad (4)$$

where $\beta_x(v)$ is the value of numerical attribute x of view v in P .

Algorithm. Given a user profile and a view data set, DESIRE returns a list of views sorted by the user's preference to view, analyze, or otherwise use them. As shown in Figure 3, two major components constitute this process. The first component infers user preferences. The second component calculates the degree of desirability for all views in the view data set.

First, the preferences for seen views are calculated, then the preferences for attributes. In the former, we use a method based on the dwell time as an implicit interest indicator (see Algorithm 1); in the latter, the preferences of attributes are obtained for both numerical and textual attributes (see Algorithm 2). The output of the preference component is a pair of sets as defined in (3) and (4).

```

procedure ATPREF({PRFv})
  for  $x = 1$  to  $|\alpha|$  do // preferences for textual attributes
    calculate PRF $\alpha_x$ 
  end for
  for  $x = 1$  to  $|\beta|$  do // preferences for numerical
  attributes
    calculate PRF $\beta_x$ 
  end for
  return {PRF $\alpha$ ; {PRF $\beta$ } // two sets: textual attribute
  preferences, and numerical-attribute preferences
end procedure

```

ALGORITHM 2: Compute preferences for attributes.

Secondly, we obtain the degree of desirability for all views in the view data set. We first calculate the similarity between attribute values in views and the attribute preferences. Formally, the set of similarity of textual attributes in view v with respect to the textual attribute preferences is defined by

$$\left\{ S_{\alpha_x(v)} = \frac{1 + f(\alpha_x(v))}{2} \mid x = 1, \dots, |\alpha| \right\}, \quad (5)$$

where $\alpha_x(v)$ is the value of textual attribute x of view v in V , and the function f is defined by the rule

$$f(\alpha_x(v)) = \begin{cases} \text{PRF}_{\alpha_x} & \text{if } \alpha_x(v) = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

Likewise, the set of similarity of numerical attributes in view v with respect to the numerical attribute preferences is defined by

$$\left\{ S_{\beta_x(v)} = 1 - \frac{|z(\text{PRF}_{\beta_x}) - z(\beta_x(v))|}{2\epsilon_{\beta_x}} \mid x = 1, \dots, |\beta| \right\}, \quad (7)$$

where $z(x)$ is the trimmed z-score of x calculated based on the mean and standard deviation of the corresponding numerical attribute for all views in V . $\beta_x(v)$ is the value of numerical attribute x of view v in V , and ϵ_{β_x} is a threshold of numerical attribute x .

Finally, S_α and S_β are combined to produce a single index that represents the degree of desirability for each view in the view data set. Since some attributes are more important than others, the desirability for view v is a weighted mean defined by

$$D_v = \frac{\sum R_{\alpha_x} S_{\alpha_x(v)} + \sum R_{\beta_x} S_{\beta_x(v)}}{\sum R_{\alpha_x} + \sum R_{\beta_x}}, \quad (8)$$

where R_{α_x} and R_{β_x} denote the relative weights for textual and numerical attributes x , respectively.

The system then recommends the top- n unseen views in U with the highest desirability.

3.6. Usage Scenario. We now want to give a step-by-step demonstration of how a typical visual-interaction session takes place. The goal is to highlight the major features employed by our system to recommend views not seen by the user but still adhering to his/her user profile. We illustrate this using a sequence of two different recommendation sessions. On each session, the system produces the top-5 recommendation set of views, sorted from highest to lowest rank number.

The initial view shown in Figure 4(a) illustrates a view containing one of the initial attribute settings employed in our system evaluation: prefix = none, min. = 25,000,000, and max. = 52,609,716. Figure 4(b) shows the results of the first top-5 recommendation session (only three views are shown here due to limited space) from a dwell time of 10 seconds. A strong similarity in the value distribution of min. and max. can be observed between the initial and the recommended views. The system also tries to diversify the values for the prefix attribute. Furthermore, the initial view, already seen by the user, is not recommended by the system. Soon after, the top-1 view is selected from the recommendation list. As shown in Figure 5(a), this view becomes the current one in the system and, as before, will not be recommended in subsequent recommendation sessions. The property values of this view are as follows: prefix = F , min. = 10,219,426, max. = 52,609,716. Finally, in the second recommendation session, Figure 5(b), from a dwell time of 80 seconds, the system recommended a view with prefix = F , which shows the effects of a higher dwell time on the selected view. Also, the system keeps similar value distribution of occurrence frequencies.

4. User Study

Using an approach similar to that used in [13], we conducted a usability study. Our goal was to explore the quality of the recommendations produced by our system. For this, we used a comparison page containing high- and low-ranked views, and we expected the participants to show interest in the high-ranked-view group. This method is described later in this section.

We employed a concept known as *occupational sex segregation*, which refers to a sociological term [19, 20] used for specific occupations in which women or men are over- or underrepresented; that is, they work almost exclusively with their same gender. One can distinguish between increasing and decreasing occupational sex segregation. Increasing segregation refers to occupations in which at the beginning of the period a weak segregation can be observed, though later this segregation is increased. Conversely, decreasing segregation refers to occupations with strong segregation at the beginning and decreasing segregation later in time. In Figure 6, hospital attendant shows increasing segregation, while farm occupations, in Figure 7, decreasing segregation. In both cases the population worked almost exclusively with their same gender.

We recruited 32 participants to participate in a visual analytics activity. Participants in this study were computer science students taking courses at our university and consisting of 21 undergraduate students and 11 graduate students

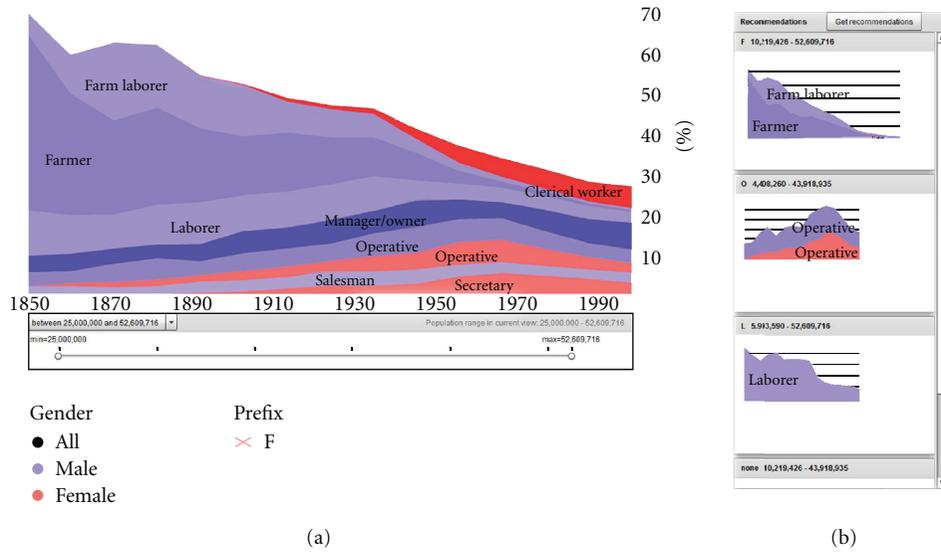


FIGURE 4: First recommendation session from initial view and dwell time of 10 seconds.

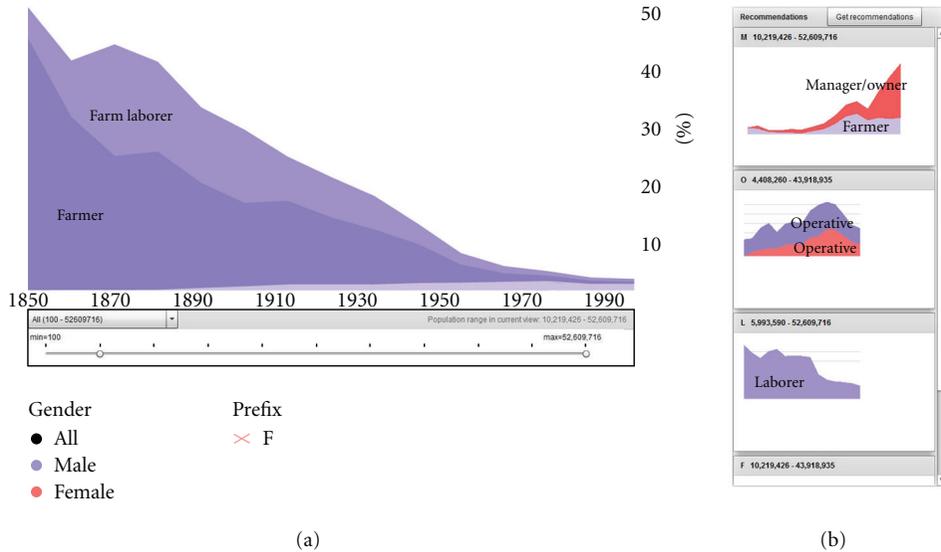


FIGURE 5: Second recommendation session, with preference on textual attribute and dwell time of 80 seconds.



FIGURE 6: View with increasing occupational sex segregation.

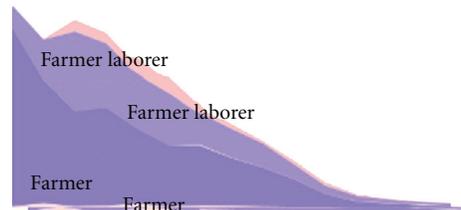


FIGURE 7: View with decreasing occupational sex segregation.

with the average age of 22.3, ranging from 20 to 27 years old. We hypothesized that recommendations based on rankings calculated by our system were better than randomly generated recommendations.

We prepared two tasks involved in searching, using the features of the visualization system, for occupations the participants believed had evidences of sex segregation. To begin with, each participant was given an introductory tutorial to

occupational sex segregation as well as the user guide of the visualization system. Next, they were asked to use the system for 10 minutes. The idea behind this is that participants could learn how to use the system before doing the actual tasks. After this period, they were given two population ranges: 10,000,000–25,000,000 and 25,000,000–52,609,716. Then, for each one, they had to answer the following question—*which is the most segregated occupation?* For each population range, each participant had 3 minutes to provide an answer.

While each participant was looking for segregated occupations, the system collected both dwell time and properties for all produced views. After three minutes, all unseen views in the view data set were ranked and a comparison page was generated for use in the subsequent phase. The purpose of the comparison page was to display unseen views that were ranked high and low by the system and measure how frequently participants chose high-ranked views on the comparison page. If the system produces good rankings based on preferences, participants should pick high-ranked views most of the time. Thus, this evaluation provides a relatively simple yet reliable way of determining how well the system works.

For each participant, the comparison page consisted of 12 unseen views, with 6 views being high-ranked and 6 views being low ranked. These views were obtained from the top-100 unseen views in the view data set. From them, the high-ranked views were selected from the top 6, and the low-ranked views from the bottom 6. To prevent biases that might arise from position effects, the positions of those 12 unseen views were randomly decided. The participant was instructed to choose five views, in terms of their usefulness for supporting the answer. After viewing the comparison page, the participant was routed to the last page in which, from the five selected views, he or she had to select the most-like view. Finally, the participant was asked to write the reasons that motivated the second selection.

5. Results and Discussions

In this section, we report observations from our study on user profile generation. The data analyzed were drawn from usage logs including 320 views chosen by the participants in the first selection and 64 views chosen in the second selection. We analyzed the results under the premise that the recommendations provided by our system are correct if users can find their answers and if those answers are supported by our system.

We first wanted to know the frequency that high-ranked views were chosen by the participants during the first selection. From the second selection, we wanted to know the rank of chosen views and whether those views were useful for the participants. Are they high-ranked views?, do they contain the users' answers? Finally, we wanted to learn from these participants their reasons that motivated the second selection.

From the first selection, we computed the percentage of times that participants chose views that were ranked high by the system. If view rankings were random, we would expect 50% of the higher ranked views be chosen in the first

selection. However, those views were chosen by participants 63.75% of the time. This difference is statistically significant (P value: 0.01) according to the Chi-square test. Although a 63.75% effectiveness score does not seem high, this should be viewed in the context that the system recommended only unseen views.

From the second selection, we computed the percentage of times that the participants chose a high-ranked view. Since the selection space was produced by each participant from the first selection, we expected a high percentage here. Views in the high-ranked group were chosen 68.75% of the time. This difference, from 63.75% is not statistically significant (P value: 0.05) and shows a consistent user criteria between the two selections.

The quality of recommendations were measured using two steps. First, we quantified the number of times that users could find their answer. An answer can be said *found* if either of the following two conditions is satisfied: (1) the written answer is contained in the most-like view, (2) the most-like view contains similar attributes as in the written answer. In the second step, we quantified the percentage of *found* answers that are associated with high-ranked views. From 30 *found* answers satisfying the first condition, 73.34% correspond to high-ranked views. Likewise, from 26 *found* answers satisfying the second condition, 69.23% correspond to the high-ranked group.

Finally, we performed a content analysis on the collected comments about reasons that motivated the second selection. After reading all comments, the authors agreed in that comments can be categorized into *simplicity* or *aesthetic*. According to participants' comments, most views were selected because of their simplicity. One participant, for instance, wrote: "*because only one occupation was shown,*" another participant commented: "*the trends between male and female occupations were easy to compare,*" one more just said "*Simple*". The adjectives *simple*, *clear*, and *easy* were very common in this category. On the other hand, participants whose reasons communicated some aesthetic feature wrote, for instance, "*the color combination was beautiful,*" "*the darkest graph was shown clearly,*" or "*the borders were shown clearly.*" Other three participants wrote that they selected the views because the occupations they selected were shown there.

6. Conclusion and Future Work

In this paper, we raised the question of how recommending can be incorporated into an information visualization system for suggesting interesting views to the user. Aiming to alleviate the overload caused in the visual exploration of more than 10,000 views, we proposed a method for adapting a recommender engine to the stacked graph visualization. Our method employs a content-based recommendation algorithm which uses dwell times to infer user preferences on stacked graph views. We rely on the premise that preferences for a view are related to preferences for properties of that view. A usability study with more than 30 subjects shows that (1) accurate user profile can be generated by analyzing user behavior and view property data, (2) dwell times can

be a useful indicator of preferences, (3) preference for a view can be expressed as a preference function of view-attribute values, and (4) attribute preferences can be transferred across views.

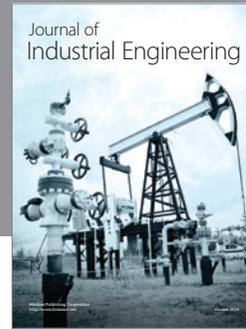
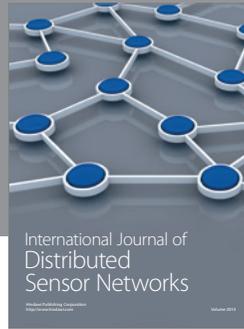
While our system shows promise, there remain several areas that can be improved upon. We would like, for example, to explore alternative interest indicators for inferring user preferences such as mouse movements, annotations on a view, revisits on a view, and so forth. In our previous work, we tested various combinations of them and would like to know their effectiveness when applied to the system. In addition, we are exploring how to integrate our recommendation engine with an interactive evolutionary algorithm to support aesthetics in the views produced by the recommendation system.

Acknowledgment

This work is supported by the MEXT Global COE Program—Digital Humanities Center for Japanese Arts and Cultures, Ritsumeikan University.

References

- [1] J. J. Thomas and K. A. Cook, *Illuminating the Path: The Research and Development Agenda for Visual Analytics*, vol. 54, no. 2, IEEE, 2005.
- [2] N. H. Riche, B. Lee, and C. Plaisant, “Understanding interactive legends: a comparative evaluation with standard widgets,” *Computer Graphics Forum*, vol. 29, no. 3, pp. 1193–1202, 2010.
- [3] F. B. Viegas, M. Wattenberg, F. van Ham, J. Kriss, and M. McKeon, “Manyeyes: a site for visualization at internet scale,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, pp. 1121–1128, 2007.
- [4] W. Willett, J. Heer, and M. Agrawala, “Scented widgets: improving navigation cues with embedded visualizations,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1129–1136, 2007.
- [5] R. Vuillemot, T. Clement, C. Plaisant, and A. Kumar, “What’s being said near Martha? Exploring name entities in literary text collections,” in *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST ’09)*, pp. 107–114, IEEE, 2009.
- [6] P. J. Crossno, D. M. Dunlavy, and T. M. Shead, “LSAView: a tool for visual exploration of latent semantic modeling,” in *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, pp. 83–90, October 2009.
- [7] D. Koop, C. E. Scheidegger, S. P. Callahan, J. Freire, and C. T. Silva, “VisComplete: automating suggestions for visualization pipelines,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, Article ID 4658192, pp. 1691–1698, 2008.
- [8] D. Gotz, M. X. Zhou, and V. Aggarwal, “Interactive visual synthesis of analytic knowledge,” in *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST ’06)*, pp. 51–58, November 2006.
- [9] D. Yang, E. Rundensteiner, and M. Ward, “Analysis guided visual exploration of multivariate data,” in *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, pp. 83–90, Citeseer, 2007.
- [10] M. Claypool, P. Le, M. Wased, and D. Brown, “Implicit interest indicators,” in *Proceedings of the 6th International Conference on Intelligent User Interfaces (IUI ’01)*, pp. 33–40, ACM, New York, NY, USA, January 2001.
- [11] D. Kelly and J. Teevan, “Implicit feedback for inferring user preference: a bibliography,” *SIGIR Forum*, vol. 37, no. 2, pp. 18–28, 2003.
- [12] M. Morita and Y. Shinoda, “Information filtering based on user behavior analysis and best match text retrieval,” in *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’94)*, pp. 272–281, Springer, New York, NY, USA, 1994.
- [13] J. Parsons, P. Ralph, and K. Gallagher, “Using viewing time to infer user preference in recommender systems,” in *Proceedings of the AAAI Workshop on Semantic Web Personalization*, vol. WS-04-09, pp. 52–63, American Association for Artificial Intelligence, 2004.
- [14] J. Heer, F. B. Viégas, and M. Wattenberg, “Voyagers and voyeurs: supporting asynchronous collaborative visualization,” *Communications of the ACM*, vol. 52, no. 1, pp. 87–97, 2009.
- [15] M. Wattenberg and J. Kriss, “Designing for social data analysis,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 4, pp. 549–557, 2006.
- [16] S. Havre, E. Hetzler, P. Whitney, and L. Nowell, “ThemeRiver: visualizing thematic changes in large document collections,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 1, pp. 9–20, 2002.
- [17] L. Byron and M. Wattenberg, “Stacked graphs—Geometry & aesthetics,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, Article ID 4658136, pp. 1245–1252, 2008.
- [18] J. Heer, N. Kong, and M. Agrawala, “Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations,” in *Proceedings of the 27th International Conference on Human Factors in Computing Systems*, pp. 1303–1312, ACM, 2009.
- [19] J. A. Jacobs, “Long-term trends in occupational segregation by sex,” *American Journal of Sociology*, vol. 95, no. 1, pp. 160–173, 1989.
- [20] M. Charles and D. Grusky, *Occupational Ghettos: The Worldwide Segregation of Women and Men*, Stanford University Press, 2005.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

