A GUI Design for Composition Discovery of View Interestingness

Xiaozhong Zhang
Computer Science Department
University of Pittsburgh
xiaozhong@pitt.edu

Xiaoyu Ge
Computer Science Department
University of Pittsburgh
xiaoyu@cs.pitt.edu

Panos K. Chrysanthis
Computer Science Department
University of Pittsburgh
panos@cs.pitt.edu

ABSTRACT

View recommendation has emerged as a powerful tool to assist data analysts in exploring and understanding big data. Due to the large search space of possible views, finding a view that shows interesting patterns is not a trivial task. Existing view recommendation approaches have proposed a variety of utility measures in selecting interesting views. Even though using a single utility measure or a linear combination of multiple utility measures might be suitable in specific scenarios for view interestingness estimation, we claim that any assumption of the composition of view interestingness could be inaccurate without verification from real users.

Therefore, in this paper, we propose a novel graphical user interface (GUI) designed to be used in a user study to shed light on the composition of view interestingness. Specifically, we first create a classification system for view recommendation tasks, and identify utility measures suitable for each category. Then, we design a GUI that uses the identified utility measures to discover how users evaluate the views with respect to the utility measures, and how they assess the overall view interestingness based on the utility measures. Finally, we use an example to illustrate how the user answers to the questions in the GUI can be used to discover the composition form of view interestingness.

1 INTRODUCTION

The ubiquitously available information sources and the advancements in data storage and acquisition techniques have led to an aggressive increase in the data volumes available for data analysis tasks. One major challenge in utilizing these abundantly available data is discovering insights from them effectively and efficiently. Examples of an "insight" include the structure, patterns, and causal relationships. To explore these massive and structurally complicated datasets, data analysts often utilize visual data analysis tools such as Tableau [1] and Voyager [8]. However, the effectiveness of these tools depends on the user's expertise and experience. Coming up with a visualization that shows interesting trends/patterns is a non-trivial issue, because the search space of possible visualizations is prohibitively large.

To address the above challenge, several methods for recommending views (i.e., histograms or bar charts) have recently been proposed (e.g., [2, 3, 5, 7, 9]). These methods automatically generate all possible views of the data, and *recommend* the *top-k interesting* views, according to some utility function (e.g., deviations, data variance, usability) that measures the interestingness of views.

However, using a single utility function to estimate the interestingness of a view is usually not enough, because the interestingness of a view usually involves multiple utility functions *simultaneously*. These utility functions measure the interestingness of a view from different aspects, and need to be considered at the same time to reach a reasonable and accurate assessment. These aspects could include the relevance of the view, conciseness of the pattern, generality of the pattern, and so on.

Several works [2, 5, 10] have adopted a view interestingness measure that involves multiple utility functions. However, all of them assume that the view interestingness is a linear combination of individual utility functions. This assumption might be suitable in specific scenarios, but is usually not accurate in a more general sense. For example, *deviation* [7] is a commonly-used utility function. However, a view with high deviation could be uninteresting if the context of the view is not relevant to the analytical task. Similarly, a relevant view could also be uninteresting due to the lack of deviation. It can be easily seen from the example that the interestingness of a view is not a linear combination of the two measures (i.e., deviation and relevance).

Since any assumption regarding the utility measure composition in the view interestingness could be inaccurate without verification from real users, user studies that record and analyze real user assessment of the view interestingness become highly needed.

In light of the above demand, in this work, we propose a graphical user interface (GUI) that is designed to be used in a user study to shed light on the composition of view interestingness.

Contributions Specifically, the contributions of this paper are the following:

- Create a classification system for view recommendation tasks, and identify utility measures suitable for each task category.
- Design a GUI that uses the identified utility measures to discover how the users evaluate the views with respect to the utility measures, and how the users assess the overall view interestingness based on the utility measures.
- Illustrate an example of how a general composition form of view interestingness could be derived from users answers to the questions in our GUI.

Outline The rest of the paper is structured as follows. Section 2 covers the background of the paper. Section 3 introduces our novel classification system and the utility measures suitable for each category. Section 4 presents our proposed GUI. Section 5 gives an example to illustrate how the user feedback can be used to discover the composition of view interestingness. Section 6 concludes the paper and discusses future works.

2 BACKGROUND

In this section, we present the necessary background details of our work. Specifically, we discuss how views can be constructed through SQL queries, and explain how utility functions can be used to recommend views.

2.1 Views & Data Visualization

To begin, we start by describing a view (i.e., histogram or bar chart) in the context of structural databases. A view v_i essentially represents an SQL query with a group-by clause over a database D [2, 10]. Under the typical multi-dimensional data models, data can be modeled as a set of measure attributes $M = \{m_1, m_2, m_3, ...\}$ and a set of dimension attributes $A = \{a_1, a_2, a_3, ...\}$. The measure attributes (e.g., number of items sold) is the set of attributes that contain measurable value and can be aggregated. The dimensional attributes (e.g., brand, year, color, size) is the set of attributes on which measure attributes are viewed. To formulate an SQL query with a group-by clause, we need to have a set of aggregation functions $F = \{f_1, f_2, f_3, ...\}$. Thus, we can represent each view v_i as a triple (a, m, f), such that one dimension attribute a is applied to one aggregation function f on the corresponding measure attribute m. Consequently, the View Space (VS), i.e., the total number of possible views is:

$$VS = |A| \times |M| \times |F| \tag{1}$$

Clearly, VS can be very large, especially with high-dimensional data.

2.2 View Recommendation

In order to recommend the set of k most interesting views from a large number of views, utility scores are required to rank all the views. To compute such utility scores, existing literature have proposed a large number of utility functions, some commonly used ones includes deviation [7], diversity [4], usability [2]. A utility function $u()$ maps a view to a real number indicating the interestingness of the view.

Definition 2.1. (*View Recommendation Problem*) Given a database D, a utility function $u()$, and the size of the preferred view recommendations k, find the top-k views $v_1, v_2, ..., v_k$ constructed from D that have the highest utilities according to $u()$ among all possible views.

The definition is straightforward. However, as mentioned earlier, using a single utility function to estimate the interestingness of a view is usually not accurate, because the interestingness is usually determined by a combination of multiple utility functions simultaneously. This observation leads to a refined definition.

Definition 2.2. (*View Recommendation with Composite Utility Function*) Given a database D, a utility function $u()$ that is a composite of a set of n utility functions $U = \{u_1, u_2, ..., u_n\}$, and the size of the preferred view recommendations k, find the top-k views $v_1, v_2, ..., v_k$ constructed from D that have the highest utilities according to $u()$ among all possible views.

It can be seen that the composition of the utility function $u()$ in Definition 2.2 plays an important role in the view recommendation problem. Recent works [2, 5, 10] have suggested that the utility functions $U = \{u_1, u_2, ..., u_n\}$ are linearly combined to form the composite utility function $u()$. In other words, they adopt the following form of a composite utility function:

$$u() = \beta_1 u_1() + \cdots + \beta_n u_n() \tag{2}$$

where β_i are the weights assigned to the corresponding utility functions $u_i, i = 1, ..., n$. The weights β_i can either be specified by the user, the system [2], or discovered during the interactive recommendation process through user feedback [10].

As previously mentioned, the above assumption that the view interestingness can be represented by a linear combination of utility functions might be suitable in specific scenarios, but is usually not accurate in a more general sense. Therefore, in this paper, we are going to introduce a user study GUI design that aims to shed light on a more general form of the composition of view interestingness.

3 THE CLASSIFICATION SYSTEM

In this section, we introduce a new classification system for view recommendation tasks, and identify utility measures suitable for each category. The identified utility measures will be used in the designed GUI (Section 4) to discover the view interestingness composition.

3.1 Classification Dimensions

The classification system has two dimensions. The first dimension is based on the exploration nature of the task, so we call it the *exploration dimension*. It has two categories: *exploratory* and *targeted*. A view recommendation task is exploratory if the user does not have a highly specific analytical goal in mind, and wants to discover as many interesting views from the data as possible (e.g., find interesting views from the census data). On the contrary, a view recommendation task is targeted if the user has a highly specific analytical goal in mind, and is looking for specific views based on the goal (e.g., find interesting views about financial situations across different work classes from the census data).

In fact, it can be seen that the exploration dimension has a continuous domain, because any view recommendation task is between the most exploratory (i.e., all possible views are candidates for interesting views) and the most targeted (i.e., only one view is the candidate for interesting views). However, for the sake of the classification purpose, we have discretized the exploration dimension into two categories, as mentioned above.

The second dimension is based on the comparison nature of the task, so we call it the *comparison dimension*. It has two categories: *non-comparative* and *comparative*. A view recommendation task is non-comparative if one data subset is involved in the views. For example, a non-comparative task could be finding interesting views about female participants from the census data. The data subset that is involved in the views is the female population. A view recommendation task is comparative if two or more data subsets are involved in the views, and can be compared in the views. For example, a comparative task could be finding interesting views about the difference between female and male participants from the census data. The two data subsets that are involved in the views are the female population and the male population.

The two dimensions together form four categories for view recommendation tasks: *exploratory non-comparative, exploratory comparative, targeted non-comparative*, and *targeted comparative*.

3.2 Utility Measures

In this part, we discuss different utility measures and the categories of view recommendation tasks that they are suitable for. The utility measures will be used in the GUI to allow the user to evaluate the view with respect to them. Concrete examples for the user evaluation of the utility measures based on an example view recommendation task will be given in Section 4 during the introduction of the GUI.

We consider six utility measures in this section.

Novelty *Novelty* measures how unfamiliar the user is with the context of the view. The *context* of a view is the information carried by the view other than the aggregate values. Recall that a view can be thought of as the query result of applying an aggregation function on a measure attribute and grouping the aggregate values by a dimension attribute over a data subset. Based on the above observation, the context of a view could include the data subset, the dimension attribute, the measure attribute, and the aggregation function.

If we assume that the previously recommended views become the knowledge of the user, then novel views will gradually help the user explore unknown or unfamiliar regions of the view space, thus helping increasing the *diversity* [6] of the recommendation set.

The measure of Novelty is especially useful in exploratory view recommendation tasks, because it can help the user explore the view space quickly and comprehensively.

Relevance *Relevance* [6] measures how relevant the context of the view is to the user's analytical goal. The information contained in the context of the view, such as the data subset, the dimension attribute, the measure attribute, and the aggregation function could play a role in the user's determination of Relevance.

Relevant views are helpful, because they may contain information that can help the user reach the analytical goal. The measure of Relevance is especially useful for targeted view recommendation tasks, because it can help the user quickly locate the targeted regions in the view space.

Conciseness *Conciseness* [4] measures the easiness for the user to perceive and remember the patterns in the view. The context of the view, the number of groups in the view, the order of the groups, and the value pattern in each group are among the factors that could affect the user's determination of Conciseness.

Concise views are helpful because they do not overwhelm the user with a large amount of information, and can be easily perceived and remembered. Conciseness is useful in all four categories of view recommendation tasks.

Diversity *Diversity* [4] measures the perceived fluctuation of the aggregate value across the groups. The number of groups in the view, the order of the groups, and the aggregate value in each group are among the factors that could affect the user's determination of the diversity.

Views with high diversity are helpful because they indicate a high correlation between the dimension attribute and the aggregate values. This measure is more useful for non-comparative tasks, where the views only contain the aggregate values of one data subset.

Deviation *Deviation* measures the perceived fluctuation of the within-group aggregate value difference across the groups. This measure is a simplified form of the *Deviation* measure in [7], in a way that this measure does not normalize the aggregate values of each data subset into a distribution. We adopt this simplified form based on the fact that it is very difficult for the user to imagine the normalized aggregate values without the assistance of any helper views.

The number of groups in the view, the order of the groups, and the aggregate value difference between the data subsets in each group are among the factors that could affect the user's determination of the deviation.

		Comparison Dimension	
		Non-Comparative	**Comparative**
Exploration Dimension	**Exploratory**	Novelty Conciseness Diversity Generality	Novelty Conciseness Deviation Generality
	Targeted	Relevance Conciseness Diversity Generality	Relevance Conciseness Deviation Generality

Table 1: View Recommendation Task Categories and Suitable Utility Measures for each Category

Views with high deviation are helpful because they indicate a high correlation between the dimension attribute and the within-group aggregate value differences. This measure is more useful for comparative tasks, where the views contain the aggregate values of two or more data subsets.

Generality *Generality* [4] measures how well, does the user thinks, the perceived patterns in the view can be generalized to the whole data subset.

Coverage information of the view and of the bars in the view are among the factors that could affect the user's determination of generality. The coverage of a view is the percentage of the number of records covered by the view against the number of records in the data subset. A view does not cover a given record if any of its dimension attribute or measure attribute is missing. Similarly, the coverage of a bar in the view (i.e., an aggregate value bar) is the percentage of the number of records covered by the bar against the number of records in the data subset. A bar does not cover a record if its dimension attribute does not belong to the group of the bar or its measure attribute is missing.

The higher the coverage, the higher the generality of the pattern will be. Views with high generality are useful, because the patterns in the view are more likely to be valid in the whole data subset as well. Generality is useful in all four categories of view recommendation tasks.

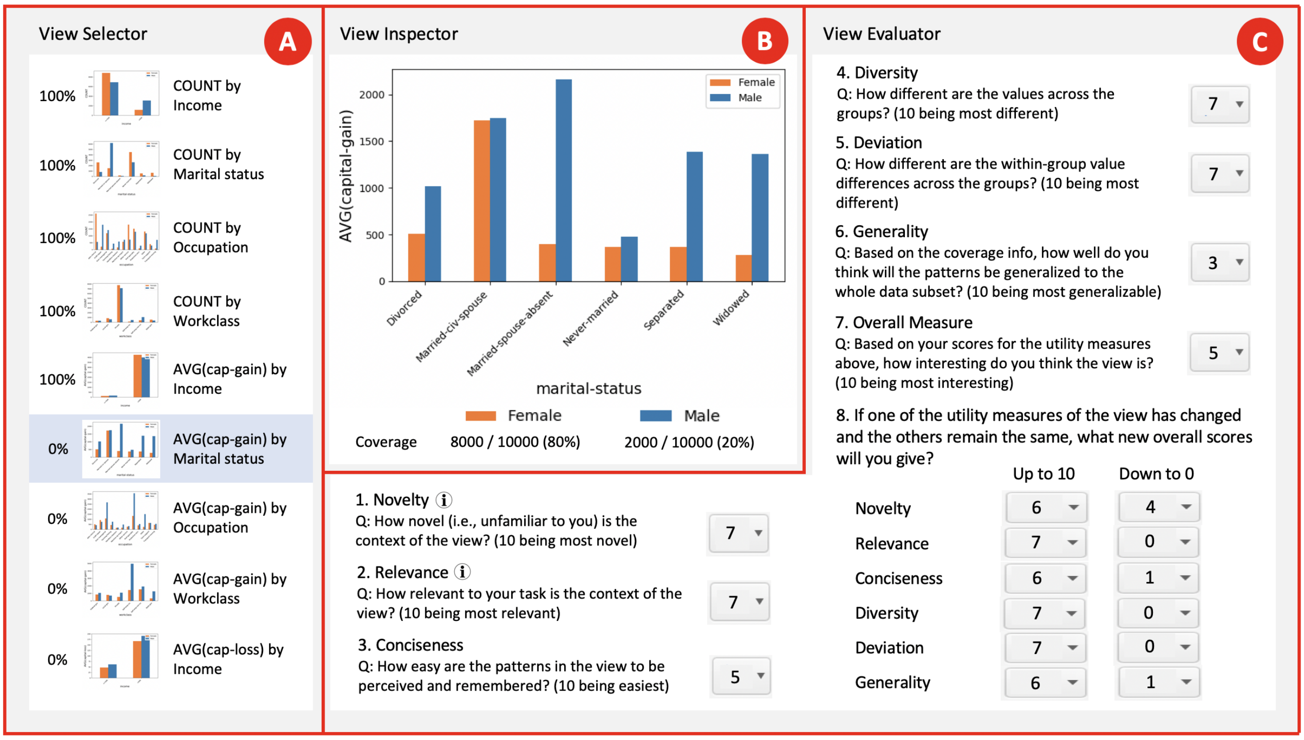
A summary of the four categories of view recommendation tasks and the utility measures suitable for each category is illustrated in Table 1.

4 THE GUI DESIGN

In this section, we discuss the details of our designed GUI that will allow users to evaluate the views with respect to the utility measures and the overall interestingness.

The GUI will reside on a web application and be used in a user study. The frontend of the web application will be developed using JavaScript, and the backend will be developed using Java.

An example view recommendation task will be used in this section to facilitate the GUI demonstration. The example task is to discover the difference between female and male participants in the census data. In other words, the user's task is to find out interesting views that show large differences between the aggregate values of the female and the male population. It can be seen that this is an *exploratory comparative* task. So the utility measures suitable for the task are Novelty, Conciseness, Deviation, and Generality. However, in order to demonstrate the measures



100% COUNT by Income

100% COUNT by Marital status

100% COUNT by Occupation

100% COUNT by Workclass

100% AVG(cap-gain) by Income

0% AVG(cap-gain) by Marital status

0% AVG(cap-gain) by Occupation

0% AVG(cap-gain) by Workclass

0% AVG(cap-loss) by Income

Coverage 8000 / 10000 (80%) 2000 / 10000 (20%)

1. Novelty ⓘ
Q: How novel (i.e., unfamiliar to you) is the context of the view? (10 being most novel) 7

2. Relevance ⓘ
Q: How relevant to your task is the context of the view? (10 being most relevant) 7

3. Conciseness
Q: How easy are the patterns in the view to be perceived and remembered? (10 being easiest) 5

4. Diversity
Q: How different are the values across the groups? (10 being most different) 7

5. Deviation
Q: How different are the within-group value differences across the groups? (10 being most different) 7

6. Generality
Q: Based on the coverage info, how well do you think will the patterns be generalized to the whole data subset? (10 being most generalizable) 3

7. Overall Measure
Q: Based on your scores for the utility measures above, how interesting do you think the view is? (10 being most interesting) 5

8. If one of the utility measures of the view has changed and the others remain the same, what new overall scores will you give?

	Up to 10	Down to 0
Novelty	6	4
Relevance	7	0
Conciseness	6	1
Diversity	7	0
Deviation	7	0
Generality	6	1

Figure 1: GUI Overview

of Relevance and Diversity, we will include them in the GUI introduction as well. We will use a different view recommendation task when introducing Relevance and Diversity.

The GUI has three parts: *View Selector, View Inspector*, and *View Evaluator*. The overview of the GUI is shown in Figure 1.

4.1 View Selector

The View Selector pane (Figure 1 Pane A) is the starting point of the user workflow. The pane contains a list of views, each of which is in the form of a bar chart or histogram. As mentioned before, each view is generated by applying an aggregation function F on a measure attribute M and grouping the result by a dimension attribute A by a back-end database server.

Besides a thumbnail of the view, each entry in the pane also includes a description and a progress indicator. The description contains the information about the aggregation function, measure attribute, and dimension attribute used to generate the view. Examples of the description are "COUNT by occupation" and "AVG(capital-gain) by work class". The progress indicator is in the form of a percentage number, indicating the completion percentage of the questions for a specific view. The entry that is being selected will have a light blue background, for example the 6th one in Figure 1, to help the user identify the view that is being inspected.

4.2 View Inspector

After the user selects an entry in the View Selector, the selected view will be displayed in the View Inspector (Figure 1 Pane B).

The example view in the View Inspector is in the form of a bar chart, and shows the average capital gain of the female and male population across the different marital statuses.

The x-axis is for the dimension attribute (i.e., the attribute by which the result is grouped). The x-axis label is the name of the dimension attribute. The x-axis tick labels are the distinct values of the dimension attribute (i.e., group names).

The y-axis is for the aggregate values. The y-axis label is a combination of the name of the aggregation function and the name of the measure attribute (i.e., the attribute on which the aggregation function is applied). The y-axis label in the example is "AVG(capital-gain)". The y-axis tick labels are value indicators for the y-axis grid lines.

Another component of the view is the legend, which identifies the two data subsets (i.e., population) being compared. The two subsets in the example are "Female" and "Male".

The main content of the view is the aggregate values across the groups of the two data subsets. There are two bars in each group. The left one is for the first subset, while the right one is for the second subset. The two bars are in different colors, so that the user can easily distinguish between the two.

The last part of the View Inspector is the additional information section. It contains information that cannot be easily embedded in the main view. One example of such information is the coverage information, as discussed in Section 3. For each subset, the coverage information contains three numbers: the number of records that the view covers, the number of all records in the subset, and the percentage of the former against the latter.

If the cursor hovers over one of the bars in the view, the corresponding aggregate value as well as the coverage information for that specific bar (i.e., the number of records covered by the bar) will be displayed in the form of a tooltip. The tooltip helps the user to get a precise reading of the aggregate value and an understanding of the coverage information at a finer granularity level.

4.3 View Evaluator

The third pane of the GUI is the View Evaluator (Figure 1 Pane C). It contains eight questions for the user to evaluate the selected view.

Each question has a title, a description, and a score selector. The title indicates the utility measure being evaluated. The description describes the utility measure and rating rules. The score selector is in the form of a dropdown menu. The options are integers between 0 and 10, inclusive. The rating rules are set such that the higher the score, the more interesting the view is with respect to the utility measure being evaluated. Term definitions will be displayed when the cursor hovers over the information icon of the questions containing the term. For example, the definition of the context of the view will be displayed for the questions of Novelty and Relevance.

4.3.1 Utility Measures. The first six questions are designed to discover how the user would evaluate the view with respect to the utility measures discussed in Section 3.

Based on the *affecting factors* identified for each utility measure in Section 3, each of the first six questions can be used to discover which factor(s) the user uses to assess the view with respect to the corresponding utility measure. We will introduce the questions and the example feedback based on the example view in Figure 1 for the first six questions in the following.

Novelty The question reads: "How novel (i.e., unfamiliar to you) is the context of the view? (10 being most novel)". If we assume that the user is not very familiar with the capital gains of the two population across different marital status, then the user could give a high score, indicating that the context of the view is novel to her.

Relevance We temporarily change the view recommendation task to a *targeted* task to make Relevance suitable for the task. The new task is to discover differences in financial situations between female and male participants across different marital status groups.

The question for Relevance reads, "How relevant to your task is the context of the view? (10 being most relevant)". Since the capital gain is an indicator of financial situations, the view is very relevant to the task, and the user could give a high score.

Conciseness The question reads, "How easy are the patterns in the view to be perceived and remembered? (10 being easiest)". Assume that the user tries to remember the patterns in the view in the following way. "For the never-married and the married with the couple living together, the capital gains of the two populations are similar. For the married with the couple not living together due to various reasons, capital gain of the female population is less than that of the male population."

The above pattern is easy to remember, but it requires some effort from the user to come up with a plan to group the original marital status groups. Therefore the user could give a medium score, indicating that some effort was required from her to perceive and understand the patterns in the view.

Diversity We temporarily change the view recommendation task to a *non-comparative* task to make Diversity suitable for the task. The new task is to discover interesting views for male participants in the census data. In the new task, only the blue bars will remain in the view.

The question for Diversity reads, "How different are the values across the groups? (10 being most different)". It can be seen that the value in the blue bar fluctuates a lot across the groups, so the user could give a high score for Diversity.

Deviation The question reads, "How different are the within-group value differences across the groups? (10 being most different)". It can be seen that the value difference between the two subsets in each group fluctuates a lot across the groups, so the user could give a high score for Deviation.

Generality The question reads, "Based on the coverage info, how well do you think the patterns will be generalized to the whole data subset? (10 being most generalizable)". The coverage information refers to the coverage percentage of the view and the individual bars. The coverage of the male population is quite low at 20%. Besides, the tooltips show that the coverage for the "Married-spouse-absent", "Separated", and "Widowed" groups of the male population are also very low. Therefore, the user could give a low score for Generality, indicating that she thinks that the patterns in the view are not very likely to be valid in the whole data subset.

4.3.2 Overall Interestingness. Questions 7 and 8 are designed to discover the utility measure composition of the overall view interestingness. In other words, they are designed to find out how each utility measure affects the overall view interestingness.

Question 7 reads, "Based on your scores for the utility measures above, how interesting do you think the view is? (10 being most interesting)". The question allows the user to consider carefully and comprehensively the different aspects of the view before providing an overall score for the view interestingness.

Question 8 reads, "If one of the utility measures of the view has changed and the others remain the same, what new overall scores will you give?". Two scenarios have been designed regarding how each utility measure could change. They are "Up to 10" and "Down to 0" (i.e., the utility measure goes up to its maximum value and down to its minimum value). These questions are designed to discover how the changes of each utility measure affect the overall interestingness.

5 COMPOSITION DISCOVERY

In this section, we discuss how the answers to the questions in our proposed GUI could shed light on the general composition form of view interestingness. Specifically, we will illustrate a potential composition form based on user feedback in the example below. For the sake of simplicity, we will refer to any specific utility measure as M and the view interestingness as I in the example.

After the user answers all the questions for a view, for each M, we will get three readings of I for three M values. The first M value (i.e., m_1) is the answer to the corresponding question for M in Questions 1 to 6. The corresponding I value (i.e., i_1) is the answer to Question 7. The second M value (i.e., m_2) is always 10, which corresponds to the "Up to 10" question for M in Question 8. The corresponding I value (i.e., i_2) is the answer to the above "Up to 10" question. The third M value (i.e., m_3) is always 0, which corresponds to the "Down to 0" question for M in Question 8. The corresponding I value (i.e., i_3) is the answer to the above "Down to 0" question. We assume that $0 < m_1 < 10$, $i_3 < i_1 < i_2$ based on user feedback. For instance, Table 2 shows the values of m_i and i_i, $i = 1..3$, for the example feedback for the view in Figure 1.

Utility Measure	m_1	i_1	m_2	i_2	m_3	i_3
Novelty	7	5	10	6	0	4
Conciseness	5	5	10	6	0	1
Deviation	7	5	10	7	0	0

Table 2: Example User Feedback

The discovery of the composition form is divided into two steps: 1) determine the *basic form* of each M in I, 2) *refine* the basic form of each M.

Basic form: Firstly, we use the value of $|i_2 - i_3|$ to determine the *basic form* of each M in I. We list some possible conditions for $|i_2 - i_3|$, which are not exhaustive:

- If $|i_2 - i_3|$ is large (e.g., larger than $20/3$) and $i_3 = 0$, which means that M has a large influence on I and will bring I to zero when it drops to zero, then we call M a type-A measure. According to Table 2, Deviation ($|i_2 - i_3| = 7$, $i_3 = 0$) could be a type-A measure.
- If $|i_2 - i_3|$ is moderate (e.g., between $10/3$ and $20/3$) and $i_3 > 0$, which means that M has a moderate influence on I, but will not bring I to zero when it drops to zero, then we call M a type-B measure. According to Table 2, Conciseness ($|i_2 - i_3| = 5$, $i_3 = 1$) could be a type-B measure.
- If $|i_2 - i_3|$ is small (e.g., smaller than $10/3$) and $i_3 > 0$, which means that M has a small influence on I and will not bring I to zero when it drops to zero, then we call M a type-C measure. According to Table 2, Novelty ($|i_2 - i_3| = 2$, $i_3 = 4$) could be a type-C measure.

If the above three types can cover all utility measures based on user feedback, then we could build a potential form of I as follow:

$$I = \left(\prod_{i=1}^{n} A_i\right)\left(\prod_{i=1}^{n}(B_i + w_{bi})\right)\left(\sum_{i=1}^{n} C_i + w_c\right) \tag{3}$$

where A_i's are type-A measures, B_i's are type-B measures, C_i's are type-C measures, and w_{bi}'s and w_c are constants. For the sake of simplicity, we have omitted any non-essential scaling or offset constants in Equation 3.

It can be seen that the behaviors of the measures in Equation 3 satisfy the conditions for the corresponding types. In other words, if all utility measures are of similar values, then the influence of A_i (i.e., the gradient: $\partial I/\partial A_i$) is larger than that of B_i, which in turn is larger than that of C_i. Besides, any A_i's dropping to zero will bring I to zero, while the drops of any B_i or C_i will not.

Refined form: Secondly, we use all three points of each M to refine the basic form of M in I. We introduce a simple refinement below, which is the addition of *influence change rate* (i.e., gradient change rate) to the basic form.

If we call the three value pairs for each M as $X(m_1, i_1)$, $Y(m_2, i_2)$, and $Z(m_3, i_3)$, we can draw the three points onto an MI-plane. The three points for Deviation is illustrated in Figure 2.

If we assume that the relationship between I and each M can be approximated as exponential in the form of $I = M^e$, then we will be able to use curve fitting to determine the exponent e for each M. For example, the three points in Figure 2 are almost on a line, and thus the exponent e for Deviation would be close to 1.

Then we can refine Equation 3 by adding the exponential relationship information for each M to get:

$$I = \left(\prod_{i=1}^{n} A_i^{e_{ai}}\right)\left(\prod_{i=1}^{n}(B_i^{e_{bi}} + w_{bi})\right)\left(\sum_{i=1}^{n} C_i^{e_{ci}} + w_c\right) \tag{4}$$

where e_{ai}, e_{bi}, e_{ci} are the exponents indicating the influence change rates of the utility measures.

An exponent between 0 and 1 means that the exponential curve is concave. In other words, the influence of the utility measure is larger in the lower part of its domain, and smaller in the higher part of its domain. Similarly, an exponent close to 1 means that the influence of the utility measure is approximately consistent throughout its domain. An exponent larger than 1 means that the influence of the utility measure is smaller in the lower part of its domain, and larger in the higher part of its domain.

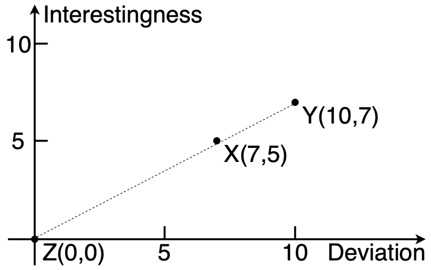


Figure 2: Example Interestingness Readings for Deviation

6 CONCLUSION

In this work, we first create a novel classification system for view recommendation tasks and identify utility measures suitable for each task category. Then, we design a GUI that uses the identified utility measures to discover how the users evaluate the view with respect to the utility measures and how the users assess the overall view interestingness based on the utility measures. Finally, we use an example to illustrate how user answers to the questions in the GUI can be used to discover the composition form of view interestingness.

In the future, we plan to conduct a user study using the proposed GUI, collect and analyze the user feedback, and discover a general composition form of view interestingness. The discovered interestingness composition form will be shared with the community to advance the development of view recommendation technologies.

REFERENCES

[1] [n. d.]. Tableau public. http://public.tableau.com. ([n. d.]). Accessed: 2019-02-09.

[2] Humaira Ehsan, Mohamed A. Sharaf, and Panos K. Chrysanthis. 2016. MuVE: Efficient Multi-Objective View Recommendation for Visual Data Exploration. In *IEEE ICDE.*

[3] H. Ehsan, M. A. Sharaf, and P. K. Chrysanthis. 2018. Efficient Recommendation of Aggregate Data Visualizations. *IEEE Transactions on Knowledge and Data Engineering* 30, 2 (2018), 263–277.

[4] Liqiang Geng and Howard J. Hamilton. 2006. Interestingness measures for data mining: A survey. *ACM Comput. Surv.* 38, 3 (2006), 9. https://doi.org/10.1145/1132960.1132963

[5] Rischan Mafrur, Mohamed A. Sharaf, and Hina A. Khan. 2018. DiVE: Diversifying View Recommendation for Visual Data Exploration. In *ACM CIKM.*

[6] Manasi Vartak, Silu Huang, Tarique Siddiqui, Samuel Madden, and Aditya G. Parameswaran. 2016. Towards Visualization Recommendation Systems. *ACM SIGMOD Record* 45, 4 (2016), 34–39.

[7] Manasi Vartak, Sajjadur Rahman, Samuel Madden, Aditya G. Parameswaran, and Neoklis Polyzotis. 2015. SEEDB: Efficient Data-Driven Visualization Recommendations to Support Visual Analytics. *PVLDB* 8, 13 (2015), 2182–2193.

[8] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2016. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization & Computer Graphics* 1 (2016), 1–1.

[9] Xiaozhong Zhang, Xiaoyu Ge, and Panos K. Chrysanthis. 2019. Leveraging Data-Analysis Session Logs for Efficient, Personalized, Interactive View Recommendation. In *IEEE International Conference on Cognitive Machine Intelligence.*

[10] Xiaozhong Zhang, Xiaoyu Ge, Panos K. Chrysanthis, and Mohamed A. Sharaf. 2019. ViewSeeker: An Interactive View Recommendation Tool. In *Proceedings of the EDBT 2019 BigVis Workshop.*