

Duet: Helping Data Analysis Novices Conduct Pairwise Comparisons by Minimal Specification

Po-Ming Law, Rahul C. Basole, and Yanhong Wu

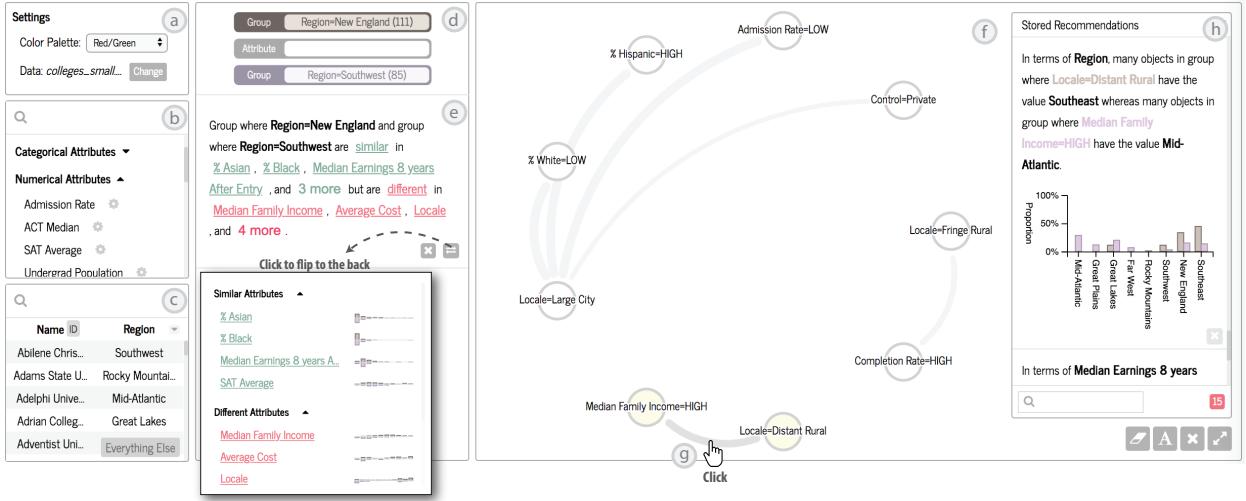


Fig. 1. The Duet interface consists of a settings menu (a) an attribute list (b), a small table (c), group and attribute shelves (d), a result panel (e), and a relationship map (f). The user is comparing New England colleges and Southeast colleges (d) and the similar and different attributes are shown in the result panel (e). She is revisiting the stored recommended attributes by clicking on a link between two nodes (g). The stored attributes are displayed in the stored recommendation panel (h).

Abstract—Data analysis novices often encounter barriers in executing low-level operations for pairwise comparisons. They may also run into barriers in interpreting the artifacts (e.g., visualizations) created as a result of the operations. We developed Duet, a visual analysis system designed to help data analysis novices conduct pairwise comparisons by addressing execution and interpretation barriers. To reduce the barriers in executing low-level operations during pairwise comparison, Duet employs minimal specification: when one object group (i.e. a group of records in a data table) is specified, Duet recommends object groups that are similar to or different from the specified one; when two object groups are specified, Duet recommends the similar and different attributes between them. To lower the barriers in interpreting its recommendations, Duet explains the recommended groups and attributes using both visualizations and textual descriptions. We conducted a qualitative evaluation with eight participants to understand the effectiveness of Duet. The results suggest that minimal specification is easy to use and Duet’s explanations are helpful for interpreting the recommendations despite some usability issues.

Index Terms—Pairwise comparison, novices, data analysis, automatic insight generation

1 INTRODUCTION

Pairwise comparison is imperative to decision making. From consumers choosing between smartphone models to high school students comparing colleges, pairwise comparison shapes many of the decisions we make. The increasing availability of a wide range of data is creating a tremendous opportunity for diverse users to make more informed and potentially better decisions by comparing various aspects of objects and object groups. Yet, while many online tools have been developed to facilitate pairwise comparison (e.g., [2, 4]), it has remained challenging for data analysis novices.

Consider the following example. The New York City Police Department stopped approximately 500,000 pedestrians for suspected criminal involvement in 2006 alone [41]. 89% of the stops involved non-whites, indicating large racial disparities [41]. Knowing the domain very well but not necessarily equipped with the right skills for data analysis, how does a data analysis novice like a police captain analyze the similarities

and differences between stopped white and non-white suspects? In answering the question, the police captain needs to find a strategy and map the strategy to the tools available. For instance, he might have decided to create visualizations (the strategy) to understand whether the stopped white and non-white suspects have different rates of frisk, and use of force and arrest. He might then perform a series of operations in a visualization tool to construct the desired visualizations (mapping the strategy to the tool). While it may seem easy to experts in data analysis, due to lack of experience, it is likely that the police captain does not have a good sense of what strategies to use and how to map the strategy to the low-level operations in the tool. The difficulties in pairwise comparison are compounded by the inherent complexity of comparison. Although it is well-known that visualizations amplify our cognition during data analysis [12], comparison is still considered challenging with the aid of visualizations: the items for comparison and the relationships between them can be highly complicated [25]. In short, lack of experience in data analysis and the complex nature of comparison make pairwise comparison difficult for data analysis novices.

In developing a system to help data analysis novices conduct pair-

• Po-Ming Law and Rahul C. Basole are with Georgia Institute of Technology. E-mail: {pmlaw, basole}@gatech.edu.

• Yanhong Wu is with Visa Research. E-mail: yanwu@visa.com.

wise comparisons, we identified from the literature two barriers that novices would have in answering a question during data analysis: execution barrier [13, 15, 27, 35, 39] and interpretation barrier [10, 11, 27, 36]. For the police captain mentioned in the previous paragraph, determining what visualizations to create (finding the right strategy) and how to perform low-level operations in a visualization tool to create the visualizations (mapping the strategy to the tools available) constitute the execution barrier in his analysis. If he succeeds in overcoming the execution barrier, there will still be one barrier awaiting him: how to interpret the visualizations and connect the visualizations to the questions to see if they answer the questions at all? After performing a series of operations using a tool, some artifacts (e.g., visualizations) are obtained. Novices would grapple with making sense of the artifacts obtained and relating the artifacts to the questions of interest. This barrier pertains to interpretation. Systems that aim to support novices in data analysis should address both execution and interpretation barriers.

We explore an approach based on the characteristics of knowledge gap in answering a pairwise comparison question: while data analysis novices may not know what strategies to adopt to answer a question and how to map the strategies to the available tools, they do know what object groups they want to compare (e.g., the stopped white and non-white suspects in the case of the police captain). If, after novices minimally specify the object groups of interest, insights about similarities and differences (e.g., the similar and different attributes between the stopped white and non-white suspects) could be automatically generated, they would not have to dive into the details of what strategies to use and how to carry out low-level operations to execute the strategies for question answering.

In this paper, we present Duet, a visual analysis system that facilitates data analysis novices to conduct pairwise comparisons of object groups in tabular data. We address the execution barrier during pairwise comparison by the minimal specification technique, by which novice analysts only need to specify object groups of interest for pairwise comparison. When one object group is specified, Duet infers and recommends object groups that are similar to or different from the specified one; when two object groups are specified, Duet recommends the similar and different attributes between them. Minimal specification is therefore a better match for what novice analysts know (the object groups of interest) and what they might not know (the strategies to look for the answer and the steps required to execute the strategies). We address the barrier in interpreting Duet’s recommendations by explaining them using both visualizations and textual descriptions. To understand the effectiveness of Duet, we conducted a qualitative study with eight data analysis novices. The findings suggest that minimal specification is easy to use and that Duet’s explanations are helpful for interpreting the recommendations in spite of some usability issues. Our work demonstrates how visualizations and automatic insight generation, together, can help novices conduct data analysis.

2 RELATED WORK

2.1 Execution and Interpretation Barriers

The barriers people encountered in data-driven question answering are well-investigated by the visualization community. Prior work identified two major barriers: the execution barrier and the interpretation barrier.

Execution barrier concerns the difficulties in determining what strategies to use and how to carry out low-level operations to answer a question during data analysis. A number of studies have been conducted to elucidate the challenges in translating a question into the operations in a tool for answering the question. Grammel et al. [27] found that, during visualization construction, InfoVis novices faced difficulties in selecting relevant data attributes based on their higher-level questions and in transforming the selected data into visual representations that support answering their questions. Kwon et al. [15] observed that, while using Jigsaw in an investigative analysis scenario, novice investigators struggled to choose appropriate views and execute an appropriate set of interactions in a view to retrieve information needed for question answering. These findings in lab studies are supported by other InfoVis and HCI researchers. Lam’s framework of interaction costs includes costs to translate a question in mind into system operations [35]. In

their model of guidance in visual analytics, Ceneda et al. [13] defined the notion of knowledge gap that exists when analysts do not know what to do to achieve a desired goal. Indeed, the difficulties in executing low-level operations to achieve a goal, such as answering a question, is so well-known in the HCI community that it is termed the gulf of execution [39]. Minimal specification employed by Duet addresses execution barrier. It enables users to specify only the object groups of interest for pairwise comparison, making it a better match for what users know (the object groups of interest) and what they might not know (the strategies and the system operations).

Interpretation barrier pertains to the difficulties involved in interpreting the artifacts (e.g., visualizations) created after performing a series of low-level operations. The term “interpretation barrier” was first used by Grammel et al. [27] in depicting the problems encountered by InfoVis novices when they tried to make sense of a visualization and answer their questions using the visualization. In a related study, Lee et al. [36] investigated how novices make sense of unfamiliar visualizations and constructed a grounded-theory model to depict the experience of InfoVis novices in dealing with the barrier in interpreting visualizations. Interpretation barrier is also described by Amar and Stasko [10] as the worldview gap that refers to the difficulties in relating visualizations to high-level analytic goals. The artifacts users create to answer their questions are not limited to visualizations. For instance, to understand the similarities and differences among objects, analysts can apply a clustering algorithm to generate clusters of objects. Cao et al. described the challenges in interpreting clustering results [11]. In response, they proposed a visualization technique to help analysts interpret multidimensional clusters. When Duet offers a recommendation to users, users may grapple with understanding why the recommendation is provided. To reduce the barrier to interpreting its recommendations, Duet explains them using both visualizations and textual descriptions.

2.2 Pairwise Comparison Tools

There exists a plethora of tools and techniques that support pairwise comparison. Experimenters often use statistical tools such as SPSS to understand whether two group means are significantly different. Tukey’s HSD test [46] is a post-hoc analysis technique to spot a pair of groups with significantly different means. Lacking data analysis skills, novices would struggle to adopt these advanced statistical methods. Data analysts can also conduct pairwise comparisons using visualization systems like Tableau. However, Grammel et al. [27] demonstrated that novice users often encounter execution and interpretation barriers during visualization construction. Microsoft Excel is probably the most widely-used data analysis tools for novices. Albeit popular among data analysis novices, Excel requires users to write scripts or go over layers of menus to perform simple comparisons, creating execution barrier to novices. Finally, some online tools support pairwise comparisons of different objects (e.g., colleges [2], cities [4] and cars [1]). Despite their simplicity, their functionality is highly limited, restricting the variety of questions users can ask about their data. For instance, users can only compare two objects but not two object groups using these online tools.

2.3 Recommendation Systems for Data Analysis

Much research has been devoted to designing visualization recommendation systems [18, 19, 26, 31, 47–50]. There is also a recent attempt to computationally generate insights for recommendation [45]. Many of these systems recommend interesting visualizations (e.g., [19, 47]) or facts (e.g., [45]) based on some statistical properties of the data. They assume that users do not have specific questions in mind and would benefit from wandering in the space of insights. While some commercial tools such as Google Sheet [20] enable users to directly ask a question to obtain an answer, they only support a limited set of questions and are not tailored to pairwise comparison. Duet extends this line of research by using recommendations for reducing the barriers data analysis novices might encounter when answering pairwise comparison questions. Different from prior art, we contribute methods for recommending similar and different groups to a target group, and similar and different attributes between two target groups.

3 DESIGNING MINIMAL SPECIFICATION INTERFACE TO ADDRESS EXECUTION AND INTERPRETATION BARRIERS

Due to lack of experience in data analysis, novice analysts might encounter difficulties in translating pairwise comparison questions into low-level system operations. To reduce this *execution barrier*, techniques for novices should shield them from low-level operations. We designed minimal specification to be a **better match for what users know (the tasks users try to perform) and what they might not know (the low-level-operations to perform the tasks)**. Minimal specification enables users to minimally specify objects of interest (what users know when they ask a pairwise comparison question) rather than focusing on translating their questions into system operations (what users might not know) to obtain pairwise comparison results. As users specify object group(s), a system that employs minimal specification provides recommendations. For example, the system recommends similar and different attributes when users specify two object groups. The idea of a better match resonates with prior research in various sub-fields of HCI: programming by examples allows end users to provide examples of the text they want to extract (what end users know) so that they do not need to write programs to extract text from documents (what end users might not know) [30, 38, 51]; interrogative debugging let programmers debug their programs by asking why and why not questions about their programs failure (what programmers know) to shield them from mapping their debugging strategies to debugging tools' limited capabilities (what programmers might not know) [32, 33]; using natural language interfaces for visual data analysis, users can directly state their questions (what users know) without having to learn the interface or translate their questions into system operations (what users might not know) [24, 29, 42, 44]. Nevertheless, novices in data analysis may grapple with interpreting the recommendations offered when they specify object group(s) of interest. To reduce *interpretation barrier*, the **recommendations should be explained to users**. The idea of explaining recommendations echoes with research in Explainable Artificial Intelligence (XAI), which aims to promote trust and understanding of AI decisions by explanations [5, 9, 34, 37].

In designing an interface that supports pairwise comparison by minimal specification, we further derived four considerations from the literature:

C1. Enabling flexible definition of object groups. Flexibility has been identified as one of the major usability issues in visualization systems [22]. It is frustrating when users want to answer a question using a system but find that there is no way to do it due to the system's limited capabilities. As users may ask pairwise comparison questions about various kinds of object groups, they should be endowed with the capability of flexibly defining object groups with various characteristics.

C2. Shielding users from being overwhelmed by a large number of recommendations. A general issue with recommendation systems is that a large number of recommendations are often generated [47, 49]. When there are many recommendations, they should be displayed in a way that is less overwhelming to users.

C3. Allowing users to save interesting recommendations. Another issue with recommendation systems is that not all recommendations generated are meaningful to users [49]. The system should allow users to save interesting recommendations during their analysis. This enables users to share findings with their colleagues or retrieve the findings later for summarization at the end of analysis.

C4. Catering explanations for novices. Studies show that many laypeople have difficulties in understanding very simple bar and pie charts [23]. The system should cater the explanations for the recommendations to this group of users.

4 DUET'S USER INTERFACE

Duet's interface consists of five major components: an attribute list (Fig. 1b), a small data table (Fig. 1c), group and attribute shelves (Fig. 1d), a result panel (Fig. 1e) and a relationship map (Fig. 1f). To illustrate how Duet supports pairwise comparison by minimal specification and visualizations' role in minimal specification, consider the following usage scenario.

Ella is an international freshman studying in a US college. Not being knowledgeable about US colleges more broadly, she is curious about how different US colleges compare. She acquired online a college dataset that contains 1215 colleges, each consisting of 4 categorical attributes (e.g., Region) and 20 numerical attributes (e.g., Admission Rate). This dataset was derived from the College Scorecard Data released by the US Department of Education [3]. Ella would like to explore the data casually and share the findings with her friends. Upon loading the dataset, Duet displays the categorical and numerical attributes in the attribute list (Fig. 1b) and updates the small data table (Fig. 1c). The small data table allows users to view the raw data. It has only two columns due to space limitation. The first column always displays the ID of records (e.g., college name) and the second column displays another attribute, which can be changed by selecting an attribute from the attribute list.

4.1 Specifying One Object Group

Ella has a few questions in mind: Q1) What kinds of colleges are similar to those in New England, where Ivy League are located? Q2) Colleges that support continuing education tend to accept older students as freshmen. What kinds of colleges are different from these colleges for adult education? Q3) As she is studying in a college in the Southeast with a low admission rate, she wonders if there are other colleges that are similar to hers. 4) Finally, she wants to know what kinds of colleges are similar to her college with respect to tuition and admission rate. All these questions involve specifying *one* object group that satisfies some conditions (e.g., colleges where Region=*Southeast* and Admission Rate=*low*). Duet offers multiple ways for flexibly specifying object groups (C1):

Specifying a group that satisfies a categorical attribute-value pair. As Ella is interested in knowing what colleges are similar to the New England colleges (Q1), she specifies the colleges that have the value *New England* for the attribute Region. Users can select an attribute by clicking it on the attribute list and select a value by dragging it from the small table to a group shelf. When Ella clicks on Region in the attribute list, the second column of table is changed to Region (Fig. 2①). She then drags *New England* from the table to a group shelf to indicate her interest in New England colleges (Fig. 2②).

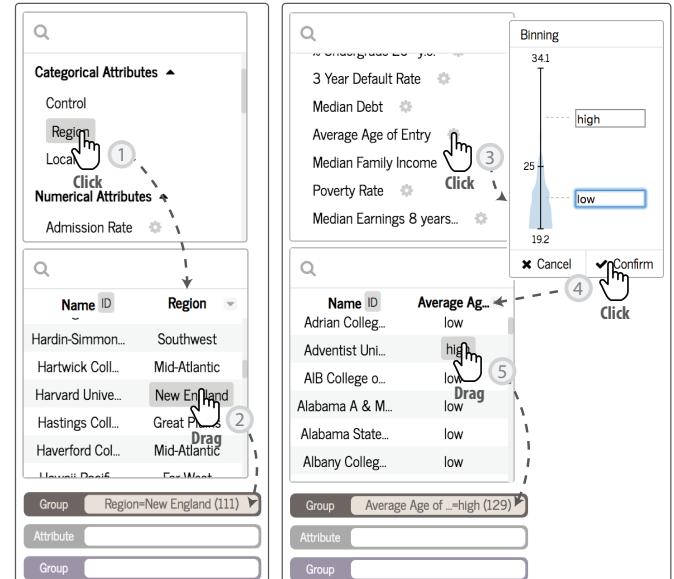


Fig. 2. Specifying one object group. Ella is specifying (left) the colleges that have the value *New England* for the attribute Region and (right) the colleges that have a value above 25 for the attribute Average Age of Entry.

Specifying a group that satisfies a numerical attribute-value pair. Ella considers the colleges with average age of entry higher than 25 as colleges for adults. To dive into the colleges that are different from these colleges (Q2), she specifies the colleges that have a value higher than 25 for the attribute Average Age of Entry. Ella click on next

to Average Age of Entry in the attribute list to open the binning widget (Fig. 2③), which uses a density plot to assist users to create custom bins. She labels the colleges with Average Age of Entry higher than 25 as “high” and the colleges with Average Age of Entry below 25 as “low”. When she clicks ✓ Confirm (Fig. 2④), the second column of the data table changes to Average Age of Entry and displays *low* (*high*) for colleges with Average Age of Entry below (above) 25. To specify the group of interest, she drags *high* from the table to a group shelf (Fig. 2⑤).

Specifying a group that satisfies multiple attribute-value pairs. Ella then investigates the colleges that are similar to Southeast colleges with low admission rates (what her college is like) (Q3). To specify colleges with the value *Southeast* for Region, she selects Region from the attribute list and drag *Southeast* from the table to a group shelf. To narrow to the Southeast colleges with low admission rate, she creates custom bins for the attribute Admission Rate using the binning widget and drags *low* from the data table to the same group shelf. By clicking on the group shelf, Ella changes the name of the group to “Colleges like mine”.

Focusing on a subset of attributes. Ella wonders what kinds of colleges are similar to Southeast colleges with low admission rate in tuition and admission rate (Q4). After specifying “Colleges like mine”, Ella drags tuition and admission rate from the attribute list to the attribute shelf between the two group shelves.

4.2 Recommending Similar and Different Groups

As users specify an object group (e.g., a group of colleges), Duet recommends a ranked list of similar object group and a ranked list of different object groups. An object group is ranked higher in the similar (different) list if it is more similar to (different from) the object group specified by users. These recommended groups are shown on a “card” in the result panel (Fig. 1e). The result panel keeps all the cards created during users’ analysis to help users keep track of their analytic provenance.

A card has a front side and a back side to show recommended groups in two different ways. Users can flip between the two sides by clicking ☰. The front side presents the recommended groups in a one-sentence summary that shows the top three similar groups and the top three different groups (Fig. 3 bottom) while the back side shows the complete lists of similar and different groups (Fig. 4). Depending on the size of a dataset, there can be hundreds of recommended groups in the ranked lists. We display fewer information on the front side to prevent users from being overwhelmed by the potentially large number of recommendations (C2).

To inspect the attributes that are similar to (different from) the specified group, users can select a similar (different) group from the back side of a card (Fig. 4). Duet shows a similar (different) attribute list in which a minuscule diverging bar chart is displayed alongside an attribute name. The minuscule bar chart serves to give a rough idea about how similar or different the attribute is between the specified group and the selected group. The bar charts above and below the horizontal axis show the distributions for the specified group and the selected group. An alternative design is a grouped bar chart. The bars in a grouped bar chart are narrower because all bars are squeezed into the same side, making this design less preferable for the minuscule bar chart.

4.3 Specifying Two Object Groups

Being a freshman in a Southeast college, Ella has a couple more questions about Southeast colleges: Q5) how does Southeast colleges and the colleges in New England (where the Ivy League schools reside) compare? Q6) Are Southeast colleges and New England colleges different in tuition? Q7) What are the unique characteristics of Southeast colleges? These questions involve specifying *two* object groups and require users to place an object group on each of the two group shelves. For instance, to answer Q5, Ella drags *New England* and *Southeast* from the small table to the top and bottom group shelves respectively.

Focusing on a subset of attributes. To consult Duet about the difference in tuition between Southeast and New England colleges (Q6),

Ella simply drags *Tuition* from the attribute list the attribute shelf after dragging *Southeast* and *New England* to the group shelves. Narrowing to a subset of attributes is useful when there are many attributes and users are interested only in a small number of them.

Comparing an object group with all other records. Understanding the unique characteristics of an object group entails comparing it with all other records in a data table. Being intrigued by the unique characteristics of Southeast colleges (Q7), Ella first drags *Southeast* from the table to a group shelf. The *Everything Else* tag appears at the bottom right of the small table (Fig. 1c bottom right) to allow users to compare the specified group with all other records in the data table. Ella drags *Everything Else* to the unoccupied shelf to investigate the uniqueness of Southeast colleges.

4.4 Recommending Similar and Different Attributes

As users specify two object groups (e.g., Southeast colleges and New England colleges), Duet recommends a ranked list of similar attributes and a ranked list of different attributes between the two groups. Similar to how it presents the recommended groups, Duet presents the recommended attributes in two different ways. The front side displays the recommended attributes in a one-sentence summary that shows only the top three similar attributes and the top three different attributes (Fig. 1e) to prevent users from being overwhelmed by a large number of recommendations (C2). The back side shows the complete lists of similar and different attributes (Fig. 1e).

4.5 Interacting with Duet’s Recommendations

Recommended groups and attributes are shown as underlined text on a card. By default, *green underlined text* encodes a similar group or attribute while *red underlined text* encodes a different group or attribute. Users can select a colorblind-safe palette from the settings menu (Fig. 1a). To help users interpret the recommendations, Duet shows an explanation when users hover over the underlined text. To enable users to revisit interesting recommendations, Duet saves a recommended group or attribute when users click on it.

Hover over a recommended attribute to see an explanation. As Ella specifies New England colleges and Southeast colleges to compare them (Q5), Duet recommends Median Family Income as a different attribute. As she hovers over *Median Family Income*, Duet provides its explanation for why family income is different in a small window (Fig. 3 top). The small window shows a grouped bar chart that visualizes the distribution of Median Family Income, and a textual description that depicts the difference in average. We use grouped bar chart as it aids comparison by showing the two distributions along the same x and y axis. Different from the minuscule bar chart, the window contains enough space to show wider bars. We use simple templates to generate the textual descriptions. As people with low visual literacy likely constitute a large part of the data analysis novice population, using the textual descriptions, we seek to help them interpret the grouped bar chart (C4).

Click to save a recommended attribute. Ella finds the difference in median family income between New England and Southeast colleges informative. She wants to save the recommended attribute and share it with a friend later. Ella clicks on *Median Family Income* to save it (C3). Duet stores a recommended attribute as a group - attribute - group (GAG) structure (e.g., New England colleges - Median Family Income - Southeast colleges) (Fig. 3 top). GAG structures are visualized as a relationship map (Sec. 4.6).

Hover over a recommended group to see an explanation. When Ella investigates “Colleges like mine” (Q3), Duet says that they are the most similar to fringe rural colleges. As Ella hovers over *Locale=Fringe Rural*, Duet explains that there are nine attributes in which the two groups of colleges are similar (Fig. 3 bottom). Users can flip to the back side of the card to see a complete list of similar (different) attributes between the specified group and any similar (different) groups. By flipping to the back side of the card and selecting *Locale=Fringe Rural* (Fig. 4), Ella finds that “Colleges like mine” are similar to fringe rural colleges in attributes such as Undergrad Population and %Hispanic.

Click to Save a Recommended Group. Ella wants to save all the nine similar attributes between “Colleges like mine” and fringe rural colleges. She clicks on [Locale=Fringe Rural](#) on the front side of the card (Fig. 3 top) (C3). Nine GAG structures, each corresponds to a similar attribute, is saved.

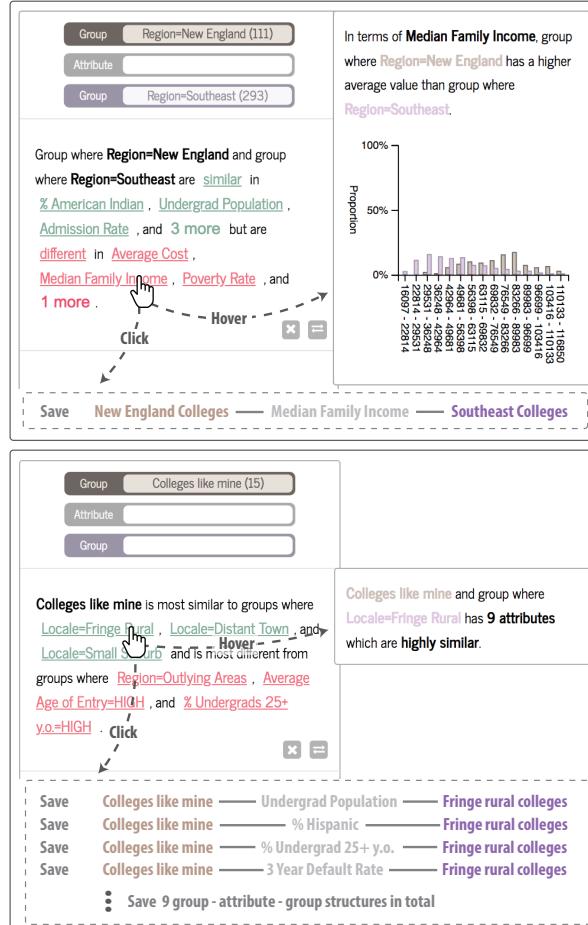


Fig. 3. Interacting with (top) recommended attributes and (bottom) recommended groups. Users can hover over a recommendation to see the explanation and click on it to save it.

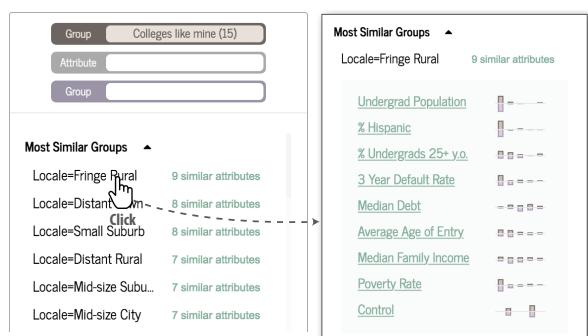


Fig. 4. Ella clicks on [Locale=Fringe Rural](#) to see the complete list of nine similar attributes.

4.6 Exploring the Relationship Map

The relationship map visualizes the recommendations saved to allow users to revisit them later. Figure 5 illustrates how a relationship map is constructed from the saved GAG structures. Unique groups are visualized as nodes and attribute(s) between two groups are visualized as a link between the two nodes. The thickness of a link between two nodes encodes the number of attributes that were saved between the two

groups. The nodes are initially organized in equal distance along the circumference of a circle. We use a circular layout because the nodes move in a more predictable way: as users save Duets recommendations, the newly added nodes and the existing nodes always move along the circumference. Alternatives include a force-directed layout in which a newly added node pushes the other nodes in various directions. The unpredictable movement of nodes might distract users from their analysis and is hard to keep track of. Duet supports two basic interactions with the relationship map:

Click on a link to display the saved recommendations. After exploring the college dataset for a while, Ella saves dozens of recommended groups and attributes, the relationship map created by her is shown in Figure 1f. She clicks on the link between [Locale=Fringe Rural](#) and [Admission Rate=HIGH](#) to see the saved similar and different attributes between the two groups in the stored recommendation panel (Fig. 1g).

Click on a node to show a radial layout. To further compare and contrast the saved groups, users can click on a node in the relationship map. For example, clicking on the node [Locale=Large City](#) helps Ella answer questions like how colleges in large cities are similar to or different from the other groups of colleges. Selecting a node changes the layout from a circular layout to a radial layout (Fig. 9). The selected node becomes the focal node and all links are hidden. The radial distance between the focal node and another node encodes the similarity between them: the closer the more similar. The periphery nodes are also color-coded using a diverging color scheme from red to green: deeper green means more similar and deeper red means more different. Users can select a colorblind-safe palette from the settings menu. While other layouts such as an MDS layout can be used to depict the distances between pairs of objects, we use a radial layout as it is a more faithful representation of the pairwise distances between objects [14].

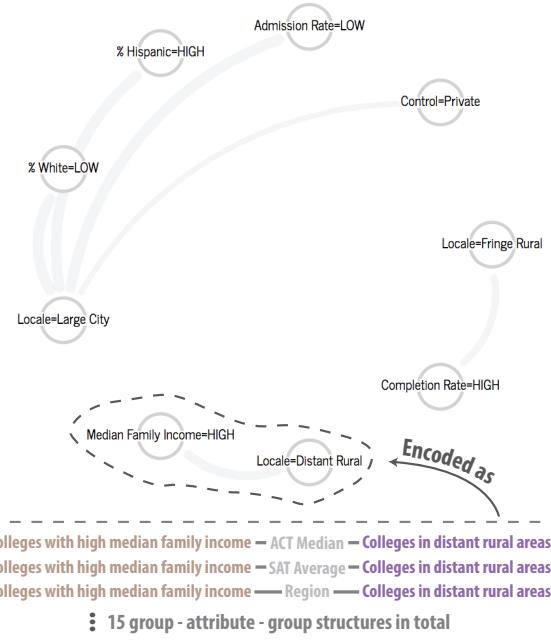


Fig. 5. Constructing the relationship map from the GAG structures.

5 REALIZING MINIMAL SPECIFICATION

In this section, we present how minimal specification can be realized computationally. We first describe, in general, how minimal specification can be achieved (Sec. 5.1 and 5.2). We then illustrate the specific computational method we use for minimal specification (Sec. 5.3).

5.1 Recommending Similar and Different Attributes

When users specify two object groups (e.g., Southeast and New England colleges), similar and different attributes are recommended. Each object group contains m distributions for a data table with m attributes. Two object groups, together, have m^2 distribution pairs (Fig. 6b). A

distribution pair are two sets of values (e.g., tuitions of Southeast colleges and tuitions of New England colleges) that correspond to an attribute (e.g., *Tuition*). If the attribute is a numerical (categorical) attribute, the distribution pair contains two sets of numbers (categories). To recommend similar and different attributes, Duet classifies each distribution pair D_i into similar (S), different (D) and somewhere in the middle (M) using a classification function f (Fig. 6c). The black box function f will be explained in detail in Sec. 5.3. Formally,

$$f : D_i \rightarrow \text{class} : \text{class} \in \{S, D, M\}$$

The similar (different) attributes correspond to the distribution pairs that are classified similar (different). Distribution pairs that are classified somewhere in the middle are not recommended to users (Fig. 6d). Each similar (different) attribute is associated with a value of similarity (dissimilarity). The list of similar (different) attributes is ranked based on the values of similarity (dissimilarity).

Rather than classifying the distribution pairs as similar, different and somewhere in the middle, an alternative approach is to compute the statistical distance for each of the m distribution pairs without classification. The m attributes can be ranked from the most similar to the most different based on the statistical distances. Such an approach would require users to browse through a list of attributes to determine which attributes are similar enough and different enough. When the list contains hundreds of attributes, browsing through it will be cognitively demanding. This issue might be worse for data analysis novices because they are not familiar with analyzing which attributes are similar or different. Hence, we choose to classify attributes into similar and different as users specify two object groups.

5.2 Recommending Similar and Different Groups

When users specify one object group, similar and different groups are recommended. Duet first generates a list of groups from attributes (Fig. 6f). For categorical attributes, one group is generated for each category. For example, *Locale* consists of 12 categories (e.g., *Large City* and *Remote Rural*). 12 groups (i.e. colleges where *Locale=Large City*, colleges where *Locale=Remote Rural* and so on) are created. For each numerical attribute, two groups (*LOW* and *HIGH*) are created. For example, Duet computes a threshold for *Tuition*. This threshold is defined in a way to ensure that the number of colleges with *Tuition* above and below the threshold are approximately the same. Two group (i.e. colleges with *Tuition* below the threshold and colleges with *Tuition* above the threshold) are then created. For each generated group, Duet computes the number of similar attributes and the number of different attributes (Fig. 6g). A ranked list of similar (different) groups is obtained by sorting the list of generated groups in descending order of the number of similar (different) attributes (Fig. 6h).

The current version of Duet only generates groups that satisfy a single attribute-value pair (e.g. colleges where *Locale=Large City*). Generating a list of groups that satisfy more than one attribute-value pair (e.g. colleges where *Locale=Large City* and *Tuition=LOW*) and evaluating the number of similar and different attributes for each group can be computationally expensive due to the exponentially large number of groups that can be generated. We would like to explore this in future.

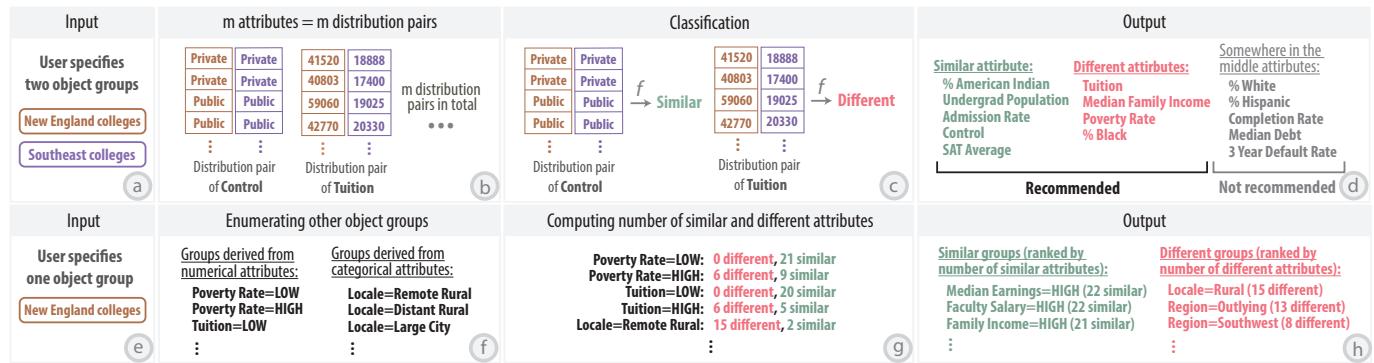


Fig. 6. (Top row) When users specify two object groups, Duet recommends a list of similar attributes and a list of different attributes. (Bottom row) When users specify one object group, Duet recommends a list of similar groups and a list of different groups.

5.3 Promoting Trust in Recommendations

Promoting trust is a crucial consideration in designing recommendation systems. Researchers have been investigating ways to inspire users' trust in recommendations (e.g., [21, 28, 43]). Users may lose faith in a recommendation system like Duet if they observe many false positive results (e.g., the system claims that an attribute is similar between two object groups but users disagree upon verification). To promote trust, Duet's classifications should be consistent with human judgement. To operationalize consistency, we define it on a population level. For example, if the majority of 100 people think that an attribute is similar, classifying the attribute as similar is considered consistent with human judgement. In this section, we discuss the feasibility of optimizing the consistency of the classification function f mentioned in Sec 5.1.

5.3.1 Measuring Similarity

The classification function f takes as input a distribution pair (two sets of values), which corresponds to a numerical attribute or a categorical attribute. The classification function first converts each distribution pair into a probability distribution pair. Before the conversion, numerical attributes are converted to categorical attributes by discretizing them into bins. For a single set of values, applications such as Excel derive from the number of values n the number of bins $k = \sqrt{n}$ to reduce information loss. As there are two sets of values in a distribution pair, we compute the number of bins for each numerical attribute as $k = \sqrt{(n_1 + n_2)/2}$, where n_1 and n_2 are respectively the number of values in the first set and the number of values in the second set.

The classification function f then computes the similarities between the two probability distributions. We use Bhattacharyya (Bh) coefficient, which is a widely used measure of the degree of overlapping between two probability distributions [16, 17, 40]. A higher value means more overlapping between two distributions (more similar) and a lower value means less overlapping (more different). We note, however, that other statistical distances, such as Kullback-Leibler divergence and Kolmogorov-Smirnov statistics, can be used and should be tested as future work.

5.3.2 Multinomial Logistic Regression Model for Classification

Having computed the Bh coefficient for a distribution pair, the next question concerns how to determine whether the value is high enough to be considered similar or low enough to be considered different. Several approaches are commonly used, including the top-k approach (e.g., [47]), the p-value approach (e.g. [45]), and the arbitrary threshold approach. Using these approaches, arbitrariness are involved in deciding whether the Bh coefficient is high or low enough.

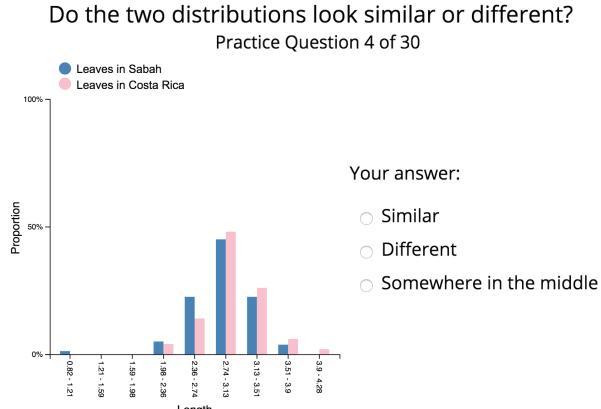
Consider the arbitrary threshold approach that set arbitrary upper and lower thresholds. If Bh coefficient is above the upper threshold, a distribution pair is considered similar. If the upper threshold is set too low, some distribution pairs that are not very similar will be classified as similar; if it is set too high, some distribution pairs that should have been classified as similar will not be not classified as similar.

Promoting trust would require optimizing the upper and lower thresholds in a way that is consistent with human judgement to reduce the

likelihood of unreasonable classifications. As an example of such an optimization, if the majority of 100 people consider a distribution pair as similar, the upper threshold should probably be set somewhere below the Bh coefficient for this distribution pair so that it is classified as similar. To avoid setting arbitrary thresholds, we use multinomial logistic regression to model the relationship between perceived similarity of a distribution pair and Bh coefficient. While a full description of how the logistic regression model was built is beyond the scope of this paper, we briefly outline the procedure involved as follows:

Data collection. To increase the diversity of the distribution pairs, we collected 520 distribution pairs from 83 real-world datasets, a majority of which are datasets from the R statistical software [7]. During data collection, one of the authors labelled the distribution pairs as “similar”, “somewhere in the middle” and “different” by visualizing the distribution pairs as grouped bar charts. This kept us aware of whether the three labels are balanced in the collected data. This yielded 160 similar distribution pairs, 202 different distribution pairs and 158 somewhere in the middle distribution pairs. The labels represent perceived similarity of the distribution pairs.

Ensuring data quality. To ensure the quality of the labels, we selected 150 marginal cases (e.g., distribution pairs that are preliminarily labelled as similar but do not look highly similar) for relabelling. 10 graduate students with background in data analysis were recruited to relabel these 150 marginal cases. The subjects were required to go through a tutorial session during which they labelled 30 distribution pairs. After the tutorial session, they labelled 75 distribution pairs in the actual session. The interfaces used in the tutorial session (Fig. 7) and the actual session were the same but the stimuli used in the two sessions did not overlap. The order of the stimuli was also randomized in both sessions. From the subject, we collected 5 labels for each of the 150 marginal cases. We used a majority vote to determine the final labels. When there was a tie (e.g., 2 subjects labelled a distribution pair as similar and 2 labelled it as somewhere in the middle), we labelled the distribution pair as somewhere in the middle.



6.2 Procedure

The study was conducted in a quiet lab environment using a Macbook Pro with 13-inch display of resolution 2560 x 1600 and a mouse. During the two analysis sessions, the participants' interactions with Duet were screen-captured and their verbalizations were audio-recorded. Each study session lasted around 90 minutes. The procedure of the study is described as follows:

Training session (~15 minutes). The participants watched a 5-minute tutorial video that introduced various aspects of Duet. After watching the tutorial video, the participants were instructed to finish a series of practice tasks in 10 minutes. The investigator helped overcome difficulties the participants encountered. We used a car dataset for both the tutorial video and the practice tasks.

First data analysis session (~20 minutes). The participants were asked to analyze the first dataset (four participants started with the college dataset and another four started with the city dataset). Before the analysis session, the participants received an attribute list with description of each attribute. They had 2-3 minutes to review the attribute list. The participants were then provided a task description with a short scenario. They were told to keep talking aloud, and report at least five findings by comparing different groups. They were also informed that they can stop the analysis session early if they think nothing more can be found from the dataset.

Second data analysis session (~20 minutes). The participants were asked to analyze another dataset by following a similar procedure as above.

Interview (~15 minutes). Finally, we interviewed the participants concerning their experience using the system. The participants completed a questionnaire at the end of the interview.

6.3 Results

Figure 8 summarizes the participants' ratings in the questionnaire. Here, we discuss a few key points from our observations of the participants during their analysis, the end-of-study interviews and their responses to the questionnaire.

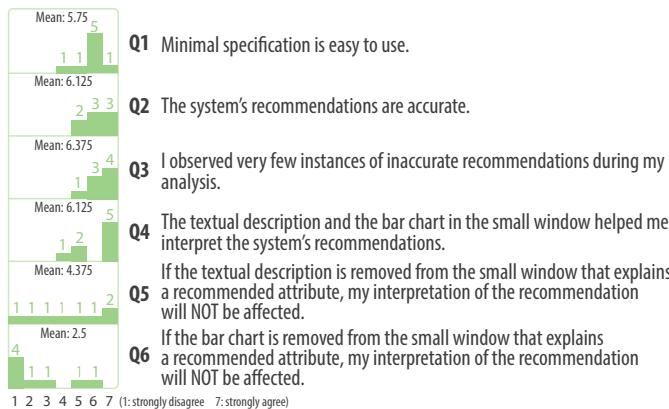


Fig. 8. Participants' ratings in the end-of-study questionnaire.

6.3.1 Minimal Specification

We received positive feedback about the ease of use of minimal specification in spite of some usability issues. Several participants appreciated the ease of use of minimal specification when they have a pairwise comparison question in mind: “*When I know what question I am trying to answer, it is really easy to just plug and play*” (P2) and “*I think for pairwise comparison, minimal specification is good because if you have the specific things to compare, you just add the groups and attributes you want to compare*” (P7). P5 noted how minimal specification reduced the amount of effort required for pairwise comparison: “*It instantly provides you with a lot of matches [recommendations] which you might otherwise have to do a lot of work to see. It would make analyzing so many things really easy.*”

Participants also encountered some usability issues while they were using minimal specification during analysis. P5 commented that having to click on an attribute in the attribute list to pull it off to the data table and dragging a tag from the data table to a group shelf took extra steps. P4 and P6 raised concerns on the learning curve of minimal specification: “*Because it is minimal specification, there are a lot of recommendations that come from the comparison. Sifting through all of them was a learning curve*” (P6), and “*Creating more complex groups wasn't immediately intuitive to me*” (P3).

6.3.2 Interpreting Duet's Recommendations

Besides rating whether Duet's explanations helped interpret its recommendations (Q4 in Fig. 8), during the interview, the participants were asked to comment on the roles of grouped bar chart, and textual descriptions in interpreting Duet's recommendations. They found the text and the bar chart helped interpretation in various ways.

Roles of grouped bar charts in interpreting recommendations. During the end-of-study interview, the participants were asked to comment on how removing the grouped bar chart from the small window (like the one in Fig. 3 top) might affect their interpretation of a recommended attribute. The participants commented on three different roles of the grouped bar chart in interpreting a recommended attribute:

Comprehension. P1 used a metaphor to explain how comprehension will be hampered if the grouped bar chart was removed: “*I have seen the trailer of Black Panther. Seeing that trailer, I would know the characters but I wouldn't know anything of the movie until I watch it. The trailer may paint a completely different picture.*” He thought that after removing the grouped bar chart, users only know whether an attribute is similar or different, which was not enough. The mere fact that an attribute is similar or different is like the movie trailer that paints a small picture. Like the movie itself, the bar chart offers crucial information about how similar or different the attribute is.

Trust. P2 said that she would not trust the system's recommendations if grouped bar chart was removed from the small window that explains a recommended attribute: “*If I can fully trust your system, sure, I don't need to see the bar chart. The bar chart is the only thing giving you the concrete data that is supporting the recommendations. I think it might come down to the issue of trust if you get rid of the actual data backing the claim for the recommendations.*”

Freedom. P7 used how Facebook manipulated our feed to explain how the grouped bar chart provides users freedom to discover the information they want to discover: “*It is more like Facebook feed. It's Facebook deciding what you see vs. you decide what you want to see on your feed. When you present a visualization, it's up to the user what insights they get from it vs. in this case [only present text without the grouped bar chart], I can only know what you tell me and there is no way I can explore more. I would like to give the control to users and let them decide what they want to discover.*”

Roles of textual descriptions in interpreting recommendations. During the end-of-study interview, the participants were also asked about how their interpretation of a recommended attribute might be affected if the textual description is removed from the small window that explains the recommended attribute. Some participants pointed out that the textual description is not very informative: “*I don't think it is necessary. I am not seeing anything new from the textual description*” (P8) and “*I never read any of the textual descriptions. I just look at the picture*” (P4). However, other participants commented that the text helps interpret Duet's recommendations in two different ways:

Reminder. Several participants commented that the textual descriptions serve as reminders that got them back on track when they felt lost during their analysis. P1 explained, “*When I am trying to analyze something, I might forget in between. These [textual descriptions] are sort of like indices. When I am looking at these values, then I remember what I am doing.*” P3 provided reasons for why he liked the textual description, “*If there is a moment where I kind of get a little bit confused, If I just read that little text description, I immediately like 'oh yeah, that right'. It is useful for when I get a bit flustered or for a second forgot what I*

was doing, just take a second to reread the text reset my focus" (P3). *Complementary to grouped bar chart.* P2 gave a 2 (strongly disagree) for Q5 and a 6 (strongly agree) for Q6. She commented on how low visual literacy might contribute to her preference for text: "For someone like me, I absolutely need the text. For some reasons, I have a hard time following the graph. I guess I don't look at graphs on a daily basis. Having the text really helped me try to make sense of the graph. I think I would have been a lot worse off if I do not have the text."

6.3.3 Usage of the Relationship Map

Some participants were enthusiastic about the relationship map. P4 depicted how the relationship map can be used in a marketing campaign: "Say, I am Coca Cola. I want to create a commercial and I want to appeal to the biggest group which is like the baby boomer population. With this (the relationship map), you might see there is similarity between millennial population and the elderly people. Then their combine may be greater than their original idea doing the baby boomer population." Furthermore, we also observed interesting usage of the relationship map. While P8 was investigating what might contribute to the prevalence of violent crime, he created a graph with 10 nodes (Fig. 9). During his analysis of the college dataset, P6 created a mind map to organize his thoughts (Fig. 1f) and illustrated how he liked the feature: "This is a super cool feature! I am seeing these things [recommendations] and they are really interesting to me. I don't want to lose those thoughts. One of the things that I would like to do during my analysis is I would like to connect it with thoughts."

At the same time, several participants commented on why they did not use the relationship map a lot: "the window on the right consumes a lot of space but I don't think that it is really necessary" (P1) and "creating groups and then also nodes on the far-right screen ... that part was not as straight forward" (P3).

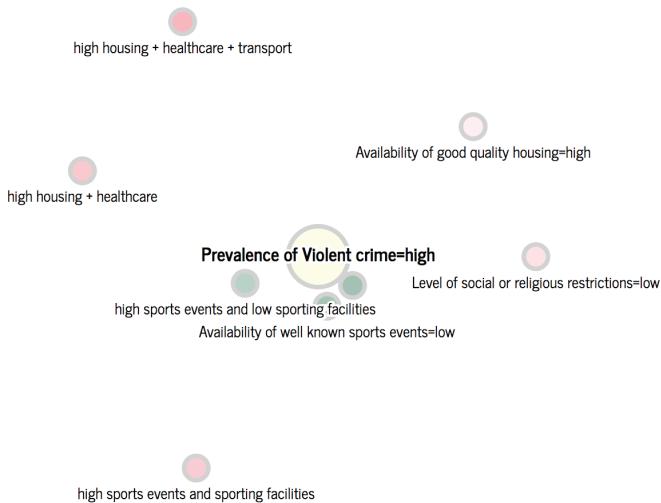


Fig. 9. The relationship map created by P8 while he was investigating the factors that contribute to prevalence of violent crimes. The node labelled Prevalence of Violent crime=high is selected as the focal node.

7 DISCUSSION

Our study provides preliminary evidence that data analysis novices find minimal specification for pairwise comparison easy to use. Unsurprisingly, some participants compare minimal specification with other tools in terms of pairwise comparison: "With Excel, you have to use the pivot table to be able to compare within a column so this [minimal specification] is quicker" (P4), and "This [minimal specification] is even simpler than Tableau for comparing groups and attributes" (P6). A further study is required to compare minimal specification and other techniques in term of their effectiveness for pairwise comparison.

While many participants feel that Duet's recommendations are accurate (the average ratings for Q2 and Q3 in the questionnaire are

6.125 and 6.375 respectively), subjective measures of accuracy using questionnaires can easily be contaminated by participants' biases due to observer-expectancy effect. Furthermore, the "accuracy" of Duet's recommendations need to be further investigated. Two challenges are involved in operationalizing the notion of accuracy of recommendations: how to define accuracy and how to measure accuracy. First, defining accuracy of a recommendation is difficult because there may not exist a ground truth. For example, determining whether an attribute is similar between two groups can be subjective, especially for marginal cases. If some people consider an attribute similar and some people consider it different, whose judgement should be used as the ground truth? An alternative solution is to consider accuracy on a population-level: if 80 out of 100 people think an attribute is similar, then similar should be used as the ground truth. Second, to measure the accuracy of recommendations, one needs to compare the system's result with people's judgement. For example, if both the system and people say that an attribute is similar, the system's recommendation is probably accurate. However, people's judgement can be biased. We got high rating for both Q2 and Q3 probably because Duet's recommendations biased the participants (e.g., they thought an attribute was similar because Duet told them it is but their judgement would be different if Duet's classification is not presented). Investigating a less-biased way of collecting people's judgement will be an interesting future direction.

We are also aware of several limitations of Duet. Our template-based approach to generating natural language description is very limiting. While some participants liked the text, other participants commented that the textual description for explaining a recommended attribute "is not very informative", "is redundant" and "can be removed". Future work is necessary to understand how natural language can be more informative in describing a visualization. Duet also assumes that determining whether two groups are similar or different is domain-independent, which is not true. In some domains, people might have a more stringent requirement on whether an attribute is similar between two groups. As future work, we would like to investigate how users can incorporate their domain knowledge into Duet's recommendations. Yet another limitation concerns scalability. While we have not experimentally tested the scalability of Duet, interactive latency can be high for data tables with a couple thousands rows. Duet currently does not have a backend. We envision that moving expensive operations to a backend will improve Duet's scalability.

There is a recent surge in work that help users glean insights by automatic insight generation in both the academia [45, 47] and the industry [6, 8, 19, 20]. As visualizations are a medium for gaining insights from data, the role of visualizations in data exploration overlaps with these automatic techniques. As visualization researchers, we clearly need to rethink how visualizations fit into the future of data exploration where more automation will inevitably occur. We believe that our user studies shed light on the symbiosis between visualizations and insights automatically generated by algorithms. As we observed in our user study, visualizations can add to automatic insights by aiding comprehension, promoting trust and giving freedom for users to explore the full picture. On the other hand, automatic insights, when they are presented in other forms such as text, grant people who have low visual literacy (like P2) access to the colorful world of visualizations.

8 CONCLUSION

In this paper, we presented Duet, a system that employs minimal specification to help data analysis novices conduct pairwise comparisons. Duet uses the minimal specification technique to reduce execution barrier during pairwise comparison. It also explains recommendations using both grouped bar charts and textual descriptions to reduce the barrier in interpreting why a recommendation is offered. Results of our qualitative study suggest that the participants found minimal specification easy to use and the explanations helped interpret Duet's recommendations. Our research provides insights into the roles of visualizations in the world of *automatic insights* and how both of them, together, can help novices conduct data analysis. Duet is available as an open-source software: <https://duetpaircomp.github.io/>

REFERENCES

- [1] Car comparison — usnews.com. <https://cars.usnews.com/cars-trucks/compare>. [Accessed: 31st March 2018].
- [2] College compare — usnews.com. <https://www.usnews.com/best-colleges/compare>. [Accessed: 31st March 2018].
- [3] College scorecard data. <https://collegescorecard.ed.gov/data/>. [Accessed: 31st March 2018].
- [4] Compare cities. https://www.numbeo.com/crime/compare_cities.jsp. [Accessed: 31st March 2018].
- [5] Explainable artificial intelligence. <https://www.darpa.mil/program/explainable-artificial-intelligence>. [Accessed: 31st March 2018].
- [6] Power bi — interactive data visualization bi tools. <https://powerbi.microsoft.com/en-us/>. [Accessed: 31th March 2018].
- [7] R data sets. <https://vincentarelbundock.github.io/Rdatasets/datasets.html>. [Accessed: 31th March 2018].
- [8] Salesforce einstein is artificial intelligence in business technology - salesforce.com. <https://www.salesforce.com/products/einstein/overview/>. [Accessed: 31th March 2018].
- [9] Workshop on explainable smart systems. <http://explainablesystems.comp.nus.edu.sg>. [Accessed: 31st March 2018].
- [10] R. A. Amar and J. T. Stasko. Knowledge precepts for design and evaluation of information visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):432–442, 2005.
- [11] N. Cao, D. Gotz, J. Sun, and H. Qu. Dicon: Interactive visual analysis of multidimensional clusters. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2581–2590, 2011.
- [12] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999.
- [13] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, H.-J. Schulz, M. Streit, and C. Tominski. Characterizing guidance in visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):111–120, 2017.
- [14] J. Chuang, D. Ramage, C. Manning, and J. Heer. Interpretation and trust: Designing model-driven visualizations for text analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 443–452. ACM, 2012.
- [15] B. chul Kwon, B. Fisher, and J. S. Yi. Visual analytic roadblocks for novice investigators. In *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 3–11. IEEE, 2011.
- [16] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 142–149. IEEE, 2000.
- [17] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.
- [18] Z. Cui, S. K. Badam, A. Yalçin, and N. Elmquist. Datasite: Proactive visual data exploration with computation of insight-based recommendations. *arXiv preprint arXiv:1802.08621*, 2018.
- [19] Ç. Demiralp, P. J. Haas, S. Parthasarathy, and T. Pedapati. Foresight: Recommending visual insights. *Proceedings of the VLDB Endowment*, 10(12):1937–1940, 2017.
- [20] K. Dhamdhere, K. S. McCurley, R. Nahmias, M. Sundararajan, and Q. Yan. Analyza: Exploring data with conversation. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pp. 493–504. ACM, 2017.
- [21] F. Du, C. Plaisant, N. Spring, and B. Shneiderman. Finding similar people to guide life choices: Challenge, design, and evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 5498–5544. ACM, 2017.
- [22] C. Forsell and J. Johansson. An heuristic set for evaluation in information visualization. In *Proceedings of the International Conference on Advanced Visual Interfaces*, pp. 199–206. ACM, 2010.
- [23] M. Galesic and R. Garcia-Retamero. Graph literacy: A cross-cultural comparison. *Medical Decision Making*, 31(3):444–457, 2011.
- [24] T. Gao, M. Dontcheva, E. Adar, Z. Liu, and K. G. Karahalios. Datatone: Managing ambiguity in natural language interfaces for data visualization. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pp. 489–500. ACM, 2015.
- [25] M. Gleicher. Considerations for visualizing comparison. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):413–423, 2018.
- [26] D. Gotz and Z. Wen. Behavior-driven visualization recommendation. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, pp. 315–324. ACM, 2009.
- [27] L. Grammel, M. Tory, and M.-A. Storey. How information visualization novices construct visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):943–952, 2010.
- [28] J. L. Herlocker, J. A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, pp. 241–250. ACM, 2000.
- [29] E. Hoque, V. Setlur, M. Tory, and I. Dykeman. Applying pragmatics principles for interaction with visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):309–318, 2018.
- [30] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer. Wrangler: Interactive visual specification of data transformation scripts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3363–3372. ACM, 2011.
- [31] A. Key, B. Howe, D. Perry, and C. Aragon. Vizdeck: Self-organizing dashboards for visual analytics. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pp. 681–684. ACM, 2012.
- [32] A. J. Ko and B. A. Myers. Designing the whyline: a debugging interface for asking questions about program behavior. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 151–158. ACM, 2004.
- [33] A. J. Ko and B. A. Myers. Finding causes of program output with the java whyline. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1569–1578. ACM, 2009.
- [34] T. Kulesza, M. Burnett, W.-K. Wong, and S. Stumpf. Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, pp. 126–137. ACM, 2015.
- [35] H. Lam. A framework of interaction costs in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6), 2008.
- [36] S. Lee, S.-H. Kim, Y.-H. Hung, H. Lam, Y.-a. Kang, and J. S. Yi. How do people make sense of unfamiliar visualizations?: A grounded model of novice's information visualization sensemaking. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):499–508, 2016.
- [37] B. Y. Lim, A. K. Dey, and D. Avrahami. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2119–2128. ACM, 2009.
- [38] M. Mayer, G. Soares, M. Grechkin, V. Le, M. Marron, O. Polozov, R. Singh, B. Zorn, and S. Gulwani. User interaction models for disambiguation in programming by example. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pp. 291–301. ACM, 2015.
- [39] D. Norman. *The Design of Everyday Things: Revised and Expanded Edition*. Basic Books (AZ), 2013.
- [40] K. Okuma, A. Taleghani, N. De Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *European Conference on Computer Vision*, pp. 28–39. Springer, 2004.
- [41] G. Ridgeway. *Analysis of racial disparities in the New York Police Department's stop, question, and frisk practices*. Rand Corporation, 2007.
- [42] V. Setlur, S. E. Battersby, M. Tory, R. Gossweiler, and A. X. Chang. Eviza: A natural language interface for visual analysis. In *Proceedings of the 29th Annual Symposium on User Interface Software & Technology*, pp. 365–377. ACM, 2016.
- [43] R. Sinha and K. Swearingen. The role of transparency in recommender systems. In *CHI'02 Extended Abstracts on Human Factors in Computing Systems*, pp. 830–831. ACM, 2002.
- [44] A. Srinivasan and J. Stasko. Orko: Facilitating multimodal interaction for visual exploration and analysis of networks. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):511–521, 2018.
- [45] B. Tang, S. Han, M. L. Yiu, R. Ding, and D. Zhang. Extracting top-k insights from multi-dimensional data. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pp. 1509–1524. ACM, 2017.
- [46] J. W. Tukey. Comparing individual means in the analysis of variance. *Biometrics*, pp. 99–114, 1949.
- [47] M. Vartak, S. Rahman, S. Madden, A. Parameswaran, and N. Polyzotis. Seedb: Efficient data-driven visualization recommendations to support visual analytics. *Proceedings of the VLDB Endowment*, 8(13):2182–2193, 2015.

- [48] G. Wills and L. Wilkinson. Autovis: Automatic visualization. *Information Visualization*, 9(1):47–69, 2010.
- [49] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):649–658, 2016.
- [50] K. Wongsuphasawat, Z. Qu, D. Moritz, R. Chang, F. Ouk, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager 2: Augmenting visual analysis with partial view specifications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2648–2659. ACM, 2017.
- [51] K. Yessenov, S. Tulsiani, A. Menon, R. C. Miller, S. Gulwani, B. Lampson, and A. Kalai. A colorful approach to text processing by example. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software & Technology*, pp. 495–504. ACM, 2013.