The Effect of Edge Bundling and Seriation on Sensemaking of Biclusters in Bipartite Graphs

Maoyuan Sun, Jian Zhao, Hao Wu, Kurt Luther, Chris North and Naren Ramakrishnan

Abstract—Exploring coordinated relationships (e.g., shared relationships between two sets of entities) is an important analytics task in a variety of real-world applications, such as discovering similarly behaved genes in bioinformatics, detecting malware collusions in cyber security, and identifying products bundles in marketing analysis. Coordinated relationships can be formalized as biclusters. In order to support visual exploration of biclusters, bipartite graphs based visualizations have been proposed, and edge bundling is used to show biclusters. However, it suffers from edge crossings due to possible overlaps of biclusters, and lacks in-depth understanding of its impact on user exploring biclusters in bipartite graphs. To address these, we propose a novel bicluster-based seriation technique that can reduce edge crossings in bipartite graphs drawing and conducted a user experiment to study the effect of edge bundling and this proposed technique on visualizing biclusters in bipartite graphs. We found that they both had impact on reducing entity visits for users exploring biclusters, and edge bundles helped them find more justified answers. Moreover, we identified four key trade-offs that inform the design of future bicluster visualizations. The study results suggest that edge bundling is critical for exploring biclusters in bipartite graphs, which helps to reduce low-level perceptual problems and support high-level inferences.

Index Terms—Bicluster, edge bundling, seriation, visual analytics.

1 Introduction

Coordinated relationship exploration is an important task in various domains (e.g., investigating coordinated threats in intelligence analysis [1], detecting malware collusion in cyber security [2], and discovering similarly behaved genes in bioinformatics [3]). It is hard to manually find coordinated relationships, since analysts need to aggregate multiple entities by considering shared connections. This requires a significant amount of cognitive effort for checking individual relationships between pairs of entities.

Computational methods have been applied to help this. Specifically, biclusters, algorithmically identified groups of relationships, have been applied in visual analytics tools to support coordinated relationship explorations [4], [5], [6]. A bicluster is a grouped relationships between two sets of entities (e.g., persons and locations), where each entity in one set is related to all entities in the other. A bicluster reveals a specific coordinated relationship (e.g., four people visited the same three cities).

Biclustering algorithms find biclusters based on co-occurrence (e.g., CHARM [7] and LCM [8]), rather than semantic meanings. Referring to semantic meanings requires *domain knowledge* that computation lacks. This calls for visualizations that enable human to use domain knowledge for analysis (e.g., displaying biclusters in context of entities to direct user attention to meanings of entity labels). This is an important goal of visual analytics [9]. Moreover, computed biclusters are in a machine readable format (e.g., collections of entity IDs) and may overlap each other by sharing entities, so it is not easy for analysts to understand them and identify useful ones. For making biclusters usable, visualizations are necessary.

The fundamental challenge of visualizing biclusters are Euler diagram problems [10]. Because of overlaps, for clearly displaying biclusters and their entities, we have to either duplicate entities for making members of biclusters spatially near each other, or break biclusters by spatially separating their members to keep entities unique. This is the key design trade-off of bicluster visualizations: *relationship-centric* v.s. *entity-centric* [6]. In order to balance this, a bipartite graph based technique, BiSet [6], has been proposed. It groups edges into bundles, in the graph, to present biclusters and spatially seprates them, without duplicating entities. Memberships of entities are revealed by edges linking a bicluster and an entity.

While biclusters and entities are visually separated with different encodings, this edge bundling based technique may still suffer from edge crossings, when biclusters highly overlap. Moreover, we lack in-depth understanding of the impact of edge bundles on user exploring biclusters in bipartite graphs. For example, can edge bundles help users find complex domain specific coordinated relationships based on computed biclusters, by using their domain knowledge (e.g., considering the meanings of entity labels)? How much performance gain (e.g., accuracy) do bundles bring? How is the number of entity visits affected in user explorations? Are there any trade-off comparing using edge bundling to without them?

To address the edge crossing problem and answer above questions, we propose a novel bicluster-based seriation technique and conducted a user experiment to study the effect of edge bundling and the proposed ordering technique for bicluster explorations in a bipartite graph. Our key contributions in this paper are as follows:

1) We propose a novel bicluster-based seriation technique that helps to reduce edge crossings in bipartite graphs drawing, where biclusters are shown as edge bundles, and entities are displayed as nodes in the graph.

2) We present a detailed study design of a user experiment, as the first attempt to evaluate *edge bundling* and our proposed

- *Maoyuan Sun is with Department of Computer and Information Science, University of Massachusetts Dartmouth.*
  E-mail: smaoyuan@umassd.edu.
- *Jian Zhao is with FX Palo Alto Laboratory. E-mail: zhao@fxpal.com.*
- *Hao Wu is currently with Google. E-mail: wuhao723@vt.edu.*
- *Kurt Luther, Chris North and Naren Ramakrishnan are all with Computer Science Department, Virginia Tech.*
  E-mail: {kluther | north | naren}@cs.vt.edu.

Manuscript received ...

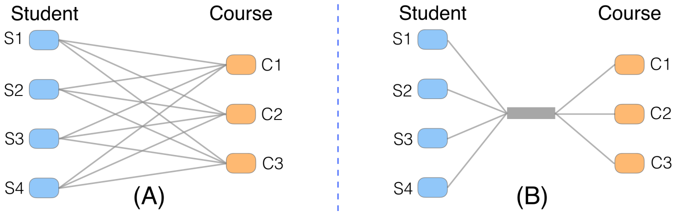

Fig. 1. An example of a bicluster that shows a coordinated relationship between four students and three courses. (A) presents detailed relationships between every student and each course. (B) illustrates the result of grouping individual relationships with an edge bundle.

seriation technique on visualizing biclusters in bipartite graphs.

3) We identify four trade-offs when using the two techniques to support biclusters exploration in bipartite graphs. They lead to useful design implications for tools that visualize relationships.

4) We find that edge bundling is critical for exploring biclusters in bipartite graphs. It helps to free users from low-level perceptual problems and support them making high-level inferences.

2 Background

2.1 Bicluster

Biclusters are outcomes from biclustering algorithms. They reveal coordinated relationships between two sets of entities. Algorithms for bicluster discovery (e.g., CHARM [7] and LCM [8]) typically try to find *closed* biclusters [11]. A closed bicluster is a complete bipartite graph, where every entity in one set is connected with each entity in the other, from a graph perspective. The cardinality of the two sets describes the size of a bicluster. Figure 1 shows an example of a closed bicluster, indicating a coordinated relationship that all four students take three courses, and its size is 4×3.

2.2 BiSet

BiSet [6] is a recently proposed technique to visualize biclusters in bipartite graphs based layout. It shows biclusters as edge bundles in-between two entity-lists. This *edge-bundles-as-biclusters* concept in BiSet is shown in Figure 1 (B). BiSet aggregates individual edges into bundles based on computed biclusters, so each bundle represents a bicluster. Moreover, on top of an edge bundle, BiSet uses two rectangles, with rounded corners, to indicate the number of involved entities. Thus, in BiSet, entities and biclusters locate in different lists, with different visual encodings.

2.3 Seriation

Seriation is an exploratory data analysis technique [12]. It permutes the order of objects to get a sequence where the regularity and pattern (e.g., clustering structure [13]) among the whole series can be well revealed. Seriation is commonly used to show patterns in a matrix by permuting rows and columns (e.g., Bertifier [14], BiVoc [15] and Termite [16]). Seriation in a matrix with M rows and N columns attempts to find orders of rows and columns that optimize certain objective function. Finding all possible combinations of ordering rows and columns in a matrix is $(M!N!)/2$, which is computationally expensive. Thus, seriation in a matrix is performed heuristically.

Different seriation methods use different objective functions to pursue heuristic solutions. For instance, Robinson [17] heuristically place the highest value along the diagonal in a matrix for seriation. The optimal leaf ordering method [18] begins with a hierarchical clustering of rows (or columns) and finds an order, which tries to minimize the sum of distances between consecutive items in the dendrogram. Statistical analysis methods have also been used for matrix seriation. For example, principal component analysis (PCA) [19] and correspondence analysis (CA) [20] treat a matrix as a high-dimensional data (rows as observations and columns as variables) and attempt to find two orthogonal axes as a 2D space to project data. In this 2D space, the total variance of the data can be maximized, and the order on the two orthogonal axes is the seriation result. In this work, we use CA to perform seriation in lists for edge crossing reduction. Section 3.2.1 discusses the connections between CA and edge crossings in lists.

2.4 Related Evaluation

Matrix is a well studied layout to show biclusters, especially in the bioinformatics domain for gene behavior analysis (e.g., [15], [21], [22], [23], [24]). Each bicluster is displayed as a matrix, which has been found helpful to support text analytics [4], [25] by directing user attention to related documents. These are exploratory studies focusing on how visualized biclusters are used to help connect information from documents, rather than coordinated relationship exploration. The size of biclusters used in these studies was at least 3×3, which helps us select the bicluster size.

Despite showing coordinated relationships, edge bundling [26] has been used in graphs for visual clutter reduction based on certain rules (e.g., force-directed model [27], spatial proximity [28], network connectivity [29], and hierarchical structure in data [30]). By reducing the number of edges displayed, edge bundling helps to improve the graph readability [29], [31]. The bundling concept has also been used to help track animated objects [32]. Seriation has been explored in matrix-based layouts to show patterns, and a comprehensive survey of seriation can be found in [33]. However, it still lacks in-depth understanding of the two techniques for visualizing biclusters to support coordinated relationship explorations.

3 Seriation in Bipartite Graphs

3.1 Design Requirement Analysis

In a bipartite graph based layout, like BiSet, the position of entities and their associated biclusters can impact edge crossings, because bicluster overlaps are revealed as edges from the same entities connecting with different biclusters. To reduce edge crossings, in the simplest case, we need to simultaneously organize elements in three lists: a bicluster-list and two neighboring entity-lists. This is challenging for two reasons. It requires sorting in a 3D space, if we consider each list as an individual dimension. Moreover, positions of elements in each dimension are constrained by positions of elements in other dimensions, if our goal is to put biclusters and their associated entities near each other. BiSet visually displays the three lists in a linear manner: a bicluster-list in-between two entity-lists. Thus, this problem can be viewed as sorting two pairs of neighboring lists: a bicluster-list with its left neighboring entity-list, and a bicluster-list with its right neighboring entity-list.

For each pair of lists, minimizing edge crossings by ordering two lists is NP-hard [34], which needs heuristic solutions. Seriation offers a possible solution [35]. As discussed before, seriation has been used to reveal patterns in a matrix. With some heuristic strategies to permute rows and columns, seriation attempts to order elements in rows and those in columns in two sequences and the combination of the two can reveal some patterns in the matrix [12].

However, the above problem cannot be solved by simply applying seriation to two pairs of lists respectively, since the bicluster-list may have two different orders. One is from the seriation between the bicluster-list and its left neighboring entity-list, and the other comes from the seriation between the bicluster-list and its right neighboring entity-list. Because of two different orders, how to organize biclusters in this bicluster-list becomes a problem.

3.2 Bicluster-based Seriation Technique

We merge biadjacency matrices to enable seriation in a list-based layout of bipartite graphs. These biadjacency matrices indicate relationships between two partitions: entities and biclusters, instead of an original bipartite graph between two sets of entities. This approach is inspired by the design of data context map [36]. It can display both data items and attributes in a 2D space based by fusing four distance matrices, which include pairwise distance of data items, attributes, attributes to data items, and data items to attributes. Using a merged biadjacency matrix, the double orders problem can be addressed, so it is possible to apply seriation to multiple lists. Specifically, in this work, we use correspondence analysis (henceforth, CA) to perform seriation.

3.2.1 Correspondence Analysis and Edge Crossings

In traditional application scenarios, CA is performed on a contingency table [37]. Categorical values represented by rows (or columns) of the table is characterized by frequency distributions of the corresponding rows (or columns), which is called *profile* in CA. CA finds a low dimensional subspace of the entire profile space (e.g. a one-dimensional line or a two-dimensional plane), which maintains the majority of dispersions of the original profiles [20]. If two profiles are close to each other in the original profile space, they would also be close to each other in the identified low dimensional subspace.

In the application of reducing edge crossings between paired entity-bicluster lists, a pair of lists can be represented as a biadjacency matrix, which is binary. *1* indicates an entity is linked with a biclsuter, while *0* means not. This matrix can be converted into a contingency table, and the profiles used in CA for entities and biclusters can be formulated. For two entities, if their connected biclusters are almost the same, their profiles in CA would be similar to each other. For two biclusters, if their associated entities are almost the same, their profiles in CA are also similar and they are close to each other in the profile space. By performing CA on this biadjacency matrix, similar entities are grouped together and close to each other on the axis identified by CA. Due to the symmetric property of CA [20], with respect to rows and columns of the contingency table, the corresponding biclusters on its identified axis by CA are also grouped and follow the similar order as that of entities. If we organize entities and biclusters in lists respectively based on their corresponding orders from CA, edge crossings between different groups of entities and biclusters will be reduced, compared with a random arrangement of entities and biclusters in the list.

3.2.2 Key Steps to Enable Seriation in Lists

This merged-matrices based seriation includes five key steps that is summarized in Figure 2.

1) *Biadjacency matrices preparation.* Based on relations in each pair of neighboring lists (an entity-list and a bicluster-list), we get an adjacency matrix, where rows are entity IDs, and columns are bicluster IDs. Each cell in such matrices has a value of *0* or *1*, indicating whether an entity is connected with a bicluster. *1* means that they are connected and *0* means that they are not.

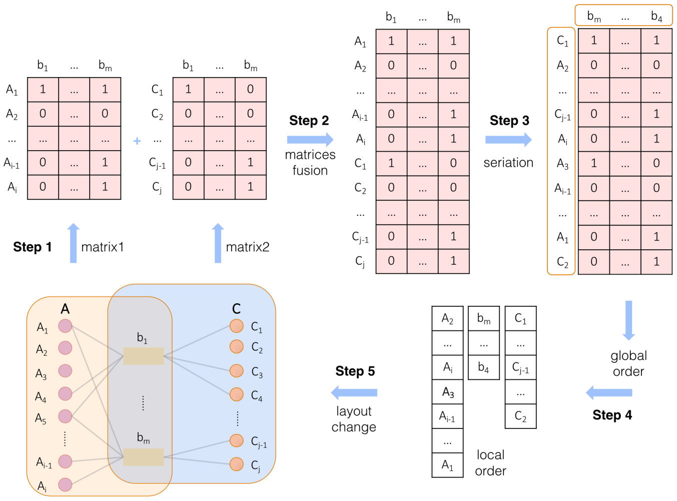

Fig. 2. Five key steps of bicluster-based seriation in a bipartite graph: 1) biadjacency matrices preparation, 2) matrices fusion, 3) seriation on the merged matrix, 4) local order generation, and 5) visual mapping.

2) *Matices fusion.* We merge these biadjacency matrices to get a fused matrix, where rows are all entity IDs and columns are all bicluster IDs from all paired neighboring lists in the previous step. When an entity is not connected with a bicluster, we fill the corresponding cell with *0*.

3) *Seriation on a fused matrix.* We apply CA to this merged matrix and get the seriated orders of entities and biclusters (as global orders), respectively. Other seriation methods can also be applied in this step. We choose CA for it can help to reduce edge crossings, as discussed before, and it has been studied for bipartite graph partitioning [38]. Moreover, based on implementations in [39], CA is effective and reasonably fast.

4) *Local order generation.* We get local orders of entities in entity-lists and local orders of biclusters in bicluster-lists based on the two global orders. For the two seriated sequences of entity IDs and bicluster IDs, we separate them into different entity-lists and bicluster-lists respectively, by their types. In each entity-list or bicluster-list, the order of entities or biclusters is determined based on their global orders.

5) *Visual mapping.* In each entity-list, entities are displayed by their local orders. In bicluster-lists, the position of biclusters is determined by the average position of their connected entities. This attempts to obtain symmetric layouts for biclusters and their associated entities for readability and aesthetic purposes [40].

This fused matrices based approach enables applying seriation to bipartite graphs (including multiple lists). If all pairs of neighboring entity-lists and bicluster-lists are included, this approach gives an organized layout for all entities and biclusters. If only one bicluster-list and its two neighboring entity-lists are involved, this approach gets an organized layout for these biclusters and entities. For the same group of lists, the former potentially gets an overall organized layout as an overview of the graph, while the latter gives them an organized layout from a focused perspective.

The seriation approach can generate layouts with fewer number of edge crossings than the greedy approach used in BiSet [6]. The greedy approach orders biclusters by size, and then it assigns orders to entities from the largest bicluster to the smallest one. If an

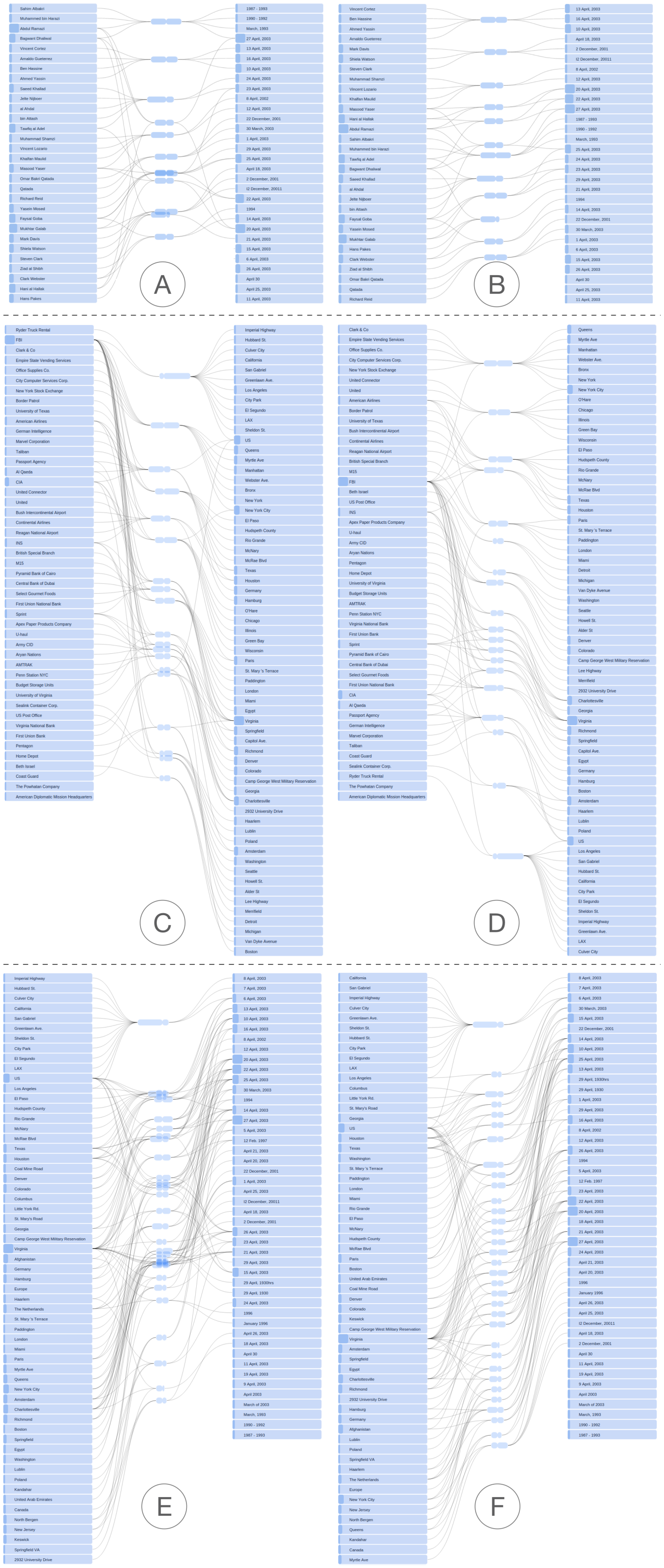

Fig. 3. Three examples of organizing one bicluster-list and two entity-lists with two approaches. (A), (C) and (E) present the greedy approach result. (B), (D) and (F) show the proposed seriation approach result.

entity is linked with multiple biclusters, its position is determined by the largest one. Finally, the position of biclusters is determined by the average position of their entities. Figure 3 show examples of organizing the same lists with the two approaches. Their detailed information is summarized in Table 1. Entities include names, locations, organizations and dates extracted from the Sign of the Crescent dataset [1], and biclusters are computed by setting the number of entities on one side as at least three. These examples show that the proposed seriation approach can generate a layout with fewer number of edge crossings. Also, it helps to address the visual overlapping problem of biclusters (e.g., see many biclusters overlap each other in (E)).

TABLE 1
A summary of the three examples shown in Figure 3.

Example	**Number of**			**edge / (entity**	**Number of Edge Crossing**		
	bicluster	entity	edge	**+ bicluster)**	greedy	seriation	reduced
(A), (B)	12	54	71	1.08	169	45	73%
(C), (D)	19	108	156	1.23	933	388	58%
(E), (F)	33	96	226	1.75	2421	1542	36%

4 User Experiment Design Rationale

4.1 Research Questions

We aim to study the effect of two techniques, *edge bundling* (visualized as two rectangles indicating edge density), and *seriation*, on sensemaking of biclusters in bipartite graphs. In order to fulfill this, we need to compare user explorations in a bipartite graph with these techniques to a graph without them. Since a bipartite graph organize information in lists, we call a bipartite graph without these techniques as a *traditional* list view and a graph using them as an *enhanced* list view. We have the following three research questions. The first one is a visual analytics oriented question, and the other two are usability oriented questions.

Q1: How can computed biclusters that are visualized using one or both of the two techniques, help users to find complex domain specific biclusters?

Q2: Compared with a traditional list view (e.g., Figure 1 (A)), does an enhanced list view with the two techniques improve user performance in bicluster explorations?

Q3: Are there any trade-offs comparing a traditional list view with an enhanced one by incorporating the two techniques for user exploration of biclusters in bipartite graphs?

4.2 User Task Design

We designed user tasks as exploring two types of biclusters, *closed* biclusters and *merged* biclusters. The former refers to biclusters computed by algorithms. The latter are domain specific biclusters that may consist information selected and merged from multiple biclusters. Finding merged biclusters needs domain knowledge that algorithms may lack (e.g., semantic meanings of entity labels). However, users can fill in the gap by bringing their domain knowledge to support merged biclusters discovery. The task of merged biclusters discovery helps us to find answers to the first research question. This also matches real-world application settings, where relationships computed from datasets do not always exactly meet user expectations, so they have to handle such results for analysis. Moreover, user explorations of the two types of biclusters allows us evaluating usability of the two techniques.

4.3 Factors Affecting Task Complexity

We identify three levels of factors that impact user task complexity. They correspond to five-level relationships (*entity*, *group*, *bicluster*, *chain* and *schema*) [41], involved in bicluster explorations.

F1. The entity and group level factor: *entity number*. The more entities are, the more user effort it takes to investigate them.

Also, the more entities belong to individual groups, instead of biclusters, the more effort users may take to find biclusters. They are considered "noise" for user explorations.

F2. The bicluster level factors: *size*, *overlap* and *number*. The bigger a bicluster is, the more information it has, which takes more user effort to explore and understand. The more biclusters overlap each other by sharing entities, the more similar biclusters are. This may need more user effort to discriminate similar ones, and select or reorganize information in them. Moreover, the more biclusters are in a dataset, the more effort users take to check each of them and find meaningful one(s).

F3. The chain and schema level factor: *domain number*. The more number of domains involved, the more information a dataset may have, which may lead to longer bicluster-chains (e.g., those connecting more number of biclusters). This requires more user effort for investigation.

These factors interleave with each other in complex correlations. For example, the *number* of biclusters relies on a specified bicluster *size*, for a given dataset. It is difficult to control or separate factors for dataset generation. Moreover, there are no existing guidelines for reasonably setting the factors and explaining how they impact user task complexity. Since factors at the bicluster level directly reveal the complexity of biclusters, we consider them for dataset generation. In order to get an initial idea about user performance corresponding to a certain setting, a pilot study is necessary. Based on its results, we can further prepare datasets with a reasonable setting for the primary user experiment.

5 Pilot Study

We conducted an informal pilot study with four volunteers, who were all graduate students from a research university. The study was performed on a 15.4-inch Macbook Pro with 2.3 GHz Intel Core i7 processor and 16GB memory. The visualization was displayed in Chrome, version 51 (64-bit), which fitted the entire screen. All participants completed study tasks with a mouse and a keyboard.

5.1 Data

We prepared the data by assigning small values to three factors at the bicluster-level. Increasing their values leads to more difficult tasks, so this setting helped us to identify a rough baseline of user performance. We generated two datasets with identical complexity and size. Each has two lists of entities (*person* and *company*), with 18 entities per list using different labels. Entities in each dataset form 9 biclusters in total, and each entity belongs to at least one of these biclusters. Moreover, these biclusters have 3 different sizes (3 biclusters per size): 2×3, 3×3 and 3×2. While in real-world applications, biclusters are computed in a larger size (e.g., 124 genes similarly behaved under 17 conditions for the Yeast and Human B-cell Lymphoma data [42]), we pick such sizes as they do not need much user effort to understand and have been used in previous study [25]. We use three different numbers of shared entities: 0, 2 and 4, corresponding to three bicluster overlap levels: *low*, *medium* and *high*. The 9 biclusters are evenly assigned to them by size. This assures that there are three different sized biclusters in each overlap level. Each pair of biclusters in the same overlap level share the corresponding number of entities. Biclusters from different overlap levels do not share any entity.

In summary, each dataset has 36 entities from two domains (*person* and *company*) and 53 individual relationships associated with at least one bicluster. We avoided isolated entities (e.g., those do not belong to biclusters), since they may increase user cognitive effort of exploration.

5.2 Tasks

Each participant was assigned two tasks, finding people with similar working experience, from the two generated datasets. The expected answers should include at least three persons and at least three companies, as evidence of their hypotheses. Entities in the two datasets remain the same order (generated with random ordering), but assigned with different labels. The difference controlled for the two tasks is the *view*. One used edge bundles (denoted as *pilot-WB*). The other was without bundles (denoted as *pilot-NB*). We did not test seriation in this pilot study. Compared with orders from seriation, randomly generated orders may need more effort and time for exploration, since entities associated with the same biclusters may be separated. Due to this, users may have to do more search with random orders. Thus, user performance with random orders potentially gives us an "upper bound" about the complexity of the current datasets.

We asked participants to find as many answers as possible, without giving them a specific number of expected answers or time limits. With this strategy, we wanted to explore when participants would stop their analysis. After finishing a task, we reviewed their findings with them and asked them to justify their answers.

5.3 Results

On average, for both views, all participants found four answers, and the majority of these answers covered three closed biclusters, sized 3×3. It took participants almost twice amount of time, on average, to finish the task in *pilot-NB* (about 9 minutes) than that in *pilot-WB*. Moreover, most answers were closed biclusters. This indicates that participants tended to stop analysis after getting all three closed biclusters. Edge bundles directly show such biclusters, so it is much easier for participants to find them.

Considering such time difference, the number of overlapped biclusters might be too complex for the view without bundles. The more biclusters overlap each other, the more complex related entities are (e.g., more involved entities with more edges). This leads to more user effort in explorations in the view without edge bundles. In *pilot-NB*, users had to check individual relationships to find an answer, because there were not obvious visual clues. This suggests that the number of overlapped biclusters in current datasets might be overbalanced for the two views.

Based on these results, we made two changes in the primary study: dataset generation and task descriptions. The former aims to balance datasets complexity for both views. The latter attempts to persuade users to continue their analysis after finding all closed biclusters. We posit that participants would explore more if they could be directed with a more clear task description (e.g., giving them the number of expected answers). With a longer period of exploration, more insightful results may be covered by users [43].

6 Primary Evaluation

6.1 Participants and Apparatus

We recruited 20 graduate students (9 males and 11 females) from our university, aged 24-33 (mean 28). Participants were from various departments, such as business management, civil engineering, computer science, food science and psychology. None of them had prior experience with biclusters. All participants had normal or corrected-to-normal vision without color vision deficiency. Similar to the pilot study, the primary study was performed on a 15.4-inch Macbook Pro with 2.3 GHz Intel Core i7 processor and 16GB memory. The visualization was displayed with Chrome, version 51 (64-bit), which fitted the entire screen. All participants completed study tasks with a mouse and a keyboard.

6.2 Data

We use synthetic data in this study to ensure generalizability of the results. We generate four datasets for four experiment conditions. They have the same level of complexity based on size and graph connectivity. Each contains two sets of entities. One entity-set is *people's name*, and the other set is *organization*, *location*, *item*, or *course*. Combinations of the two entity-sets lead to four datasets.

Considering pilot study results, we reduce the number of biclusters but increase their overlaps. The former attempts to balance the complexity for both views, since it saves user effort and time to find closed relationships. The latter increases the possibility for users to explore information from multiple biclusters. Biclusters have the same three sizes as those in the pilot study, and they have three levels of overlaps: *low*, *medium* and *high*, corresponding to sharing 1, 2 and 4 entities. In the pilot study, there were no answers that consisted of information from biclusters without sharing any entities, so we adjusted the *low* level of overlaps from 0 to 1.

For each dataset, we designed six expected answers. Two of them are closed biclusters, and others are merged ones. Three of the merged biclusters consist of two biclusters sharing 1, 2 or 4 entities. The other merged one comes from two biclusters without overlaps. For example, two groups of people have similar working experience at IT companies, although individuals may work at different companies (e.g., Google, Microsoft and Facebook).

In summary, for the primary study, we generated four datasets. Each has 58 entities with 83 individual connections in total, which leads to 14 biclusters. The size of these datasets is about 1.5 times larger than those used in the pilot study. This leads to the time for a participant to finish one task about 20 minutes.

6.3 Task and Design

We conducted a within-subjects, 2×2 factorial study with four user tasks. The two key factors are *view* and *entity order*. The former contains two levels: *with edge bundles* and *without edge bundles*. The latter also has two levels: *random order* and *seriated order*. Combinations of them lead to four experiment conditions. For each condition, a user task is assigned with one generated dataset. Considering the time cost, about 20 minutes per task (we gave extra time, about 5 minutes, in case that participants needed it), we do not replicate tasks for each experiment condition. The four experiment conditions are summarized in Table 2. To avoid order effects, the sequence of the four conditions is randomized.

TABLE 2
A summary of the four experiment conditions.

View	Order	Experiment Condition Code
No bundle	Random	*NR*
With bundle	Random	*BR*
No bundle	Seriation	*NS*
With bundle	Seriation	*BS*

The four user tasks are similar to each other, although different labels are used. Specifically, they are to find people with similar:

- *Working experience* based on companies that they worked for.
- *Travel preference* based on their travel history.
- *Shopping style* based on their shopping records.
- *Learning interests* based on the courses they have taken.

We require that each finding should contain two sets of entities (e.g., people and companies). The cardinality of each set should be in the range from 3 to 6 (including boundaries). In addition, different from the pilot study, we informed participants that there were 6 expected answers for each task, but they were free to find as many as they could. This attempted to avoid users stopping their analysis after merely finding the two 3×3 sized biclusters.

We used the proposed seriation approach to generate seriated orders for entities and biclusters. For random orders generation, we arbitrarily shuffled entities in two entity-lists, and then determined the positions of biclusters by the average position of their associated entities. In total, we generated 100 samples of such random orders and randomly select one from them for both *NR* and *NS*. Thus, entities in *NR* and *NS* have the same order based on random ordering, while entities in *BR* and *BS* remain the same order based on results of seriation.

6.4 Visualization and User Interaction

Figure 4 shows examples of visual layouts for the four conditions. In the study, we used them with different entity labels. Moreover, Figure 5 illustrates examples of three levels of bicluster overlaps. Entities are displayed in left and right lists, with a small rectangle on the left indicating its frequency. Edge bundles are displayed in between the two entity lists, on top of which two round-cornered rectangles are shown to reveal the density of entities of the bundles on both sides, as the width of the two rectangles.

Related entities are highlighted with propagations when users mouse over or select an entity. Two similar highlighting propagations are used. In a view without bundles (Figure 6 (A)), after hovering an entity in a list, connected entities in another list are highlighted. Based on these highlighted entities, all other related entities, in the same list with the hovered one, are also highlighted. In a view with bundles (Figure 6 (B)), when hovering an entity, its connected bundles are highlighted, and other entities associated with these bundles are highlighted. The former applies an *entity*-based highlighting propagation. The latter employs a *bundle*-based one. Both reveal connections between two groups of entities, based on a user-hovered entity. Moreover, when users hover or select a bundle, its connected entities are highlighted.

6.5 Procedure

This study contained four parts. It began with a brief tutorial about coordinated relationships, biclusters and edge bundling. Then we used the two datasets of the pilot study to demonstrate the two views with supported interactions. Second, as a training session, participants were asked to find a 3×3 bicluster from these two views, and we resolved their questions, if there are any.

Then, participants were informed about task description. After that, they worked on four tasks in a randomized sequence. Participants had at most 20 minutes to finish each task, and a follow-up 10 minutes to review and justify their findings. Participants were allowed to have a short break after finishing each task, if needed. It took each participant about 2 hours to finish all tasks. Finally, after they finished all tasks, we interviewed with them to learn their analysis strategies, judgment of complexity of the four tasks, and impact of edge bundle and entity order on their analysis.

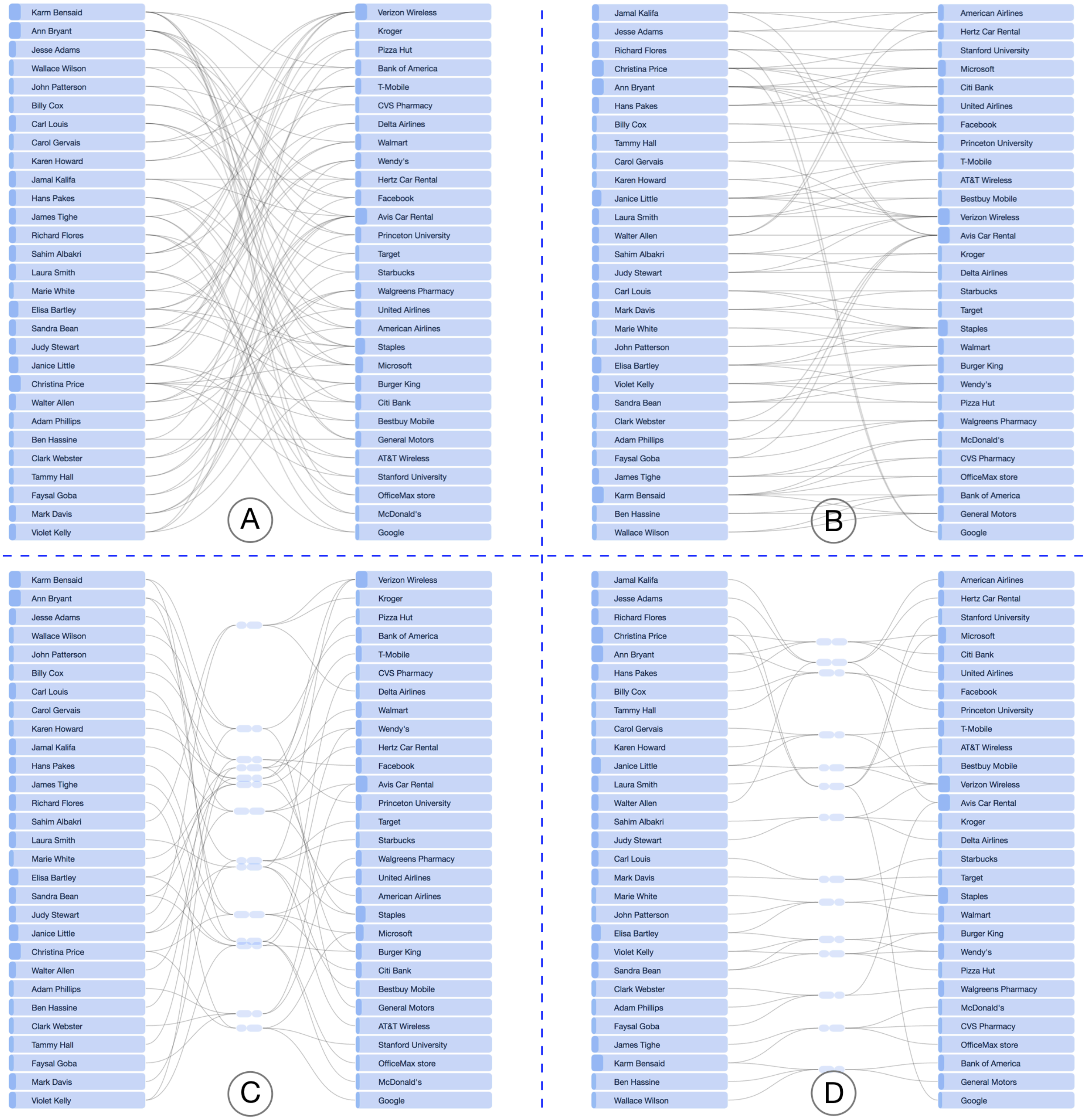

Fig. 4. Examples of visual layouts corresponding to the four experiment conditions in the primary study. (A): no bundle + random order (*NR*). (B): no bundle + seriated order (*NS*). (C): with bundle + random order (*BR*). (D): with bundle + seriated order (*BS*).

6.6 Data Collection

Data were collected from interaction logs, screen recording, observations and interviews. We logged three types of interactions during user analyses: mouse-*over* or *out* an entity or a bicluster, *selecting* or *unselecting* an entity or a bicluster, and *adding* or *removing* an entity to or from answers. For each interaction, four key components were logged: time stamp, interaction type, target object type (an entity or a bicluster), and target object ID.

6.7 Measures and Metrics

We measured user performance (for *Q1* and *Q2*) in three aspects: variance of findings, accuracy of findings, and exploration cost.

Variance of Findings. User findings include two types of biclusters: *closed* biclusters and *merged* ones. They both indicate coordinated relationships identified by participants for the given tasks. Entities of a merged bicluster come from biclusters with three levels of overlaps: *high*, *medium* and *low*, discussed before. Thus, there are four possible types of biclusters in user findings. The variance of them in user answers indicates user preference of finding biclusters in different experiment conditions. There are two possible metrics of this measure: *percent* and *count* for each type of biclusters in user answers. Because *Count* may vary a lot among participants (they were allowed to find as many answers as they wanted), we used *percent* of the four types of biclusters for

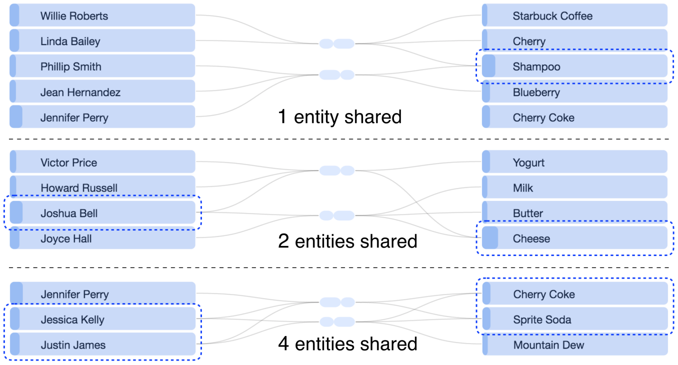

Fig. 5. Three levels of bicluster overlaps based on shared entities.

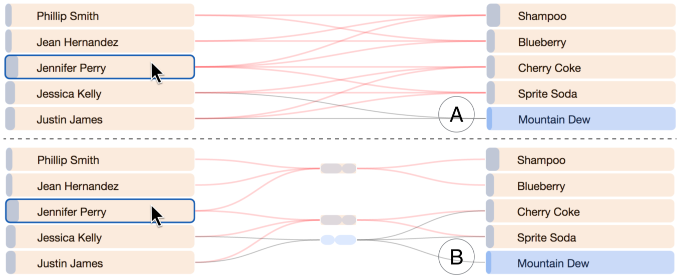

Fig. 6. Highlight propagation: entity-based (A) and bicluster-based (B). Highlighted entities are colored in orange. A user hovered or selected entity is marked with a blue border. Entities in the normal state are blue.

variance of participant findings.

Accuracy of Findings. Our designed answers serve as *gold standard* answers for evaluating participant answers. We used two rules to determine *justified* answers: 1) perfectly or partially matching expected answers (e.g., a subset of an expected answer), and 2) providing supportive evidence. If answers meet them, they are justified. Two types of supportive evidence are acceptable:

Connection based envidence: If a graph connection is provided (e.g., showing the connections between 3 people and 3 locations), then a finding is counted as a justified one.

Inference based evidence: If inference based explanations are provided (e.g., explaining that 3 people had working experience at IT companies), then a finding is considered justified.

Similar to variance of findings, we used the *percent* of justified answers as the metric to measure accuracy of participant findings.

Exploration cost. We used *entity visits* and *time* to evaluate participant exploration cost. The former refers to users interacting with entities during explorations, which indicates their interaction effort. A specific metric for entity visit is the *number of visited entities* per justified answer. Similarly, for time, we used the *time cost* per justified answer as its metric. Unjustified answers may result from random entity selections. Thus, it cannot reasonably reflect user effort on explorations. Additionally, total number of justified findings may vary among participants. The more justified answers are, the more effort users may take. Thus, we decided to use the two metrics by considering per justified answer.

6.8 Hypotheses

We made the following eight hypotheses about user performance of bicluster explorations. Specifically, *H1* and *H2* measure performance by variance of findings; *H3* and *H4* evaluate performance from the perspective of accuracy; and *H5-H8* consider performance from the perspective of exploration cost.

TABLE 3
Results of interaction test between the two factors, *view* and *order*, for the four types of biclusters in participant findings.

Bicluster Types in Participant Findings	Results of interaction test		
	$F_{(1,18)}$	p	Significant
Closed biclusters	0.266	.612	No
Merged biclusters: *high*-level overlaps	1.542	.230	No
Merged biclusters: *medium*-level overlaps	0.018	.895	No
Merged biclusters: *low*-level overlaps	0.629	.438	No

H1: With the same *order*, we expect that user findings are more likely to include *closed* biclusters when using edge bundles than without bundles. Bundles explicitly reveal closed biclusters, so it is easier for users to find them than merged ones.

H2: With the same *order*, we expect that edge bundles may lead to user finding more merged biclusters, from biclusters with higher overlap-levels, than that without bundles. Bundles help to reveal overlaps between biclusters. Users are more likely to merge biclusters with higher level of overlaps.

H3: With the same *order*, comparing a bundle-enhanced list view with a traditional one, we expect that participants find more justified answers, since bundles group some entities together.

H4: We expect that more user justified answers with seriated order than random order, with the same *view*. The former tries to place entities of the same bicluster(s) near each other, so it is more likely for users to find similar entities and group them.

H5: With the same *order*, it takes users fewer entity visits to get justified answers with bundles than without bundles, because edge bundles help to reveal entity coalitions.

H6: With the same *view*, compared with random order, we expect that it takes users fewer entity visits to find a justified answer using seriated orders. After seriation, similar entities (based on their associated biclusters) are listed near each other, so it is easier for users to find similar ones.

H7: With the same *order*, we expect that users take less time to get an answer using bundles than without bundles.

H8: With the same *view*, we expect that it takes users less time to find an answer when using seriated order than random order.

7 User Performance Results

We did 2-way repeated ANOVAs for testing the hypotheses. For assumptions of the ANOVA, we performed Shapiro-Wilk test and Mauchly's test to verify normality and sphericity of collected data, respectively. Two independent variables are *view* (with bundles vs. without bundles), and *order* (random order vs. seriated order). Dependent variables are those metrics discussed in Section 6.7.

7.1 Variance of Findings (H1 & H2)

For each type of biclusters, we checked the interaction between *view* and *order*. We found no significant interaction (see Table 3). Moreover, we checked the impact of bundles on the four types of biclusters in participant findings, shown in Figure 7.

With the same order, we found a significant impact of bundles on *closed* biclusters ($F_{(1,18)} = 4.929,\ p < .05$), merged biclusters with *medium*-level overlaps ($F_{(1,18)} = 10.892,\ p < .05$), and *low*-level overlaps ($F_{(1,18)} = 10.648,\ p < .05$). We performed post-hoc Tukey's tests (pairwise comparisons) and found that using bundles leads users to discover more closed biclusters than without bundles only for the random order group. Thus, *H1* is conditionally supported. Bundles reveal closed biclusters by linking their entities,

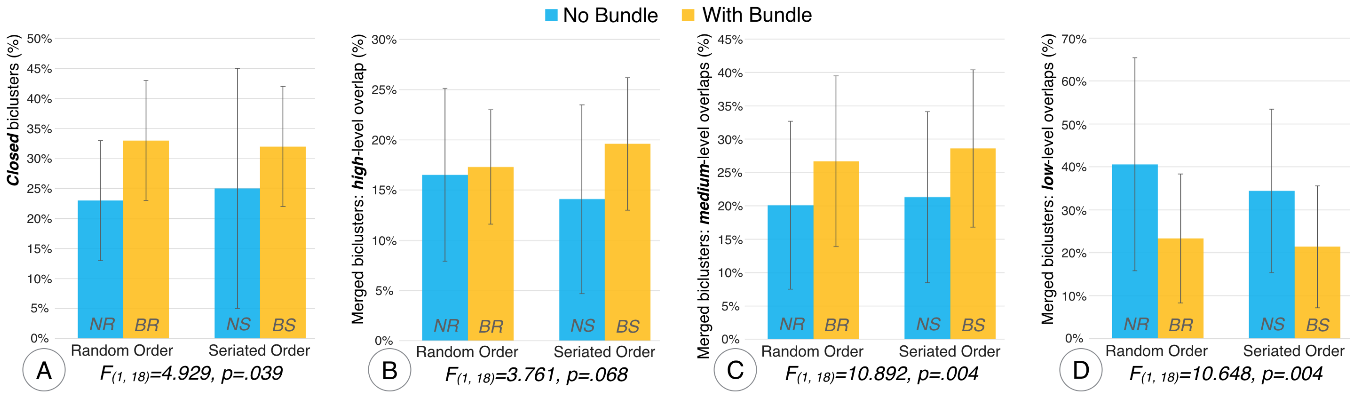

Fig. 7. A summary (mean) of the percent of the four types of biclusters in participant findings, with error bars indicating the standard deviation.

so it is easier for users to find them than without bundles. Seriated order organizes entities of the same biclusters near each other. This helps users to find closed biclusters by checking nearby entities, so the effect of bundles for the seriated order group is not significant.

For merged biclusters with *medium*-level and *low*-level overlaps, post-hoc tests reveal that using bundles leads to the increase of the former but the decrease of the latter for both random orders and seriated orders. We can also see this in Figure 7 (C) and (D). Thus, *H2* is supported when considering the *medium* and *low* overlap levels. Bundles may promote user awareness of computed entity groupings and lead them to investigate bicluster overlaps for finding answers. Without bundles, users had to manually find groupings. Lacking grouping information, participants may simply select entities that come from biclusters with a low overlap level.

For merged biclusters with *high*-level overlaps, we found no significant effect of edge bundles ($F_{(1,18)} = 3.761,\ p = .068 > .05$), with the same orders. When users select an entity, there are fewer number of entities highlighted for high-level overlapped biclusters than medium and low overlap levels. Because of fewer highlighted entities, it took participants less effort to check them, even when they are not near each other. Thus, merged biclusters with the high-level overlap in participant findings vary slightly between using bundles and without bundles for the random order condition.

With seriated orders, entities of the same biclusters are located near each other. However, without bundles, participants still need to manually check and select from neighboring entities to form groups. This takes more effort than using bundles, which explains the smaller mean of *NS* than *BS*. When entities of the same biclusters are near each other, their connecting edges spread less. This makes following edges more difficult, especially without bundles, so the mean of *NS* is smaller than that of *NR*, in Figure 7 (B). However, using bundles, when related entities are neighboring, participants may be easier to explore overlaps between biclusters. Thus, the mean of *BS* is larger than that of *BR* in Figure 7 (B). Since edge bundling has no effect on merged biclusters with *high*-level overlaps, *H2* is rejected, when comparing the *high* overlap level with the *medium* or *low* level.

7.2 Accuracy (H3 & H4)

For testing hypotheses, *H3* and *H4*, we checked the interaction between the two factors, *view* and *order*, and found no significant interactions between them ($F_{(1,18)} = 0.228, p = .639$). In addition, we checked the effect of the two factors, respectively.

With the same order, of all the participant's answers, there are more justified ones, using bundles than without bundles. We found a significant effect ($F_{(1,18)} = 14.805, p < .05$) of *view* on justified user answers. Post-hoc Tukey's tests indicate that using bundle leads to more justified answers than without bundles for both the random order group and the seriated order group. Moreover, based on Figure 8 (A), on average, there are more justified answers in *BR* and *BS*, compared with *NR* and *NS*, respectively. These results support *H3*, which means that bundles lead to participants getting more justified answers.

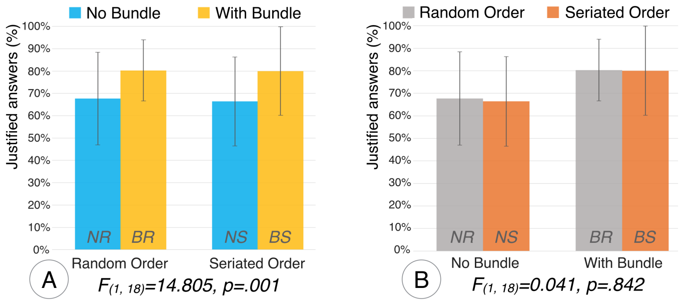

Fig. 8. A summary (mean) of the percent of justified answers in participant findings, with error bars indicating the standard deviation.

Comparing results from the same view but with different orders, (*NR*, *NS*) and (*BR*, *BS*), shown in Figure 8 (B), the percent of justified answers do not vary significantly. We found no effect ($F_{(1,18)} = 0.041, p = .842$) of *order* on justified answers. These are against *H4*, which indicates that with the same view, order does not significantly impact participants getting justified findings.

7.3 Entity Visits (H5 & H6)

For testing hypotheses *H5* and *H6*, first we found no significant interactions ($F_{(1,18)} = 0.370, p = .551$) between the two factors, *view* and *order*. Then, we checked the effect of them respectively.

With the same order, participants visited fewer entities to get a justified answer, when using bundles than without them. We found a significant impact ($F_{(1,18)} = 18.410, p < .0001$) of *view* on number of visited entities for justified answers. Post-hoc Tukey's tests show that using bundles leads to fewer entity visits than without bundles for both random orders and seriated orders. Additionally, based on Figure 9 (A), comparing results of the two pairs, (*NR*, *BR*) and (*NS*, *BS*), we can find that the mean of the number of visited entities per justified answer is smaller with bundles. These support *H5*. Edge bundles reveal entity groupings, so participants do not have to repetitiously check entity connections. Thus, they investigated fewer number of entities to find justified answers.

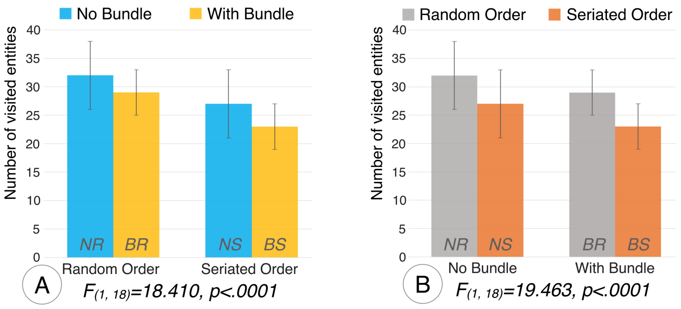

Fig. 9. A summary (mean) of the number of visited entities per justified answer, with error bars indicating the standard deviation.

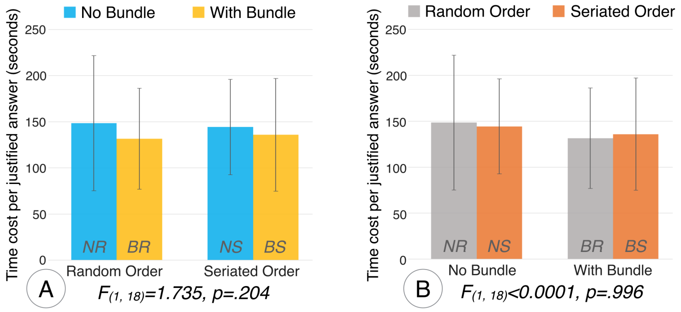

Fig. 10. A summary (mean) of time cost (seconds) per justified answer, with error bars indicating the standard deviation.

Similarly, with the same view, seriated ordering reduces entity visits for participants to find justified answers. We found order has a significant effect ($F_{(1,18)} = 19.464, p < .0001$) on the number of visited entities for justified answers. Post-hoc Tukey's tests reveal that there are fewer entity visits using seriated order than random order for both using bundles and the without bundles. In addition, comparing the results of two pairs, (*NR*, *NS*) and (*BR*, *BS*), shown in Figure 9 (B), we can find that participants visited fewer number of entities to get justified answers, with seriated orders than random orders. These support *H6*. Seriated ordering attempts to organize entities, associated with the same bicluster(s), near each other, so participants are more likely to find related entities by exploring entities around previously investigated ones. With random orders, they may have to search and check more entities before they identify useful ones for grouping. This explains the fewer number of visited entities with seriated orders.

7.4 Time Cost (H7 & H8)

The time costs per justified answer in different experiment conditions are similar, shown in Figure 10. We found no effect of view ($F_{(1,18)} = 1.735, p > .05$) and order ($F_{(1,18)} < 0.0001, p > .05$) on the time cost. Moreover, we found no significant interaction between the two factors ($F_{(1,18)} = 0.24, p = .63$). These reject *H7* and *H8*. Thus, neither edge bundling nor seriated ordering helps to reduce the time cost for users to find justified answers.

Layout interpretation may explain this. Bundles reduce entity visits by showing their coalitions. However, participants still need enough time to understand computed groupings (e.g., the category of companies). Such interpretations help them to further decide entities as findings. In addition, people tend to organize similar information spatially near each other [44]. Seriated ordering organizes entities based on graphic connections, so entities spatially close to each other are those with similar connections. However, similarity determined by graph connection may not always match their semantics (e.g., meanings of entity labels). If the two conflict, participants may take more time to understand relations of entities associated with bundles, before finally grouping some entities.

7.5 Summary of Performance Results

In summary, *H3*, *H5* and *H6* are supported, while *H1* and *H2* are conditionally supported. *H4*, *H7* and *H8* are rejected. Bundles significantly reduce entity visits (*H5*) and lead to more justified answers (*H3*), under the same order condition. Besides reducing entity visits (*H6*), order has no effect on answer accuracy and time cost. Moreover, bundles lead users to discover more closed biclusters (*H1*, with random orders), and more merged biclusters that consist of information from biclusters whose level of overlaps falls below a certain threshold (*H2*, medium-level and low-level overlapped biclusters). Neither bundles nor seriated orders impact the time cost of finding justified answers. This implies that other factors (e.g., layout interpretation) may affect the time cost, beside entity visits. These results answer *Q2*.

8 Four Trade-offs

8.1 View Simplicity vs. Task Complexity

Subjective judgement of task complexity does not always match view simplicity. Edge bundling reduce visual clutters, and seriated ordering organizes similar information spatially near each other. They attempt to get clearer views from a perspective of simplicity. With them, the number of edge crossings is reduced. Following this rationale, *BS* is the easiest condition, while *NR* is the hardest one. However, based on the interview feedback, only 7 of all the 20 participants agreed that *BS* was the easiest and *NR* was hardest, although 17 participants reported that tasks with edge bundles were easier than those without bundles.

For participants who completely or partially disagreed with the task complexity discussed above, the majority of them (over 50%) voted *BR* as the easiest condition and considered *NS* as the hardest one. They thought that there were fewer edges in *BS* than in *BR* and the same case with *NS* and *NR*. This indicates that with the same view, seriated ordering leads to perceptual illusions on fewer edges. In fact, all four tasks have the same number of edges. These participants thought it was harder to group entities with fewer edges. For example, P6 said, "*...[compared with BR], there are fewer edges [in BS], so it is harder for me to make decisions [on grouping entities]...*", and P11 mentioned, "*...[compared with NR], the graph [in NS] is sparse [with less connections]. It is more difficult to work with a sparse graph...*" The claim, "*fewer edges (or connections)*", indicates that they thought the view got simpler, but fewer connections, for them, indicates a smaller likelihood of finding possible answers.

View simplicity leads to a perceived data reduction. This seems to further reduce the user perceived opportunities to find answers. However, in fact, the task complexity does not increase, comparing seriated order with random order for the same view, since the number of edges does not change. Due to this perception illusion, a simpler view may lead to an increase of perceived task complexity.

8.2 Similarity: Connection-based vs. Semantic-driven

Similarity of entities can be determined from two aspects: *connection* or *semantic*. *Connection* based similarity means that entities are associated with the same bicluster(s), so they have similar connections with other entities. *Semantic* oriented similarity indicates

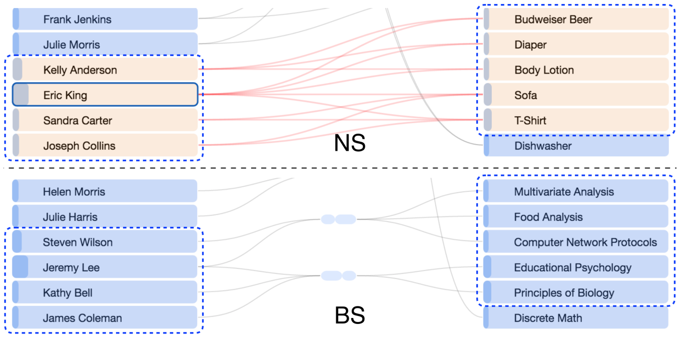

Fig. 11. An example of unreasonably justified answers in *NS* and *BS*. The selected entities are those inside the blue dotted boxes.

that the meanings of entity labels are similar (e.g., IT companies). In lists, entities similarity is revealed by spatial proximity.

Participants tend to be misled by spatial proximity of entities and make unreasonable decisions. One common type of unreasonably justified findings is simply merging entities spatially near each other. Half of the 20 participants submitted an answer in *NS*, shown in Figure 11, which aggregated two groups of entities on the bottom of two lists. When reviewing this answer, they could not give an reasonable explanation (from a semantic perspective). One popular explanation from them is "*...they are near each other, and [when hovering Eric King] they all highlight...*" This indicates that they did not pay enough attention to these entity labels for getting this finding, which result from spatial proximity of these entities. Participants can perceive such spatial proximity. For example, P5 stated, "*...entity orders [in NS] seems telling me something...[when hovering entities], highlight ones are almost aligned horizontally.*" Based on this, P5 got this answer. However, in *NR*, for the same group of entities, 3 participants chose them as a finding, which is smaller than 10 participants in *NS*.

Seriated ordering leads to a layout, where spatial proximity reveals graph connection-based similarity. Neighboring entities are potentially associated with the same biclusters. These biclusters may have different meanings. Due to this, layouts with seriated orders cannot map *semantic* level similarity between entities to their spatial proximity. However, semantic level similarity is a key factor that may impact how users organize information. If entity similarities at the two levels are not consistent, participants may get unreasonable answers. Using bundles helped this (for *Q1*).

It is hard to simultaneously reveal entity similarity at both *connection* level and *semantic* level by only using spatial proximity. Edge bundles can help, because they visually reveal groupings. Corresponding to the unreasonable finding in *NS*, for the same group entities in *BS*, shown in Figure 11, two of the 20 participants picked them as a finding. Groupings, revealed as bundles, may lead participants to consider meanings of entity labels. This helps to avoid simply merging entities, spatially near each other, together.

8.3 Connectedness vs. Coordinatedness

Two important aspects of entity coalitions are revealed by different views: *connectedness* and *coordinatedness*. The former means how entities are overall connected (e.g., strong, weak, or isolated), while the latter means how entities are specifically grouped. Thus, *connectedness* emphasizes a graphical perspective of entity coalitions, while *coordinatedness* focuses on the meaning of groupings.

Participants show different perception emphasis for different views. In *NR* and *NS*, participants tend to emphasize the perceived *connectedness* of entities; while using bundles (*BR* and *BS*), they tend to perceive *coordinatedness* of entities (for *Q1*). We observed this when participants explained their findings. Figure 12 shows an example of two findings with different perception emphasis. Entities with blue boxes in the left list and those with black border in the right list are those selected in findings.

Compared to individual edges, edge bundles reveal the *coordinatedness* of entities at the cost of perceived *connectedness*. When explaining findings, participants tended to use words, "*strongest or stronger connections*", in the view without bundles. This conveys their perceived entity connectedness. For instance, P11 explained the finding shown in Figure 12 (A), as "*...this group shows the strongest connection between three people with research university and big IT companies...*" In fact, 9 participants mentioned these words when explaining findings in *NR* and *NS*, but none of them were mentioned in *BR* and *BS*. In *BR* and *BS*, 7 of the 9 participants addressed their perceived entity coordinatedness by changing to use number. For example, P12, P17 and P20 explained the answer shown in Figure 12 (B), as "*...they [3 selected people] all visited 2 of the 3 Disney parks...*" However, no participants used number to explain their findings in *NR* and *NS*.

Comparing different ways of explanations, a list view without edge bundles helps participants to perceive entity *connectedness*. In a view with edge bundles, it is easier for users to learn entity *coordinatedness*. This indicates that edge bundles are better for *coordinatedness* oriented tasks (e.g., finding three people who all visit four locations), while a list view without bundles better fits *connectedness* oriented tasks (e.g., finding the strongest connections between people and locations).

8.4 Highlight Propagation Driven by: Entity vs. Bundle

The *entity*-based highlight propagation leads to participant deictic gestures during their analysis process. Seven participants in *NR* and *NS* used fingers to point at certain entities on the screen. Three of them used more than one finger to point at two or three highlighted entities, while others used one finger, pointing at a selected entity, and moved the mouse pointer following the edges from this entity. The latter indicates that they tried to check connections from a selected entity, and they used one finger as an additional marker for this selected entity. If multiple entities are selected, especially when near each other, additional markers may be needed. The former, using multiple fingers, indicates that participants attempted to explore *coordinatedness* of highlighted entities. *Entity*-based highlighting propagation do not reveal coordinated connections, so participants have to manually check them. However, if they selected the entities pointed by fingers, additional entities would be highlighted. In this case, they would lose the current view of highlighted entities. Thus, they used fingers to help remember these entities for further exploration.

Such physical interactions were not observed in the view with bundles, *BR* and *BS*. This suggests that bundles may help users remember a group of entities. Six participants used the word, "*power strip*", to depict the role of edge bundles, which helped them find and retrieve a group of connected entities from two lists. Thus, even multiple groups of entities are highlighted, participant can use bundles to distinguish different groups. Since bundles work as additional visual markers, they did not use fingers during their explorations in *BR* and *BS*. Considering user physical interactions, for the view with *entity*-based highlighting propagation, additional visual markers or extra highlighting mechanisms may be needed.

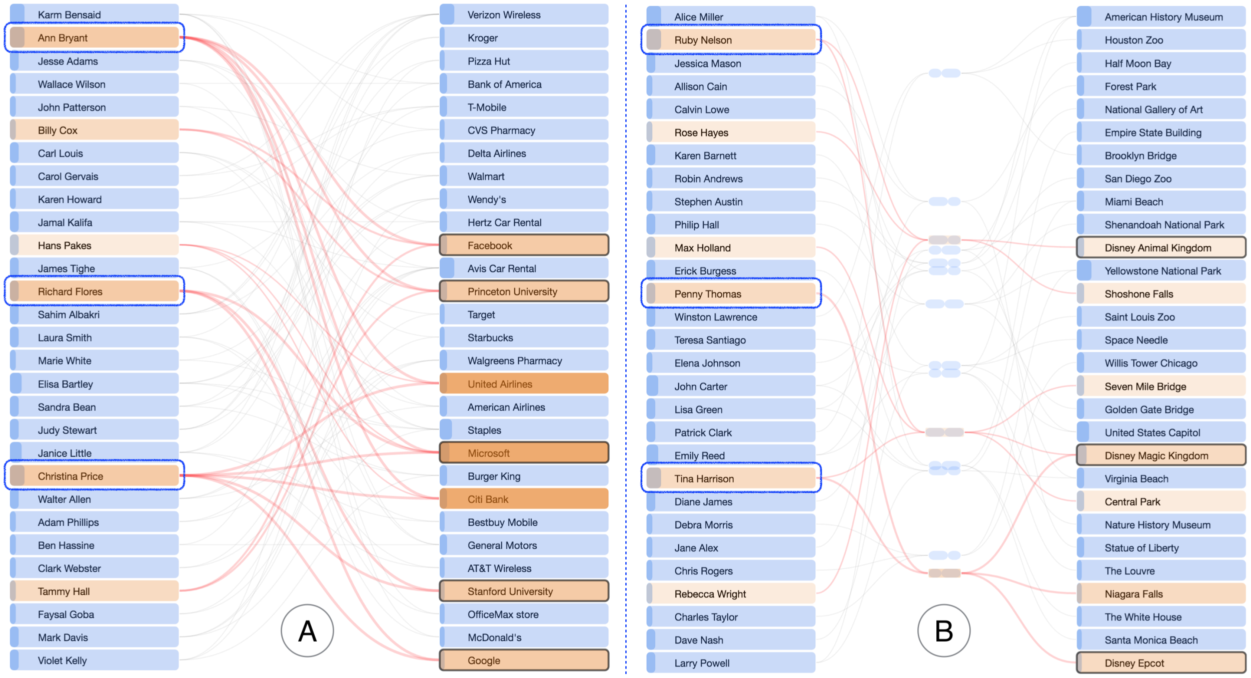

Fig. 12. An example of findings with different perception emphasis in their explanations. (A) shows an answer in *NR*, explained with emphasis on entity *connectedness*. (B) presents a finding in *BR*, with a *coordinatedness* oriented explanation.

9 Discussion

We evaluate *edge bundling* and the proposed *seriation* technique for sensemaking of biclusters in bipartite graphs. With the same *order*, edge bundles can lead to more justified findings with fewer entity visits. With the same *view*, seriated order takes fewer entity visits for users to find justified answers. These answer *Q2*. We have identified four key trade-offs regarding using the techniques (*Q3*). Such results suggest that edge bundling is critical for exploring bicluster in bipartite graphs, which promotes user awareness of the meanings of entity labels (*Q1*).

9.1 The Role of Edge Bundling

Complex tasks may not be directly solved by computation. Human effort remains necessary (e.g., users select and merge information from biclusters to get answers). While computed biclusters reveal some patterns of data, they cannot cover all answers of complex tasks. However, computed entity coalitions can still benefit users in exploring biclusters, by enabling them to see entity groupings as edge bundles and reducing entity visits to find answers.

Besides data pattern discovery, another key role of computation for visual analytics is to **free users from low-level perceptual problems and support making high-level inferences**. Based on the study results, individual edges are bundled based on computed biclusters. The bundles reduce user effort of entity visits to explore biclusters and increase justified answers. Moreover, edge bundles help users overcome perceptual problems caused by spatial proximity of entities, which helps to prevent them from making wrong judgements and being misled by spatial closeness, and leads them to consider the meaning of entity labels for high-level inference (e.g., identifying a group students as HCI students based on the courses taken). Such benefit make it critical to use edge bundling for supporting sensemaking of biclusters in bipartite graphs.

9.2 Implication for Tools of Visualizing Relations

Edge bundling empowers applications that visualize relationships for sensemaking tasks (e.g., Jigsaw [45]). Individual edges present simple relationships between entities, but they lack the capability to show entity coalitions (e.g., connected groups of entities), due to missing visual marks to reveal groupings. While highlighting helps to reveal entity groupings, it cannot serve as a handler that enables direct manipulation on groups of entities (e.g., dragging and moving them). Based on study results, besides enabling users to see entity groupings, edge bundles help them overcome perception problems and make high-level inferences. Additionally, edge bundling can be applied to variant layouts, not limited to bipartite graphs, so it is flexible enough to be applied to reveal groupings between different visualizations (e.g., connecting geolocations in a map and points in a scatterplot). Thus, edge bundling offers a good solution to support exploring complex relations,

The trade-offs indicate three key considerations to apply edge bundling and seriation for sensemaking of biclusters in bipartite graphs: 1) user-controlled dynamic ordering, 2) task-driven layout selection, and 3) coordinatedness highlight. Instead of using static orders, enabling users to dynamically organize entities and biclusters is useful to support analysis, because a simple layout does not always lead to a perceived simple task. Besides this, it is hard to encode both *semantic* oriented similarity and *connection* based similarity with one spatial order, so dynamic ordering allows users to explore data from different perspectives. Moreover, different representations may fit different user tasks. For *coordinatedness* oriented tasks, edge bundles work better, while for *connectedness* oriented tasks, a view without bundles may lead to better results. However, if users have to deal with both types of tasks, enabling them switch representations is a possible solution. In this case, we may consider adding coordinatedness oriented highlighting to the view without bundles, or enabling users to switch from individual edges to bundles (for the groups of entities under investigation), and vice versa. This helps to reveal both *coordinatedness* and *connectedness* of entities, without costing extra physical interactions (e.g., pointing at entities with fingers).

The size and density of a bipartite graph impacts user exploration of biclusters and visualization design. The larger a bipartite graph is, the more entities it has. Users have to check more entities, especially considering cases of merged biclusters discovery. As the number of entities increases, lists grow, so it takes more user effort for navigation. Moreover, given a bipartite graph with a fixed size, its density impacts the number of computed biclusters. The more dense a graph is, the more biclusters may be mined from it, and the more edges in a list-based layout. While edge bundles can help to reduce the number of edges that are displayed, a larger number of biclusters takes more user effort (e.g., investigating more entities). In some extreme cases, a layout with edge bundling and seriated ordering may remain cluttered. For such cases, other visual representations (e.g., matrix) should be considered. Furthermore, when edges are modeled with probability as real numbers, instead of a binary fashion (i.e., 0 or 1), matrix and other visualizations (e.g., BiDot [46]) can better reveal this than a list-based layout.

9.3 Study Limitations

This study does not investigate other bipartite graph visualizations, such as matrix based visualizations that can reduce visual clutters (e.g., edge crossings) for datasets with dense entity-connections. Reordering a matrix and highlighting its cells of biclique structures can also support exploring biclusters, so a comparative study with matrix based visualizations helps to validate findings of this study.

As the first attempt to evaluate the usability of edge bundling and seriation for sensemaking of biclusters in bipartite graphs, we used a fixed setting. Due to lacking existing guidelines, we did a pilot study to determine a setting of size-related factors, discussed in Section 4.3, which can impact task complexity. Because we did not used variant settings, how well the two techniques work for data with different sizes remains unclear. Also, some factors depend on others (e.g., bicluster number and overlap level rely on size). A systematic way of designing various levels of these size-related factors for future studies still needs exploration. Moreover, datasets used in our study is relatively small, which cannot match the size of data in real world. Thus, a study with real-world data is worthwhile to further validate findings of this study.

The total number of participants is relatively small, which may impact results, especially performance related findings. While this study steps first toward a better understanding of the effect of edge grouping and ordering on exploring biclusters in bipartite graphs, a study with more subjects can further verify our results, and may lead to deeper findings. Moreover, considering the tested number of hypotheses, studies with more subjects remains needed.

Familiarity with entity labels may impact study results. While task orders were randomized, we did not change the combination between datasets and experiment conditions. This leads to a fixed task domain for each condition. Participants were assigned tasks with a random order, but they may be more familiar with labels in one dataset than those in another. This may impact study results. In order to gain a better understanding the impact of familiarity with entity labels on user performance, further studies is needed.

Two similar highlight propagations, entity-based and bundle-based, used in this study, attempt to reduce user effort in checking entity groupings. As they are associated with two views, with and without bundles respectively, we did not consider them as another factor. They may impact user performance in exploring biclusters. However, if we used static visualizations without any highlighting techniques, we expect that users would perform better when using edge bundles and seriation than without them, because they help to reduce the number of edge crossings. Without highlighting, users have to trace edges to find biclusters. A larger number of edge crossings would take users more effort to trace edges. Moreover, if highlighting was considered as another factor, it should be combined with other factors for the study design. This would result in some experiment conditions as using a view with edge bundles and entity-based highlighting or a view without edge bundles but using bundle-based highlighting. For such conditions, highlighting techniques do not match layouts, so they may confuse users. Thus, considering these two situations, instead of ignoring highlighting technique, or simply treating them as a third factor, a reasonably consistent way of entity highlighting between two views (with and without edge bundles) still needs exploration. Based on this, we can further verify findings of this study, and evaluate the effect of different highlighting techniques.

10 Conclusion

In this work, we propose a bicluster-based seriation technique that can help to reduce edge crossings in bipartite graphs drawing. We conduct user experiments to study the effect of edge bundling and seriation on sensemaking of biclusters in bipartite graphs. Results of our study suggest that edge bundling is critical for exploring biclusters in bipartite graphs by reducing entity visits, leading to more justified answers, and overcoming perception problems. We have also identified four trade-offs that lead to useful implications for visualizing relationships to support sensemaking tasks. These results are a significant step toward an in-depth understanding of the two techniques for sensemaking of biclusters.

Acknowledgments

This research was supported in part by NSF grant IIS-1447416.

References

[1] F. Hughes and D. Schum, “Discovery-proof-choice, the art and science of the process of intelligence analysis-preparing for the future of intelligence analysis,” *Joint Military Intelligence College*, 2003.

[2] C. Marforio, H. Ritzdorf, A. Francillon, and S. Capkun, “Analysis of the communication between colluding applications on modern smartphones,” in *Proc. of ACM Computer Security Applications Conf.*, 2012, pp. 51–60.

[3] D. B. Allison, X. Cui, G. P. Page, and M. Sabripour, “Microarray data analysis: from disarray to consolidation and consensus,” *Nature reviews genetics*, vol. 7, no. 1, pp. 55–65, 2006.

[4] P. Fiaux, M. Sun, L. Bradel, C. North, N. Ramakrishnan, and A. Endert, “Bixplorer: Visual analytics with biclusters,” *Computer*, vol. 46, no. 8, pp. 90–94, 2013.

[5] M. Streit, S. Gratzl, M. Gillhofer, A. Mayr, A. Mitterecker, and S. Hochreiter, “Furby: Fuzzy force-directed bicluster visualization,” *BMC bioinformatics*, vol. 15, no. 6, p. 1, 2014.

[6] M. Sun, P. Mi, C. North, and N. Ramakrishnan, “Biset: Semantic edge bundling with biclusters for sensemaking,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 310–319, 2016.

[7] M. J. Zaki and C.-J. Hsiao, “Efficient algorithms for mining closed itemsets and their lattice structure,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 462–478, 2005.

[8] T. Uno, T. Asai, Y. Uchida, and H. Arimura, “An efficient algorithm for enumerating closed patterns in transaction databases,” in *International Conference on Discovery Science*. Springer, 2004, pp. 16–31.

[9] K. A. Cook and J. J. Thomas, “Illuminating the path: The research and development agenda for visual analytics,” Pacific Northwest National Laboratory (PNNL), Richland, WA (US), Tech. Rep., 2005.

[10] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers, “Visualizing sets and set-typed data: State-of-the-art and future challenges,” in *EuroVis STAR*, 2014, pp. 1–21.

[11] S. C. Madeira and A. L. Oliveira, “Biclustering algorithms for biological data analysis: a survey,” *IEEE/ACM Trans. on Computational Biology and Bioinformatics*, vol. 1, no. 1, pp. 24–45, 2004.

[12] I. Liiv, "Seriation and matrix reordering methods: an historical overview," *Statistical Analysis and Data Mining*, vol. 3, no. 2, pp. 70–91, 2010.

[13] L. Wilkinson and M. Friendly, "The history of the cluster heat map," *The American Statistician*, 2012.

[14] C. Perin, P. Dragicevic, and J.-D. Fekete, "Revisiting bertin matrices: New interactions for crafting tabular visualizations," *IEEE Transac. on Visualization and Comp. Graphics*, vol. 20, no. 12, pp. 2082–2091, 2014.

[15] G. A. Grothaus, A. Mufti, and T. Murali, "Automatic layout and visualization of biclusters," *Algorithms for Molecular Biology*, vol. 1, no. 1, pp. 1–15, 2006.

[16] J. Chuang, C. D. Manning, and J. Heer, "Termite: Visualization techniques for assessing textual topic models," in *Proc. of the International Working Conf. on Advanced Visual Interfaces*. ACM, 2012, pp. 74–77.

[17] W. S. Robinson, "A method for chronologically ordering archaeological deposits," *American antiquity*, vol. 16, no. 4, pp. 293–301, 1951.

[18] Z. Bar-Joseph, D. K. Gifford, and T. S. Jaakkola, "Fast optimal leaf ordering for hierarchical clustering," *Bioinf.*, vol. 17, pp. S22–S29, 2001.

[19] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2002.

[20] M. Greenacre, *Correspondence Analysis in Practice*. CRC press, 2007.

[21] S. Barkow, S. Bleuler, A. Prelić, P. Zimmermann, and E. Zitzler, "Bicat: a biclustering analysis toolbox," *Bioinformatics*, vol. 22, no. 10, pp. 1282–1283, 2006.

[22] J. Heinrich, R. Seifert, M. Burch, and D. Weiskopf, "Bicluster viewer: a visualization tool for analyzing gene expression data," in *International Symposium on Visual Computing*. Springer, 2011, pp. 641–652.

[23] R. Santamaría, R. Therón, and L. Quintales, "Bicoverlapper 2.0: visual analysis for gene expression," *Bioinformatics*, vol. 30, no. 12, pp. 1785–1786, 2014.

[24] J. P. Gonçalves, S. C. Madeira, and A. L. Oliveira, "Biggests: integrated environment for biclustering analysis of time series gene expression data," *BMC research notes*, vol. 2, no. 1, p. 124, 2009.

[25] M. Sun, L. Bradel, C. L. North, and N. Ramakrishnan, "The role of interactive biclusters in sensemaking," in *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2014, pp. 1559–1562.

[26] A. Lhuillier, C. Hurter, and A. Telea, "State of the art in edge and trail bundling techniques," in *Computer Graphics Forum*, vol. 36, no. 3, 2017, pp. 619–645.

[27] D. Holten and J. J. Van Wijk, "Force-directed edge bundling for graph visualization," *Comp. Graphics Forum*, vol. 28, no. 3, pp. 983–990, 2009.

[28] W. Cui, H. Zhou, H. Qu, P. C. Wong, and X. Li, "Geometry-based edge clustering for graph visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1277–1284, 2008.

[29] B. Bach, N. H. Riche, C. Hurter, K. Marriott, and T. Dwyer, "Towards unambiguous edge bundling: Investigating confluent drawings for network visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 541–550, 2017.

[30] D. Holten, "Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data," *IEEE Transactions on visualization and computer graphics*, vol. 12, no. 5, pp. 741–748, 2006.

[31] F. McGee and J. Dingliana, "An empirical study on the impact of edge bundling on user comprehension of graphs," in *Proc. of International Working Conf. on Advanced Visual Interfaces*, 2012, pp. 620–627.

[32] F. Du, N. Cao, J. Zhao, and Y.-R. Lin, "Trajectory bundling for animated transitions," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 289–298.

[33] M. Behrisch, B. Bach, N. H. Riche, T. Schreck, and J.-D. Fekete, "Matrix reordering methods for table and network visualization," *Computer Graphics Forum*, vol. 35, no. 3, pp. 693–716, 2016.

[34] P. Eades and N. C. Wormald, "Edge crossings in drawings of bipartite graphs," *Algorithmica*, vol. 11, no. 4, pp. 379–403, 1994.

[35] H. Siirtola and E. Mäkinen, "Constructing and reconstructing the reorderable matrix," *Information Visualization*, vol. 4, no. 1, pp. 32–48, 2005.

[36] S. Cheng and K. Mueller, "The data context map: Fusing data and attributes into a unified display," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 1, pp. 121–130, 2016.

[37] A. Agresti and M. Kateri, *Categorical data analysis*. Springer, 2011.

[38] H. Zha, X. He, C. Ding, H. Simon, and M. Gu, "Bipartite graph partitioning and data clustering," in *Proc. of International Conference on Information and Knowledge Management*. ACM, 2001, pp. 25–32.

[39] J.-D. Fekete, "Reorder. js: A javascript library to reorder tables and networks," in *IEEE VIS 2015*, 2015.

[40] C. Bennett, J. Ryall, L. Spalteholz, and A. Gooch, "The aesthetics of graph visualization." *Computational Aesthetics*, pp. 57–64, 2007.

[41] M. Sun, C. North, and N. Ramakrishnan, "A five-level design framework for bicluster visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1713–1722, 2014.

[42] H. Cho, I. S. Dhillon, Y. Guan, and S. Sra, "Minimum sum-squared residue co-clustering of gene expression data," in *Proceedings of the International Conference on Data Mining*. SIAM, 2004, pp. 114–125.

[43] C. North, "Toward measuring visualization insight," *IEEE Computer Graphics and Applications*, vol. 26, no. 3, pp. 6–9, 2006.

[44] C. Andrews, A. Endert, and C. North, "Space to think: Large high-resolution displays for sensemaking," in *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2010, pp. 55–64.

[45] J. Stasko, C. Görg, and Z. Liu, "Jigsaw: Supporting investigative analysis through interactive visualization," *Information visualization*, vol. 7, no. 2, pp. 118–132, 2008.

[46] J. Zhao, M. Sun, F. Chen, and P. Chiu, "Bidots: Visual exploration of weighted biclusters," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 195–204, 2018.

Maoyuan Sun is an Assistant Professor in the Department of Computer and Information Science at the University of Massachusetts Dartmouth. His research falls in areas of visual analytics, information visualization, and human computer interaction, with applied domains in intelligence analysis, business intelligence, cyber security and STEM education.

Jian Zhao is a Research Scientist at the FX Palo Alto Laboratory. His research interests, broadly, are information visualization, and human-computer interaction. His work contributes to the design, development, and evaluation of highly interactive visualization systems to enable data enthusiasts to effectively discover and communicate insightful knowledge in real-world applications and datasets.

Hao Wu received his Ph.D. from Virginia Tech. His research interests include machine learning, data analytics, and deep learning, particularly, multivariate machine learning models on discrete count data, unsupervised learning from relational data with maximum entropy models, and applying deep learning models to event data.

Kurt Luther is an Assistant Professor of Computer Science at Virginia Tech. He directs the Crowd Intelligence Lab, an interdisciplinary research group that explores how crowdsourcing can support creativity and discovery. His current projects focus on crowd leadership, crowd-supported investigations, and crowdsourced analysis of visual material, with applications to national security, journalism, history, and biology.

Chris North is a Professor of Computer Science at Virginia Tech. He is the associate director of the Discovery Analytics Center. His research seeks to create effective methods for human-in-the-loop analytics of big data. His work falls in areas of visual analytics, information visualization, human-computer interaction, large high-resolution display and interaction techniques, and visualization evaluation methods, with applied work in intelligence analysis, cyber security, bioinformatics, and GIS.

Naren Ramakrishnan is the Thomas L. Phillips Professor of Engineering at Virginia Tech. He directs the Discovery Analytics Center, a university-wide effort that brings together researchers from computer science, statistics, mathematics, and electrical and computer engineering to tackle knowledge discovery problems in important areas of national interest, including intelligence analysis, sustainability, and electronic medical records.