

BiSet: Semantic Edge Bundling with Biclusters for Sensemaking

Maoyuan Sun, Peng Mi, Chris North and Naren Ramakrishnan

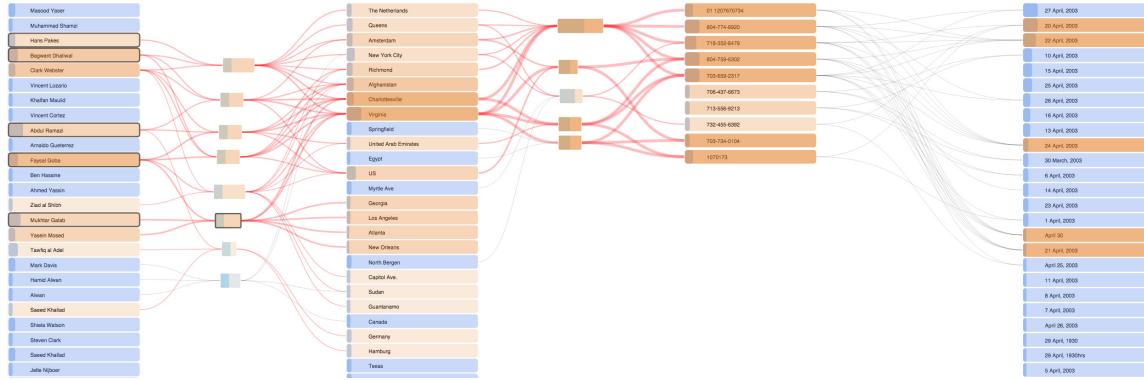


Fig. 1. An overview of BiSet. Entities are represented in lists. In the space between each neighboring pair of lists, BiSet adds a “in-between” layer, displaying edges. BiSet bundles edges based on biclusters and allows users to directly manipulate bundles. The bundles can reveal task-oriented semantic insights about coordinated relationships. BiSet also applies accumulated highlighting to entities, bundles and edges to indicate highly shared entities and relationships.

Abstract— Identifying coordinated relationships is an important task in data analytics. For example, an intelligence analyst might want to discover three suspicious people who all visited the same four cities. Existing techniques that display individual relationships, such as between lists of entities, require repetitive manual selection and significant mental aggregation in cluttered visualizations to find coordinated relationships. In this paper, we present BiSet, a visual analytics technique to support interactive exploration of coordinated relationships. In BiSet, we model coordinated relationships as biclusters and algorithmically mine them from a dataset. Then, we visualize the biclusters in context as bundled edges between sets of related entities. Thus, bundles enable analysts to infer task-oriented semantic insights about potentially coordinated activities. We make bundles as first class objects and add a new layer, “in-between”, to contain these bundle objects. Based on this, bundles serve to organize entities represented in lists and visually reveal their membership. Users can interact with edge bundles to organize related entities, and vice versa, for sensemaking purposes. With a usage scenario, we demonstrate how BiSet supports the exploration of coordinated relationships in text analytics.

Index Terms—Bicluster, coordinated relationship, semantic edge bundling

1 INTRODUCTION

Analysts often face difficult challenges in exploring complex relations and identifying meaningful ones for sensemaking [39]. Current visual analysis tools emphasize individual relationships and just display simple ones. This makes it hard for analysts to see more complex relationships (e.g., coordinated relationship). Coordinated relationships are grouped relations between sets of entities of different types (e.g., three people who all visited the same four cities). Due to the complexity, compared with simple relationship, coordinated relationship needs more cognitive effort for exploration.

Existing techniques that display individual relationships, such as between lists of entities, require repetitive manual selection and significant mental aggregation in cluttered visualizations to find coordinated relationships. For example, Jigsaw [19] provides a List View to support exploring relationships between lists of entities (e.g., people, location, date, organization, etc.). In the List View, Jigsaw applies visual links between related entities to show their connections and controls the shading of colors for entities to indicate their co-occurrence. With these visual encodings, in Jigsaw, users can recognize relations

between entities without much effort, but these relations are limited to simple individual ones (e.g., a person visited three cities). Users have to repetitiously click entities, visually check and mentally compare their linked entities to identify coordinated relationships. Since Jigsaw’s List View does not provide clear visual clues on coordinated relations, users have to manually test all possible entities before they finally find a meaningful one. This potentially forces users to solve a combinatorial problem of selection without much support. Thus, due to deficient clues to direct user selections, tools like Jigsaw have limited capabilities to support exploring coordinated relationships.

Visual analytics can potentially better support this by computationally finding complex relationships and revealing them in context. This enables analysts to see complex relations with other data (e.g., entities in lists). Specifically, we can compute coordinated relationships with biclustering algorithms and display them in context using edge bundling. In this case, edge bundles can reveal semantic insights from coordinated relationships, which is meaningful from a task-oriented perspective. The reason is that edges are bundled using semantic edge bundling that is based on results of biclustering algorithms, rather than using spatial edge bundling which is based on spatial proximity to simplify visual representations.

Biclustering algorithms compute coordinated relationships as biclusters. A bicluster can be considered a grouped relationship between two sets of entities, where each entity in one set is connected with all in another. Figure 2 shows an example of a bicluster that indicates a coordinated relation between three people and four locations. It is clear that a bicluster can bundle edges that link pairs of related entities, and group entities that belong to the same coordinated set. Biclusters provide a conceptual format to present coordinated relationships in an

• Maoyuan Sun, Peng Mi, Chris North and Naren Ramakrishnan are all with the Discovery Analytics Center, Department of Computer Science, Virginia Tech. E-mail: {smaoyuan | mipeng | north | naren}@cs.vt.edu.

Manuscript received 31 Mar. 2015; accepted 1 Aug. 2015; date of publication xx Aug. 2015; date of current version 25 Oct. 2015.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

organized manner. To take advantage of this for sensemaking, a five-level design framework for bicluster visualizations has been proposed in [49]. However, existing techniques are inefficient to support exploring coordinated relationships, and few attempt to adapt biclusters to facilitate this by following the design framework. Thus, it is still challenging to design a technique that can take advantage of biclusters and make them usable to support coordinated relationship explorations.

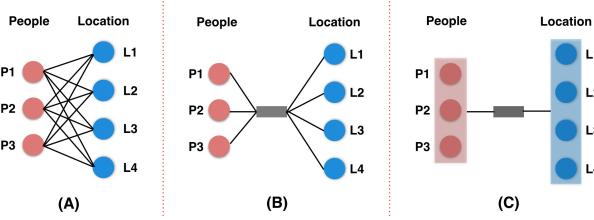


Fig. 2. An example of a bicluster, indicating a coordinated relationship between three people and four locations. (A) presents all connections between each pair of related entities from the two domains. (B) shows the result of bundling edges in this bicluster. (C) demonstrates the result of both bundling edges and grouping entities in this bicluster.

To address such challenges, we present BiSet, a visual analytics technique to support interactively exploring coordinated relationships with biclusters. Our key contributions in this paper are as follows:

- 1) We formalize coordinated relationships as biclusters and algorithmically mine them from a dataset.
- 2) We visualize the biclusters in context as bundled edges between sets of related entities. These bundles enable analysts to infer semantic insights about potentially coordinated activities.
- 3) We make bundles as the first class objects and add a new layer “in-between” lists to contain these bundle objects. We allow users to direct manipulate bundles for organizing entities represented in lists.
- 4) We apply interactions to both edge bundles and entities for revealing and organizing relevant information in a bidirectional way. Users can interact with edge bundles to forage and organize relevant entities and, vice versa, for sensemaking purposes.
- 5) We present a usage scenario to demonstrate how BiSet can support the coordinated relationship exploration in text analytics.

2 RELATED WORK

Four key aspects are involved in BiSet: **biclustering**, **list layout**, **visual link** and **edge bundling**, which outlines the discussion of related work.

2.1 Biclusters and Bicluster-Chains

Biclustering attempts to find *both* subsets of entities *and* subsets of dimensions with the restriction that for each identified subset of entities, they identically behave within the corresponding subset of dimensions [36]. **Biclusters** are computational results from biclustering algorithms that identify coordinated relations between two entity sets. An entity set refers to a set of unique objects from a specific domain (e.g., people) extracted from a dataset (e.g., documents).

Relationship between two entity sets. Given two entity sets E and F , a (binary) relationship $R(E, F)$ between E and F is a subset of $E \times F$ (the Cartesian product of E and F). We say that E is connected to F . There are different ways to model relationship R in different scenarios. In text analytics, R can be determined by word co-occurrence in documents or semantic meanings identified with natural language processing. For example, person X is related to city Y , since they are mentioned in the same document or based on semantic meanings of some sentences that indicate person X visited city Y .

Bicluster. We define a *bicluster* (E', F') on $R(E, F)$ as a set $E' \subseteq E$ and a set $F' \subseteq F$ such that $E' \times F' \subseteq R$. That is, there is a relationship between each element of E' with every element of F' . We use $|E'| + |F'|$ to denote the **size** of a *bicluster* (E', F') where $|E'|$ and $|F'|$ are the cardinality of E' and F' . In addition, *bicluster* (E', F') is **thin** if there is only one entity in either E' or F' .

Closed bicluster. A *bicluster* (E', F') is *closed* if:

- (i) For every entity $e \in E - E'$, there is some entity $f \in F'$ such that $(e, f) \notin R$, and
- (ii) For every entity $f \in F - F'$, there is some entity $e \in E'$ such that $(e, f) \notin R$.

Algorithms for bicluster mining typically aim to find closed biclusters. These algorithms (e.g., CHARM [56] and LCM [51]) function level-wise with regard to one domain (e.g., E), wherein they attempt to mine closed biclusters involving one entity of E , then closed biclusters involving two entities of E , and so on. The key parameter influencing such mining is the size of a bicluster in terms of the other domain (e.g., F), also referred to as the *minimum support threshold*. The setting of this parameter is done heuristically by users; a low threshold will yield a plethora of biclusters whereas a stringent (high) threshold will yield few (or no) biclusters. Typically, users begin with a high threshold and gradually lower it until it yields a sufficient number of biclusters [56]. In this paper, we use CHARM and LCM, although any biclustering algorithm can be utilized in BiSet.

Biclusters logically aggregate multiple individual relations to form coordinated sets, so they provide an opportunity to visually bundle edges between entities. Bicluster-based edge bundles organize edges in a semantic manner, potentially revealing semantic insights. For example, four suspicious people may collude about a terrorist attack, since they are all related to the same three terrorist organizations. This is different from spatial edge bundling that bundle edges based on spatial proximity to reduce visual clutter [58].

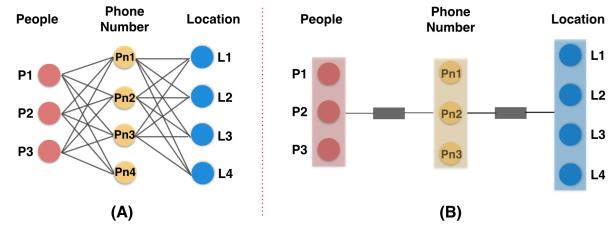


Fig. 3. An example of a bicluster-chain consisting of two biclusters. (A) presents all edges between related entities. (B) shows that the two biclusters connect together as a chain by their shared phone numbers.

Bicluster-chains. Based on shared entities, if there are any, multiple biclusters (consisting of different pairs of domains) can connect to form bicluster-chains. With compositional mining methods [29, 54], bicluster-chains can be identified from a dataset. Figure 3 shows an example of a bicluster-chain with two biclusters. One shows coordinated relations between three people and four phone numbers, and the other presents relations between three phone numbers and four locations. They share three phone numbers. One possible semantic insight revealed from this chain is: three people may visit the same four cities, since they called each other via four phone numbers, and phone calls from three of these numbers were all reported at the four cities.

A **five-level design framework for bicluster visualizations** has been proposed based on five hierarchical levels of relationships potentially existing in a dataset [49]. Keywords corresponding to the five levels are: **entity**, **group**, **bicluster**, **chain** and **schema**. **Entity-level** relations refer to those between two individual entities, while **group-level** relations are relations between one individual entity and a group of entities. **Bicluster-level** and **chain-level** relations represent two levels of coordinated relations: biclusters and bicluster-chains. The latter is more complex than the former, since a bicluster-chain consists of **multiple biclusters**. **Schema-level** relations indicates database-like patterns in a dataset, which reveals the overview of a dataset. **Relations in higher levels** (e.g., **bicluster-level** and **chain-level**) are usually constructed based on those in lower levels (e.g., **entity-level** and **group-level**), so relations in lower levels provide a critical support for the exploration and interpretation of those in higher levels. These five levels of relations systematically present the space of relationship, which works as an important guideline for us to follow. Specifically, it guides us to identify potential tasks that BiSet needs to support, by considering the implicit linkage of these five levels.

2.2 Visualizations for Exploring Coordinated Relationships

List and matrix views are two layouts that can potentially support exploring coordinated relationships. Two corresponding visualization techniques are parallel coordinates [28] and scatterplot matrix [6]. The former uses spatial position to present attributes as individual axes at once, while the latter embeds multiple scatterplots in a matrix layout that shows all pairwise combinations of attributes [37]. They are useful for discovering correlation, which potentially can be adapted for exploring coordinated relationship. Brushing is a common interaction technique in both of them to explore correlation [4, 45] and it helps users manually find correlations by selecting a group of records in an axis or in a region. However, similar to the problem of Jigsaw’s List View, users are not clearly directed to know where they should brush. Moreover, brushing is designed for selecting a group of entities, rather than an individual one, so it is hard to use this to support users to delve into detailed information of a coordinated relation. Thus, neither parallel coordinates nor scatterplot matrix, without any adaption, can effectively support visual exploration of coordinated relationship.

List (or list view) holds a similar concept to that of parallel coordinates which manages entities in lists by domains. Instead of using brushing to select a group of entities, in a list view, users usually select individual entities. ConTour [38] and Jigsaw [19] are two examples that applied a list layout to support meaningful relations discovery. The problem with Jigsaw’s List View is that its visual representation is explicit but interaction is implicit. Different from Jigsaw, ConTour suffers from the problem of implicit visual representation. ConTour highlights related entities when users hover an entity. This serves as explicit visual clues to direct users’ attention to some potential targets. However, ConTour does not show visual links between related entities. Just with color changes, it is hard for users to visually discriminate one coordinated relationship from another, especially when there are many and some of them overlap each other. ConTour applies recursive nesting to reduce such visual clutter, but it requires entity duplication, which may confuse users. BiSet is inspired by the good parts of both Jigsaw and ConTour, which provides both visual links and visual clues to guide users for coordinated relationship exploration.

Matrix is a preferred layout for exploring coordinated relationship (in *bicluster-level*) that has been well explored in bioinformatics domain (e.g., BiCAt [3], BiCluster viewer [22], BiOverlap 2.0 [43], BiGGEsts [18], BiVoc [21], Expression Profiler [31] and GAP [55]). By reordering and duplicating rows and columns, biclusters can be visually revealed in a matrix [21, 22]. However, similar to the problem of ConTour, row and column duplication in a matrix may cause more confusion than entity duplication in a list. In addition, interacting with entities in a matrix is not as easy as that in a list. Thus, compared with list, matrices are not flexible enough to support coordinated relationship exploration (e.g., drilling down to details of relations), although it can provide a visual representation for such relationship.

Node-link diagrams in the context of multivariate or heterogeneous networks can also be used to visualize coordinated relationships. For example, PivotPaths [12] applies a modified node-link diagram to support relationship exploration in heterogeneous networks. It separates a 2D space into three regions to contain entities from different domain (e.g., authors, articles and keywords). This explicitly separates nodes in a node-link diagram into different groups based on domains, which is similar to a list view. Entities in the middle region are horizontally aligned. Edges are presented to show connections between entities in two neighboring domains. By following edges, users can manually discover coordinated relationships in PivotPaths (e.g., finding three co-authors of four papers with the same two keywords). OnionGraph [44] used a similar visual representation to show bibliographic networks. Pretorius et al. also applied a similar layout to show structures of multivariate graphs and edge labels [40]. In their method, edge labels are listed in the middle region. Related entities are highlighted when users select these labels. This is a good example of enabling interactivity on edges for revealing relevant nodes. However, their method lacks the ability to manage nodes via interactions on the labels. Related nodes are highlighted but they do not move close to the selected label. Thus, users still have to navigate in a graph (e.g.,

scrolling up and down) to find highlighted nodes, if the graph is large.

Hybrid layout combines two or more layouts together. Usually it uses a list or a node-link diagram as the basic layout and replaces entities with matrices (or other types of visualizations). Matchmaker [34] and VisBrick [33] took a *list view as the major layout*, while Bixplorer [14], Furby [47] and NodeTrix [23] applied a *node-link diagram as the key layout*. The former group organizes relations in lists and the latter group uses a 2D space to manage relations. Since entities are replaced with relations, it is difficult to further explore detailed information (e.g., entities and *entity-level* relations) to interpret a coordinated relationship. This may be even harder here than using a matrix layout.

In BiSet, we choose list as the key layout. The detailed rationale for our decision is discussed in Section 3.2.

2.3 Visual Links and Edge Bundling

Visual links (e.g., edges in graphs) are important for assisting visual navigation [24] and indicate certain types of relations (e.g., causality [57]). By following links, users can navigate their foci from one part of a visualization to another, or across different visualizations [52] or applications [53]. This helps to direct users to potentially related content for comparison and evaluation [9, 46] or assist users to explore visually hidden (or being covered) content [16]. However, cases are not always optimistic. If too many edges exist, a visual layout (e.g., graph) will become a hairball of visual clutter [37]. Edge bundling is a useful technique to reduce visual clutter and reveal high level edge patterns by visually aggregating edges based on certain rules (e.g., force-directed model [26], image-based rule [50], geometry-based rule [11], etc.). However, there are two major problems with these edge bundling techniques: 1) spatial-based bundling in the visual level (losing relations in the data level), and 2) lack of interactions on edge bundles.

Traditional edge bundling techniques simply group edges based on spatial proximity (e.g., the position of nodes or edges), which may ignore some implicit relations in a dataset. Since visual adjacency is determined by layout algorithms (e.g., force-directed layout), rather than knowledge discovery algorithms (e.g., biclustering), bundling visually adjacent edges does not guarantee that a bundle of these individual relations reflect meaningful semantic insights from the dataset. To deal with such problems, a hierarchical edge bundling technique is proposed in [25] that bundles adjacent edges by considering the hierarchical relations in a dataset. However, compared with coordinated relationships, hierarchical relationships are relatively simple because they can not reveal high level semantic insights implied by the coordination of individual relationships. Despite this, the hierarchical edge bundling technique inspires our design of BiSet that bundle edges based on coordinated relationship. This potentially enables users to infer semantic insights from these edge bundles.

Deficient interaction on edge bundles is another problem with existing edge bundling techniques. Such bundles have limited capabilities to support users exploring the space of relationship, although they help to reduce visual clutter. This partially results from the previous problem since edge bundles depend on the layout of nodes. If positions of nodes change, the existing bundles may also change. This means that these edge bundles are not stable. In BiSet, we map algorithmically discovered coordinated relationships to edge bundles. This assures that each bundle visually presents a certain coordinated relationship. Thus, bundles in BiSet are more independent from the layout of nodes than existing techniques. Based on this, BiSet allows interactions on edge bundles which enables users to manipulate bundles to forage related information, and organize spatializations for synthesis [2].

3 DESIGN REQUIREMENT ANALYSIS

3.1 A Design Trade-off

Figure 2 shows that a bicluster can both bundle connections and group entities and this can be clearly conveyed in a list view. However, cases are not always that easy. When certain entities belong to more than one bicluster, it becomes difficult to visually group all entities of the same bicluster together. Figure 4 shows an example of this case. There are three biclusters indicating three different coordinated relations between people and locations. They share some entities (e.g., P1 and

We either organize the vis based on the entities or the relationships and both have their drawbacks

P2 are associated with both bicluster A and bicluster B). This brings about an Euler diagram problem [1, 42] when visually grouping entities. When the number of shared entities increases, it becomes more difficult to present a visual representation that show the membership of these entities without replicating some of them. This suggests a key design trade-off: **entity-centric versus relationship-centric**, which means that we cannot easily achieve the goal of clearly presenting both entities (without duplications) and relationships (without separations). Techniques such as bubble sets [10] and untangling Euler diagrams [41] attempt to balance this trade-off by using a 2D space. They show relationships with their members in a calculated spatial layout with certain boundaries. However, they do not scale up well. In a list, there is just one dimension to use for organizing entities, so it is even harder to balance this trade-off.

Entity-centric requires that entities that belong to a certain domain (e.g., people in Figure 4) should be listed in a certain order without duplication. The positions of entities can be reordered to fulfill some purpose (e.g., listing names in an alphabetical order or ranking them based on frequency in documents). Since entities cannot be duplicated, relationships that consist of shared entities may be separated apart. Thus, an entity-centric design can help to avoid the ambiguity caused by entity replication, but it costs the completeness of relationships.

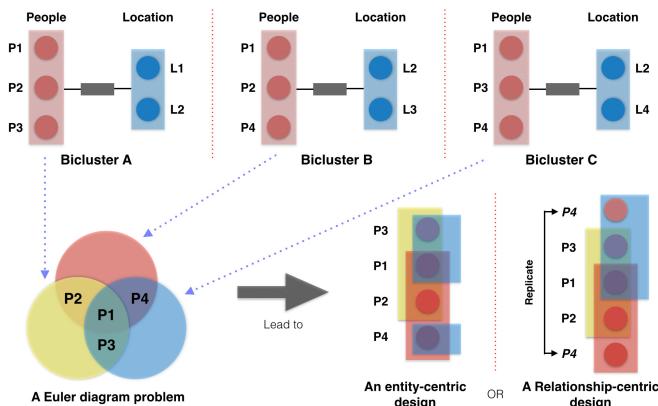


Fig. 4. A detailed example to illustrate the Euler diagram problem that arises when visually presenting the membership of entities in the domain of people shared by three biclusters. This problem indicates a key design trade-off: **entity-centric** versus **relationship-centric**.

Relationship-centric requires that entities that belong to the same relationships (e.g., biclusters) should be placed near each other, which visually preserves the completeness of relationships. To achieve this, entities may be duplicated, particularly when several relationships share two or more entities. A relationship-centric design (e.g., bubble sets technique [10]) can maintain the integrity of relationships with the cost of entity duplication. This may confuse users, especially when they see some entities appear at several different positions in a list.

3.2 Layout Candidate Selection

Three layouts can be potentially applied to show coordinated relationships: node-link diagram, matrix, and list (including parallel coordinates). A detailed discussion about applying them in bicluster visualizations are addressed in [49]. Because of the following two advantages, we choose list as the major layout of BiSet.

Consistent, domain-based entity management. In a list layout, entities are organized in a consistent manner by domains. In a node-link diagram, nodes are placed either randomly or based on certain layout algorithms (e.g., force-directed layout [15]), so it is hard to separate entities of one domain (e.g., people) from those of another (e.g., location). Compared with a node-link diagram, a matrix is a better organized layout, where relations are more readable [17]. Entities in a matrix are organized in two orthogonal directions based on domains. However, when a new group of entities are to be added, it is impossible to add them in the existing matrix. A new matrix has to be built

to show connections between the newly added entities and those from one domain of the existing matrix. This leads to two problems: entity duplication and direction (row or column) selection. To build the new matrix, entities from one domain of the existing matrix have to be duplicated to form either row names or column names in the new matrix. If replicated entities work as the row names in the new matrix, newly added ones will be the column names, and vice versa. Compared with matrices, in a list whenever a group of entities are to be added or removed, we can easily add or remove one list. Thus, compared with the other two layouts, lists organize entities in a consistent manner.

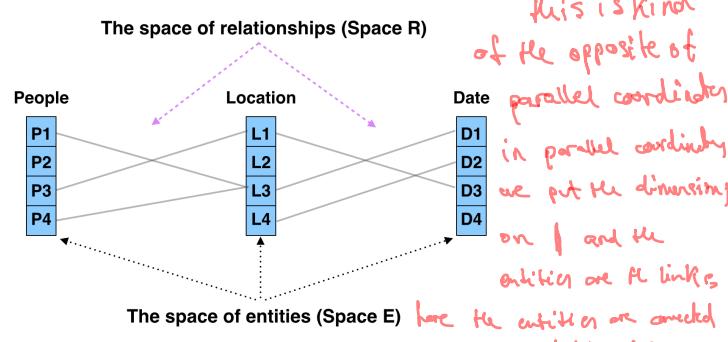


Fig. 5. A 2D space is sliced into two types of subspaces in a list view.

Organized alternate subspaces for entities and relationships. In a list view, a 2D space is sliced into two types of subspaces, where **entities and relationships appear alternately**. Figure 5 gives an example of such slicing that generates three subspaces for entities and two subspaces for relationships. This provides an opportunity to leverage the design trade-off by using the relationship subspace. In node-link diagrams, the space of entities is intertwined with that of relations so there is no clear boundary between the two spaces. Matrices potentially emphasize the space of relationships (e.g., the structure of a dataset) [17], so it is difficult to support simple relationship (e.g., *entity-level* or *group-level* relations) exploration in a matrix. Visually, in a matrix, the proportion of the space for relations (the total area of all cells in a matrix) is larger than that for entities (one column and one row in the matrix) and the ratio of the two increases when the size of a matrix gets larger. Thus, compared with the other two layouts, lists slice a 2D space for entities and relations in a clearly organized and usable way.

3.3 Requirements Description with Identified Tasks

Based on the design trade-off and the selected layout, we identify four important requirements with key tasks that BiSet needs to support.

R1: Entity and relationship encodings. Efficient visual encodings for both entity and relationship are necessary which should attempt to achieve four important goals. First, visual encodings should assist users to **discriminate entities from relations** (a). Then visual encodings (particularly for relationships) should potentially help to **reduce visual clutter** (b). Third, visual encodings should provide clues to **reveal the membership of entities** (c). Finally, visual encodings should **reflect the changes** when the state of entities or relations updates (d).

R2: Four types of exploration. There are four possible types of exploration in a list layout based on the space slicing: *i) from entity to entity*, *ii) from entity to relationship*, *iii) from relationship to relationship*, and *iv) from relationship to entity*. The first type of exploration refers to when users start from entities and focus on **finding related entities**. The second one may happen in the scenario where users take the strategy of **following the clues from identified meaningful entities** [30] and search for more relevant information. The third type of exploration may be performed when **users seek additional relationships based on current one**. This is a possible case for text analytics which has been reported in [48]. The last one **may be used to compare several hypotheses**, which has been identified as a common **intelligence analysis strategy** [7]. For example, several biclusters share some individual relations, but they still have their unique ones. This may form (partially) conflicted relations that lead to different hypotheses. In this

why lists are better than node diagrams and matrices

tasks and way of performing them.

Table 1. A Summary of Identified Tasks in the Four Types of Exploration.

Types of Exploration	Task Description		
	From One to Many ($1:m$ or $1:m:n; \dots :z$)		From Many to Many ($m:n$ or $m:n; \dots :z$)
From Entity to Entity	T1: given one entity in an entity-list, find entities in the same list or multiple entity-lists that are related to the given entity.		T2: given multiple entities in one entity-list or multiple entity-lists, find entities in neighboring entity-list(s) or multiple entity-lists that are related to any or (possibly) all of the given entities.
From Entity to Relationship	T3: given one entity in an entity-list, find biclusters in neighboring relationship-list(s) or multiple relationship-lists that are directly or indirectly (e.g., via bicluster-chains) associated with the given entity.		T4: given multiple entities in one entity-list or multiple entity-lists, find biclusters in neighboring relationship-list(s) or multiple relationship-lists that are directly or indirectly (e.g., via bicluster-chains) associated with any or (possibly) all of the given entities.
From Relationship to Relationship	T5: given one bicluster in a relationship-list, find biclusters in the same list or multiple relationship-lists that are associated with the given bicluster via shared entities.		T6: given multiple biclusters in one or multiple relationship-list(s), find biclusters in multiple relationship-lists that are associated with any or (possibly) all of the given ones via shared entities.
From Relationship to Entity	T7: given one bicluster in a relationship-list, find entities in neighboring entity-lists that are directly related to the given bicluster.		T8: given multiple biclusters in one relationship-list or multiple relationship-lists, find entities in neighboring entity-lists or multiple entity-lists that are directly or indirectly (e.g., via bicluster-chains) associated with any or (possibly) all of the given biclusters.

case, users may have to compare these biclusters by drilling down to detailed level of information (e.g., entities) for competing them.

There are two ways to perform the four types of exploration: *from one to many* and *from many to many*. For instance, users may want to find related entities based on one or multiple biclusters. Based on this, we identify eight user tasks involved in such explorations, which are summarized in Table 1. For each type of exploration, users can start from either an entity (or a bicluster) or multiple entities (or biclusters) and then try to find relevant entities or biclusters. A user's analytical process may consist of a series of these tasks that iteratively forage relevant information and identify some meaningful pieces [30]. Detailed examples of these tasks are addressed and labeled in Section 5.

R3: Organizing entities and relationships. Entities and relationships should be visually represented in an organized way. This can help users to easily find useful information. In addition, users may want to make changes to the automatically generated layouts so that they can organize entities or relationships in personalized, meaningful ways (e.g., using spatialization) for sensemaking [2].

R4: Retrieving original data for reference. To evaluate algorithmically discovered coordinated relationships, users may refer to the content from the original dataset (e.g., documents) because they need contextual information to help them to interpret and further evaluate these relations [19]. BiSet should attempt to efficiently direct users to useful information, rather than keep them from reading documents.

4 BiSet TECHNIQUE

Three key aspects are involved in BiSet: coordinated relationship discovery in data level; *bundles as objects* and a “in-between” layer in visual level; and interactions to support four types of exploration and two ways of organizing information. In this section, we discuss them in detail and explain how identified design requirements are satisfied.

4.1 Data Level: Bicluster Discovery

Coordinated relationship discovery is the fundamental step in BiSet since it determines how edges are bundled. In BiSet, we formalize coordinated relationships as biclusters. Suppose that entities have been extracted from a dataset (e.g., documents) with named entity recognizers such as LingPipe [5] or similar tools. We use closed itemset algorithms (e.g., LCM [51] and CHARM [56]) to discover biclusters based on extracted entities. Each unique pair of entity types (e.g., people and location, people and date, location and date, etc.) is considered a type of coordinated relationship and is computed separately to generate results that include all unique pairs of entity types. Results are stored in a database and associated with the dataset under investigation.

The mined biclusters indicate different coordinated relationships and some of them may share entities and relations in *entity-level* or *group-level*. This suggests that some entities and edges (individual relationships) are members of biclusters. Thus, membership in BiSet, in the data level, can be considered from two aspects: entity and edge.

4.2 Visual Level: Bundles as Objects and “In-between”

In BiSet, we propose two important concepts to balance the key design trade-off: *making bundles as first class objects* and *adding a new layer “in-between” lists to contain bundle objects*. The former enables users to directly manipulate relationships (*relationship-centric*) and the latter helps to visually reveal membership of entities in two neighboring lists without duplicating entities (*entity-centric*).

Making Bundles as Objects In BiSet, we make bundles the first class objects so users can directly manipulate them for sensemaking purposes (e.g., organizing information). BiSet bundles edges based on computed biclusters that reflect algorithmically discovered coordinated relationships. Different from spatial edge bundling techniques that emphasize bundling based on spatial proximity, BiSet bundles edges based on coordinated relationships that reveal task-oriented semantic insights. This assures that edge bundles remain stable, regardless of the positions of associated entities. Thus, bundles potentially enables users to use space (e.g., vertical position) to organize information (e.g., entities) (for R3), and safely retrieve related information by interacting with edge bundles (for R2, iii and iv).

Adding an “in-between” Layer To make bundles usable, we add a new layer, called “in-between”, visually locating in the space between two neighboring entity-lists. It contains bundles and edges (e.g., those that do not belong to any coordinated relationship). In this layer, BiSet allows users to manipulate bundles for sensemaking (e.g., organizing entities and checking their membership), so bundles can support users interactively exploring coordinated relationships.

4.2.1 Semantic Edge Bundling in BiSet

BiSet has two types of edges: *independent* and *associated*, which are mutually exclusive. The former refers to edges that do not belong to any coordinated relationship and the latter are those that can form one or more coordinated relationships. For instance, in Figure 6, the edge on top in (A) is an independent edge and other edges are associated ones. Independent edges can reflect *entity-level* and *group-level* relationships, but they are not associated with others to form coordinated relationship. Based on membership, associated edges that belong to the same bicluster can be aggregated and represented as an edge bundle. BiSet takes the following three steps to bundle edges (for R1(b)).

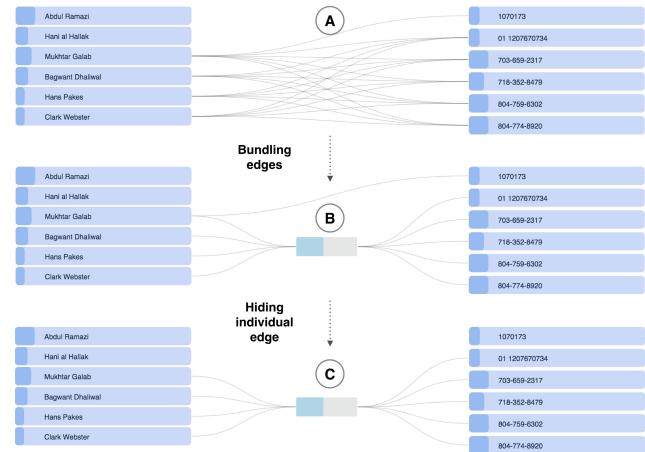


Fig. 6. Three modes in the “in-between” layer for displaying edges. (A) is the *edge only mode* that shows all edges between related entities. (B) is the *hybrid mode*, which presents bundles with individual edges. (C) is the *bundle only mode* that just displays bundles.

1) *Grouping edges based on membership.* We separate associated edges into different groups based on their associated biclusters. For

those in multiple biclusters, we duplicate and assign them respectively to multiple groups, so each group has a complete number of edges.

2) *Bundling edges based on groups*. For each group obtained from the previous step, we bundle all its edges together and visually replace these edges with a rectangle to indicate an edge bundle.

3) *Connecting bundles with entities*. We link bundles and entities based on membership. This assures that entities and their associated bundles are fully connected (for R1(c)).

This bicluster-based edge bundling can potentially reduce visual clutter and clearly present a coordinated relationship (for R1(b)). As is shown in Figure 6, compared with (A), (B) clearly illustrates the coordinated relationship between four people and five phone numbers.

In fact, BiSet supports three modes to show edges: edge only mode (*EM*), hybrid mode (*HM*) and bundle only mode (*BM*), shown as (A), (B) and (C) respectively in Figure 6. In *EM*, BiSet shows edges without bundling. In *HM*, BiSet shows independent edges and bundles. In *BM*, BiSet just displays bundles. The three modes attempt to meet different user needs. For example, *EM* potentially reveals the overview of relationships between two entity sets (e.g., (A) in Figure 6). *BM* enables users to focus on analysis just with coordinated relationships. *HM* can help to visually organize groups of individual relations into multiple levels (e.g., coordinated bundles with individual entity-level relationships). In BiSet, users can switch modes during their analysis. An example of using semantic edge bundling in BiSet is shown in Figure 7, which reduces 164 edges to 9 bundles. In this example, we use LCM to calculate biclusters and set the *minimum support* parameter to three, which assures that each calculated bicluster has at least three entities in one of the two related domains (here is the people's name).

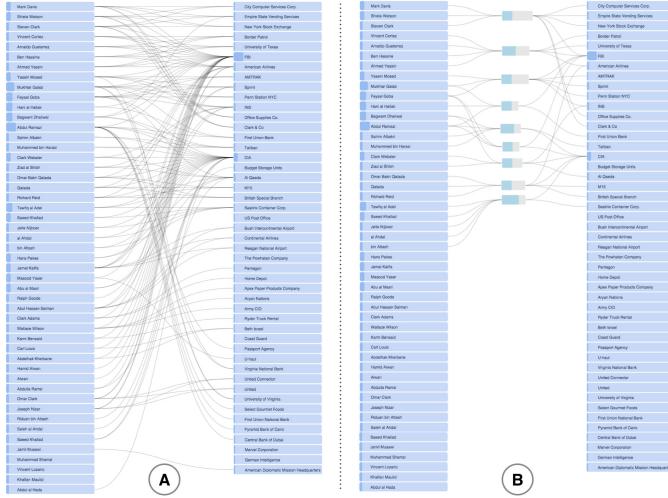


Fig. 7. A semantic edge bundling example in BiSet. (A) shows the original 164 edges. (B) After semantic edge bundling, there are 9 bundles.

In addition to improving readability, bundles in BiSet preserve the coordinated attribute of entities and edges. This enables users to infer semantic meanings about potentially coordinated activities. For example, why are the four people all related with the five phone numbers in Figure 6? Perhaps they colluded about a terrorist assault. Such semantic insights cannot be easily revealed from separated entity-level or group-level of relations. Thus, edge bundles in BiSet serve two important roles: *improving readability* and *revealing semantic insights*.

4.2.2 Visual Encoding in BiSet

BiSet uses four major visual channels [37] to encode *bundles*, *entities* and *edges*: **shape**, **size**, **color** and **position**. Figure 8 shows a detailed example of visual encodings in BiSet.

Shape and Size In BiSet, entities and bundles are represented as rectangles (e.g., (1) and (2) in Figure 8). Edges are visualized as Bézier curves. We choose Bézier curves since they can generate more smooth edges, compared with polylines [32].

Length, width and font size are three specific types of size channel used in BiSet. Rectangles indicating entities are equal in length, while those representing bundles are not. BiSet applies a linear mapping function to determine the length of a bundle based on the total number of its related entities. In a bundle, BiSet uses two colored regions (light green and light gray) to indicate the proportion between its related entities in the left list and those in the right list. In an entity rectangle, a small rectangle is presented on the left to indicate its frequency in a dataset. The length of these small rectangles is determined by the frequency of the associated entities. Based on these with different color encoding and position, users can easily discriminate entities from bundles (for R1(a)). In addition, the width of edges can reflect results of user selections. For instance, in Figure 8, compared with the width of those in (3), the width of edges in (8) is larger, since two relevant entities are selected. Moreover, when hovering an selected entity or bundle, related entities will be displayed in larger fonts. This helps users to review relevant information of previous selections (for R1(c)).

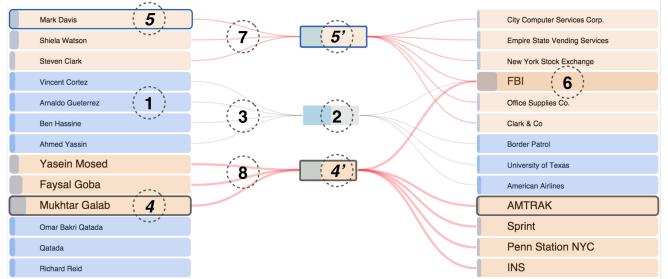


Fig. 8. Visual encodings in BiSet. (1), (2) and (3) present the normal state of an entity, a bundle and edges, respectively. (4) and (4') show the selected state of an entity and a bundle with accumulated highlighting. (5) and (5') present the mouseover state of an entity and a bundle. (6) shows accumulated highlighting of an entity. (7) presents the highlighting state of edges. (8) shows the accumulated highlighting of edges.

Color Coding BiSet applies color coding to entities, bundles and edges to indicate their states. In BiSet, entities and bundles have three different states (normal, mouseover and selected), and edges has two different states (normal and highlighting). In Figure 8, (1), (2) and (3) respectively present the normal state of an entity, a bundle and edges; (4) and (4') show the selected state of an entity and a bundle; (5) and (5') illustrate the mouseover state of an entity and a bundle; and (7) and (8) demonstrate the highlighting state of edges. In addition, two different colored borders (blue and black) are used to help users further discriminate the mouseover state from the selected state (for R1(d)). When hovering an entity or a bundle, a blue border will be added to the rectangle. This border will change to black after user selection, which indicates that the state has changed from mouseover to selected.

Accumulated highlighting is important in BiSet, which is triggered by mouseover and selection. Different from simple highlighting, accumulated highlighting provides useful visual clues (e.g., darker in orange) for shared entities (for R1(c)) and bundles. BiSet applies accumulated highlighting to entities, bundles and edges by increasing the shading of their colors. For example, in Figure 8, the entity in (6) is in darker orange, compared with those in (4) and (5), since its highlighting is accumulated based on selections of the entity in (4) and AMTRAK, and the mouseover on (5).

Position Position is used to organize entities and bundles in BiSet. A set of entities of a certain domain (e.g., people) is organized as an entity-list. In between two neighboring entity-lists (the “in-between” layer), there is one relationship-list that contains coordinated relationships as biclusters (visually as edge bundles).

In entity-lists, the positions of entities can be determined in three ways (for R3): in an *alphabetical order*, based on *frequency*, or based on the *order of (one-side) associated bundles*. Alphabetical and frequency-based ordering can help to organize entities. However, they may lead to a severe problem of membership separation, since entities

organize

based

on ↑

definitive

belonging to the same bicluster may not be listed close to each other. This results from the trade-off discussed in Section 3.1. To balance this trade-off, BiSet provides the third approach to organize the position of entities based on the order of (one-side) associated bundles. For example, entities in the left list in Figure 8 are ordered based on their associated biclusters in the middle list.

The larger a bicluster’s size is, the more important it is likely to be. A bicluster in larger size contains more information, so it is more likely to reveal potentially meaningful coordinated relationships. With this rationale, we apply a greedy algorithm, listed below, to organize entities based on the size of their associated biclusters.

Algorithm 1: Get positions of entities associated with biclusters

```

input : curPos, initialized as 0 for the top position
       orderedBics, a list of ordered biclusters
       bicEntDict, a dictionary stores entities for each bicluster
       entBicDict, a dictionary stores biclusters for each entity
       rankedEntSet, a set stores entities already ranked

1 foreach bic in orderedBics do
2   entList = bicEntDict(bic);
3   foreach entity in entList do
4     if not entity in rankedEntSet then
5       bicList = entBicDict(entity);
6       totalRank = 0, num = 0;
7       foreach bicluster in bicList do
8         totalRank += bicluster.rank;
9         num += 1;
10      end
11      entity.rank = totalRank / num;
12      rankedEntSet.add(entity);
13    end
14  end
15 end
16 orderedEnts = entList.sort(sel.rank);
17 foreach entity in orderedEnts do
18   entity.pos = curPos++;
19 end
```

BiSet also allows automatically changing positions of groups of entities (for R3) by dragging a bundle in the “in-between” layer. Figure 9 shows such an example. After dragging a bundle, two groups of entities in its two neighboring lists automatically move to new positions.

In the “in-between” layer, BiSet supports two ways to organize positions of bundles: automatically adjusting positions based on related entities, and manually dragging and moving bundles to new positions. The former allows listing bundles based on the positions of entities in either or both neighboring list(s), while the latter enables users to adjust the automatically generated layout based on their ad hoc needs (e.g., synthesis with created spatializations).

4.3 Interaction: Exploring and Organizing Information

Revealing and organizing information in a bidirectional manner serves as a key design principle for interactions in BiSet. By enabling users to directly interact with entities in entity-lists, and bundles in the “in-between” layer, BiSet can potentially support four types of exploration and two ways of organizing information.

Revealing Information BiSet supports bidirectional information revealing. Specifically, users can find relevant bundles by selecting or hovering over entities, and vice versa (for R2, ii and iv). Relevant entities or bundles in entity-lists or “in-between” layer will also be highlighted in BiSet (for R2, i and iii), when users interact with an entity or a bundle. This means that when users select (or hover over) entities, BiSet can use accumulated highlighting to reveal four types of potentially relevant information: 1) entities in the same list that belong to the same bicluster(s) with the selected entities, 2) entities in other list(s) that are related with the selected entities, 3) bundles in neighboring “in-between” layer that are directly connected with the

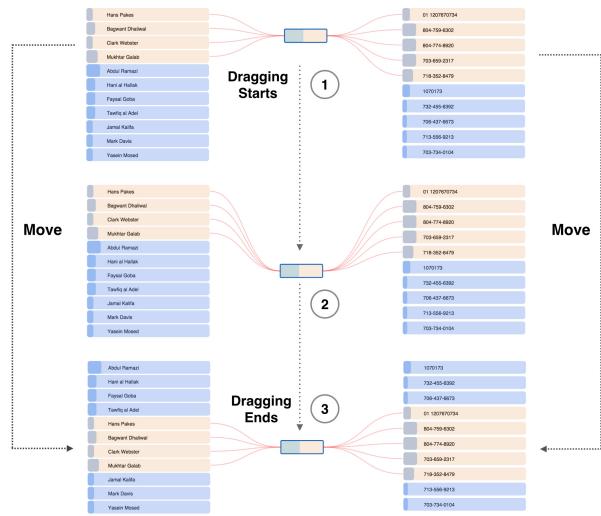


Fig. 9. Dragging a bundle in the “in-between” layer. Entities associated with this bundle automatically move to their new positions.

selected entities, and 4) bundles in other relationship-list(s) that are associated with the selected entities via bicluster-chain(s). 1) and 2) satisfy the requirement of T1 and T2 (from entity to entity), and 3) and 4) support T3 and T4 (from entity to relationship). Similarly, when users select (or hover over) bundles, BiSet can reveal the same four types of information, but they are related with selected bundles. In this case, 1) and 2) fulfill the requirement of T7 and T8 (from relationship to entity), and 3) and 4) can support T5 and T6 (from relationship to relationship). For example, in Figure 8 when users hover the entity in (5), three entities in the left list and six entities in the right list are highlighted. Of the six entities, the FBI one is in darker orange, which indicates that it is shared with another bundle (bicluster) on the bottom. The bundle on top is also highlighted, since it is directly related to the entity. With such ways of revealing information, BiSet can support the four types of exploration.



Fig. 10. The document view from bundles in BiSet. (A) shows bicluster ID, related document and associated entities. (B) shows the content of a document. (C) lists all document ID(s) in the data with a search function.

Organizing Information BiSet supports two ways of organizing information, which is also bidirectional. Users can organize the position of entities based on bundles, and vice versa. BiSet uses vertical positions to visually externalize the organized entities and bundles. As discussed above, for bundles in the “in-between” layer, BiSet can not only automatically organize them based on the position of related entities, but also enable users to manually adjust their positions by dragging and moving them. For entities, BiSet provides three options to automatically order them in an entity-list. When users drag and move a bundle, associated entities in two neighboring lists move with it, as is shown in Figure 9. This enables users to manually adjust positions of a group of related entities by using bundles. Thus, in BiSet, entities and bundles can mutually impact each other, which provides a flexible way for users to organize information in lists.

BiSet also allows users to review documents directly from bundles and entities with a right click menu (for R4), shown in Figure 10. When finding an interesting bundle or entity, users can use a right click menu to open a popup view where relevant documents are listed. This view is on top of the view for relationship exploration. After reading, users can quickly return to previous view by closing it.

5 USAGE SCENARIO

In this section, we walk through a text analytics scenario to demonstrate how BiSet supports an analyst to identify a coordinated activity. We use *The Sign of the Crescent dataset* [27] which contains 41 fictional intelligence reports regarding a coordinated terrorist plot in three US cities, and each plot involves a group of (at least four) suspicious people. In fact, 24 of them are relevant to the plot. We used LCM [51] to generate biclusters from the dataset with the *minimum support* parameter set to 3, which assured that each bicluster has at least three entities from one domain. This resulted in 284 unique entities, 495 relationships, and 337 biclusters (including 122 thin biclusters).

Suppose that Sarah is an intelligence analyst. She is assigned a task to read intelligence reports and identify potential terrorist threats with key persons. She opens BiSet, selects four identified domains (people, location, phone number and date) and starts her analysis. Figure 11 shows the key steps of Sarah's analytical process.

Rapid discovery of insights from coordinated relationships with bidirectional exploration. Sarah begins analysis by hovering individual entities in the list of people's names. BiSet highlights related bundles and entities, each time when she hovers the mouse over an entity (T1 and T3). Immediately she finds that *A. Ramazi* is active in three bundles, which indicates that this person is involved in three coordinated activities (T3). Sarah selects it (Figure 11 (1)) to focus on highlighted entities of the three bundles (T8). She finds that *A. Ramazi* is involved in two cells with five other people. One is in Germany and the other is more broad including four countries. *A. Ramazi* is the only person connecting the two cells, and there are two overlapped subgroups of people involved in the broader cell. Moreover, each subgroup has its unique person (*B. Dhaliwal* and *F. Goba*) (T7). BiSet quickly directs Sarah to such insights with just one click. Without BiSet, Sarah has to select and deselect many entities to find these meaningful semantic insights, particularly with tools that only display simple relationships (e.g., *entity-level* or *group-level*), such as Jigsaw.

Easily gaining insights from bundles for exploration in entity-space. Then Sarah decides to explore the two overlapped subgroups, because she wants to know what brings the unique people to them. She checks *B. Dhaliwal* first by hovering the mouse over it. After this, two bundles are highlighted (T3). By following edges from them, Sarah finds that they share two people's name and three locations, but the bigger one (shown in Figure 11 (2)) is related to a new name (*H. Pakes*) (T8). Then she examines *F. Goba* in the same way. This time three bundles and three names are highlighted (T1 and T3), and one name (*M. Galab*) has a high frequency. This quickly catches Sarah's attention, so she decides to temporarily pause the analytical branch of *B. Dhaliwal*, and moves on with the branch of *F. Goba*. Sarah hovers the mouse over *M. Galab* to check whether it leads to more information (T1 and T3). However, it turns out that no additional bundles or names are highlighted. Sarah realizes that people potentially related with *M. Galab* have already been highlighted in her current workspace. The bundle (shown in Figure 11 (3) as the black dot box in the middle) clearly reveals the people related with *M. Galab*, and their activities are all in the US (T7). With this bundle, Sarah easily acquires this key insight revealed by a group of locations. The relations revealed in this bundle are important, and Sarah infers that the three people (*M. Galab*, *Y. Mosed* and *Z. al Shihb*) may work on something together in the US. Thus, she decides to find more relevant information by following this tail [30] (T2, T4 and T7). Without bundles, Sarah has to mentally aggregate such pieces of information (18 edges crossed with each other) to gain this insight (e.g., in Jigsaw's List View).

Visually connecting semantic insights using bundles for exploration in relationship-space. Sarah selects the same bundle. BiSet highlights relevant bundles that potentially form bicluster chains with

the selected one (T5). She finds that five bundles, in the space between the location list and the phone number list, are highlighted, and two bundles, in the space between the phone number list and the date list, are highlighted. Relevant entities in lists are also highlighted (T7).

In the two lists of newly highlighted bundles, Sarah finds that there are two big ones (relatively longer in width shown in Figure 11 (4)) in each list. These two bundles seem useful since they contain more relations. Sarah wants to investigate these first and tries to check how bundles from different relationship lists are connected (T6). For bundles between the location list and the phone number list (from top to bottom), Sarah finds that the first bundle and the third one share two locations (*Charlottesville* and *Virginia*) with the selected bundle, and other highlighted bundles just share one location with the selected one (T6). Compared with the third bundle, the first one is related with more locations that are not associated with the selected bundle (T7). Sarah wants to focus on information highly connected with the selected bundle, rather than more additional information. Thus, she considers the third bundle, in this case, a useful one. Using the same strategy in another relationship-list, she finds that the bigger bundle is more useful.

After this, Sarah hides edges of other bundles with the right click menu to keep a clear view. Then her workspace shows that three bundles connected to each other through two shared locations and three shared phone numbers. Sarah feels that she has found a good number of relations, connecting four groups of entities, which may reflect a suspicious activity. Therefore, she decides to read relevant documents to find details of such connections and generate her hypothesis.

Efficiently directing to relevant documents based on bundle objects. The three connected bundles direct Sarah to eight reports, which are all relevant to the plot. Sarah quickly goes through these reports by referring to the entities with bright shading in the four connected groups (shown in Figure 11 (4)). The darker shading of an entity indicates that it is shared more times. Sarah uses this to help keep her attention to more important entities in reports. After reading the reports, she identifies a potential threat with four key persons.

Aiding hypothesis generation. By visually following the bicluster-chain (linked bundles, shown in Figure 11 (4)), Sarah quickly remembers what she has already read [2]. Finally she makes a hypothesis of the identified attack as follows:

F. Goba, M. Galab and Y. Mosed, following the commands from A. Ramazi, plan to attack AMTRAK Train 19 at 9:00 am on April 30.

6 CONCLUSION AND FUTURE WORK

We present BiSet, a visual analytics technique, which bundles edges based on biclusters to reveal task-oriented semantic insights about potentially coordinated activities for sensemaking. In BiSet, we make *edge bundles the first class objects* that enable users to directly manipulate relationships. In addition, we add a new layer, “*in-between*”, *containing bundles*, which assists to visually reveal membership of entities in neighboring entity-lists without entity duplication.

By applying interactions on bundles and entities, BiSet enables bidirectional information foraging to support exploration between the entity-space and the relationship-space. Moreover, BiSet allows organizing entities in lists, either automatically based on the positions of associated bundles or manually by dragging and moving bundles. With a usage scenario, we demonstrate how BiSet can potentially support an analyst to explore coordinated activities. However, there are still three challenges that need further explorations.

C1: Seriation in lists. In BiSet, we apply a greedy approach to order entities in lists based on biclusters in neighboring relationship-list(s). Compared with traditional ordering (in an alphabetical order or by frequency), this attempts to keep entities, belonging to the same biclusters (especially those in a larger size), close to each other. However, this approach may lead to a membership separation problem for smaller sized biclusters (those with a small number of entities), especially if some of their entities are shared by bigger sized biclusters. In fact, this brings about an ordering problem in lists when considering the design trade-off discussed before. Seriation methods, commonly used in matrices [35], can reveal clustering structure by permuting the presentation order [8]. However, applying a seriation method to order-

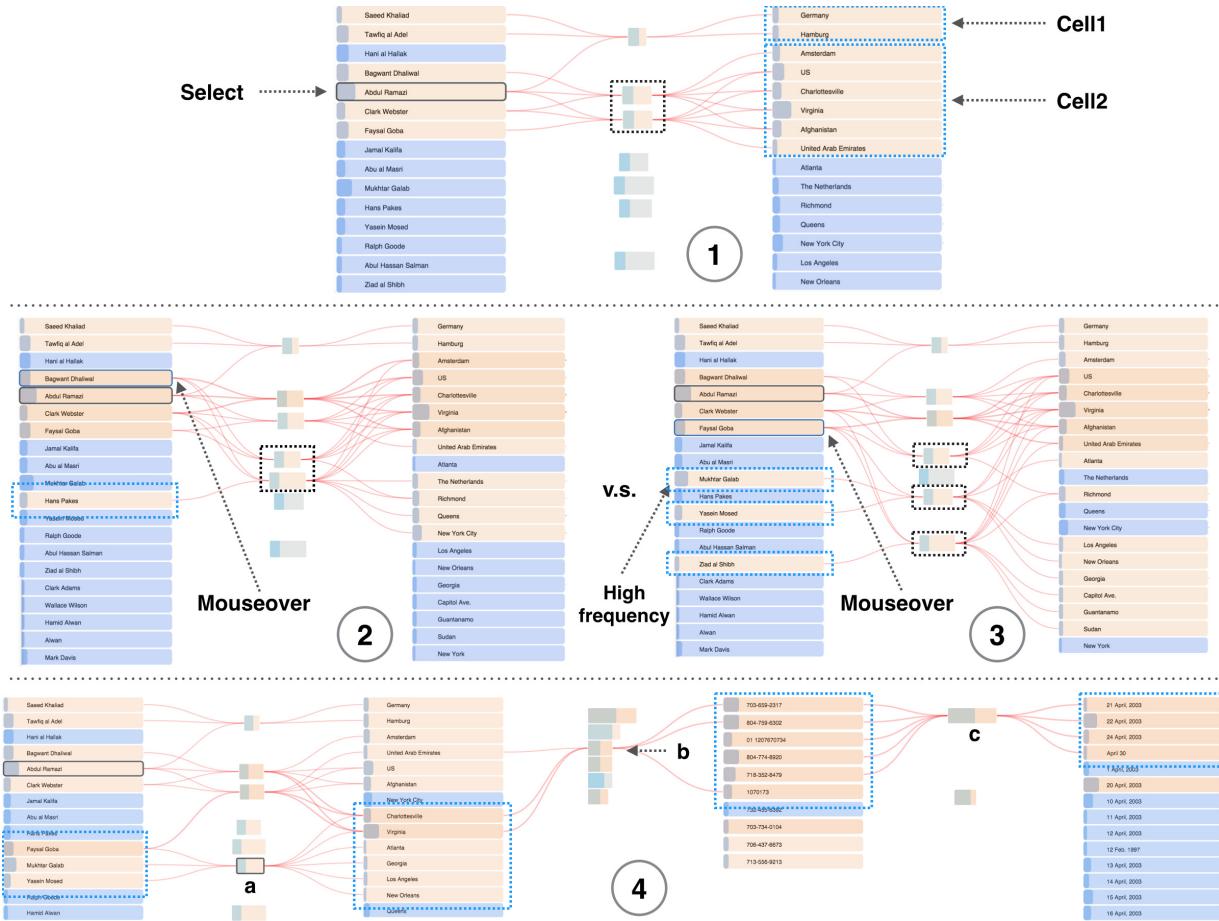


Fig. 11. A process of finding a major threat plot with key steps. (1): Based on A. Ramazi, finding that there are two similar bundles and two cells. (2): One name and two bundles are highlighted when hovering the mouse over B. Dhaliwal. (3): Three names and three bundles are highlighted when exploring F. Goba. (4) Referring to the four connected groups of useful entities for hypothesis generation.

ing entities in lists, based on coordinated relationships, is challenging. Seriation in lists based on biclusters may better organize entities and reduce edge crossings between shared entities and edge bundles.

C2: Variant visualizations in lists. BiSet chooses list as its major layout to organize entities and applies interactions to bundles to enhance its capability to utilize the relationship-space. However, organizing entities in a list may not always be the best choice. For example, entities of locations (e.g., Chicago, Boston, Seattle, etc.) may better reveal meaningful, contextual information, if visualized in a map, rather than simply listed as vertically piled bars. Similar to the technique proposed in [20], it is possible to substitute some entity-lists with variant types of visualizations in BiSet, which may help users to infer contextual clues. If such substitution could be switched on and off, users would infer semantic insights from lists, and gain contextual clues from variant views (e.g., maps).

C3: Semantic Interactions in lists. Semantic interaction provides an efficient way to facilitate visual analytics by enabling users to implicitly steer machine learning algorithms to support the human reasoning process [13]. This allows users to modify algorithmic outcomes for sensemaking purposes. Currently BiSet does not support this. Semantic interactions on bundles for organizing entities, such as splitting and merging bundles (similar to those in NodeTrix [23]), can help users to adjust algorithmically discovered biclusters and efficiently retrieve relevant information. In addition, merging bundles also provides a good opportunity for BiSet to scale up to a large dataset.

ACKNOWLEDGMENTS

The authors wish to thank Kurt Luther for his feedback on the initial version of BiSet, and Junyang Chen and Bin He for supporting the

backend development of BiSet. This research was supported in part by a grant from L-3 Communications and NSF grant IIS-1447416.

REFERENCES

- [1] 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-STARs*. Eurographics Association, 2014.
- [2] C. Andrews, A. Endert, and C. North. Space to think: large high-resolution displays for sensemaking. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 55–64. ACM, 2010.
- [3] S. Barkow, S. Bleuler, A. Prelić, P. Zimmermann, and E. Zitzler. Bicat: a biclustering analysis toolbox. *Bioinformatics*, 22(10):1282–1283, 2006.
- [4] R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.
- [5] B. Carpenter. Phrasal queries with lingpipe and lucene: ad hoc genomics text retrieval. In *TREC*, 2004.
- [6] D. B. Carr, R. J. Littlefield, W. Nicholson, and J. Littlefield. Scatterplot matrix techniques for large n. *Journal of the American Statistical Association*, 82(398):424–436, 1987.
- [7] G. Chin Jr, O. A. Kuchar, and K. E. Wolf. Exploring the analytical processes of intelligence analysts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 11–20. ACM, 2009.
- [8] J. Chuang, C. D. Manning, and J. Heer. Termite: Visualization techniques for assessing textual topic models. In *Proceedings of International Working Conference on Advanced Visual Interfaces*, pages 74–77. ACM, 2012.
- [9] C. Collins and S. Carpendale. VisLink: Revealing Relationships Amongst Visualizations. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1192–1199, 2007.
- [10] C. Collins, G. Penn, and S. Carpendale. Bubble sets: Revealing set relations with isocontours over existing visualizations. *Visualization and*

- Computer Graphics, IEEE Transactions on*, 15(6):1009–1016, 2009.
- [11] W. Cui, H. Zhou, H. Qu, P. C. Wong, and X. Li. Geometry-based edge clustering for graph visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1277–1284, 2008.
- [12] M. Dörk, N. H. Riche, G. Ramos, and S. Dumais. Pivotpaths: Strolling through faceted information spaces. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2709–2718, 2012.
- [13] A. Endert, P. Fiaux, and C. North. Semantic interaction for sensemaking: inferring analytical reasoning for model steering. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2879–2888, 2012.
- [14] P. Fiaux, M. Sun, L. Bradel, C. North, N. Ramakrishnan, and A. Endert. Bixplorer: Visual analytics with biclusters. *Computer*, 46(8):90–94, 2013.
- [15] T. M. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991.
- [16] T. Geymayer, M. Steinberger, A. Lex, M. Streit, and D. Schmalstieg. Show me the invisible: visualizing hidden content. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 3705–3714. ACM, 2014.
- [17] M. Ghoniem, J. Fekete, and P. Castagliola. A comparison of the readability of graphs using node-link and matrix-based representations. In *Information Visualization, IEEE Symposium on*, pages 17–24. IEEE, 2004.
- [18] 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*, 2(1):124, 2009.
- [19] C. Gorg, Z. Liu, J. Kihm, J. Choo, H. Park, and J. Stasko. Combining computational analyses and interactive visualization for document exploration and sensemaking in jigsaw. *Visualization and Computer Graphics, IEEE Transactions on*, 19(10):1646–1663, 2013.
- [20] S. Gratzl, N. Gehlenborg, A. Lex, H. Pfister, and M. Streit. Domino: Extracting, Comparing, and Manipulating Subsets Across Multiple Tabular Datasets. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):2023–2032, 2014.
- [21] G. A. Grothaus, A. Mufti, and T. Murali. Automatic layout and visualization of biclusters. *Algorithms for Molecular Biology*, 1(1):15, 2006.
- [22] J. Heinrich, R. Seifert, M. Burch, and D. Weiskopf. Bicluster viewer: a visualization tool for analyzing gene expression data. In *Advances in Visual Computing*, pages 641–652. Springer, 2011.
- [23] N. Henry, J. Fekete, and M. J. McGuffin. Nodetrix: a hybrid visualization of social networks. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1302–1309, 2007.
- [24] I. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *Visualization and Computer Graphics, IEEE Transactions on*, 6(1):24–43, 2000.
- [25] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):741–748, 2006.
- [26] D. Holten and J. J. Van Wijk. Force-directed edge bundling for graph visualization. In *Computer Graphics Forum*, volume 28, pages 983–990. Wiley Online Library, 2009.
- [27] 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. *Washington, DC: Joint Military Intelligence College*, 2003.
- [28] A. Inselberg and B. Dimsdale. Parallel coordinates. In *Human-Machine Interactive Systems*, pages 199–233. Springer, 1991.
- [29] Y. Jin, T. M. Murali, and N. Ramakrishnan. Compositional mining of multirelational biological datasets. *ACM Transactions on Knowledge Discovery from Data*, 2(1):1–35, Mar. 2008.
- [30] Y.-a. Kang, C. Gorg, and J. Stasko. Evaluating visual analytics systems for investigative analysis: Deriving design principles from a case study. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, pages 139–146. IEEE, 2009.
- [31] M. Kapushesky, P. Kemmeren, A. C. Culhane, S. Durinck, J. Ihmels, C. Körner, M. Kull, A. Torrente, U. Sarkans, J. Vilo, et al. Expression profiler: next generation—an online platform for analysis of microarray data. *Nucleic acids research*, 32(suppl 2):W465–W470, 2004.
- [32] A. Lambert, R. Bourqui, and D. Auber. Winding Roads: Routing edges into bundles. *Computer Graphics Forum*, 29(3):853–862, Aug. 2010.
- [33] A. Lex, H. Schulz, M. Streit, C. Partl, and D. Schmalstieg. VisBricks: Multiform Visualization of Large, Inhomogeneous Data. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2291–2300, 2011.
- [34] A. Lex, M. Streit, C. Partl, K. Kashofer, and D. Schmalstieg. Comparative Analysis of Multidimensional, Quantitative Data. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1027–1035, 2010.
- [35] I. Liiv. Seriation and matrix reordering methods: An historical overview. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 3(2):70–91, 2010.
- [36] S. C. Madeira and A. L. Oliveira. Bioclustering algorithms for biological data analysis: a survey. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 1(1):24–45, 2004.
- [37] T. Munzner. *Visualization Analysis and Design*. CRC Press, 2014.
- [38] C. Partl, A. Lex, M. Streit, H. Strobel, A. M. Wassermann, H. Pfister, and D. Schmalstieg. ConTour: Data-Driven Exploration of Multi-Relational Datasets for Drug Discovery. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):1883–1892, 2014.
- [39] P. Pirolli and S. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of international conference on intelligence analysis*, volume 5, pages 2–4, 2005.
- [40] A. J. Pretorius and J. J. Van Wijk. Visual inspection of multivariate graphs. In *Computer Graphics Forum*, volume 27, pages 967–974. Wiley Online Library, 2008.
- [41] N. H. Riche and T. Dwyer. Untangling euler diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1090–1099, 2010.
- [42] P. Rodgers. A survey of euler diagrams. *Journal of Visual Languages & Computing*, 25(3):134–155, 2014.
- [43] R. Santamaría, R. Therón, and L. Quintales. Bicoverlapper 2.0: visual analysis for gene expression. *Bioinformatics*, page btu120, 2014.
- [44] L. Shi, Q. Liao, H. Tong, Y. Hu, Y. Zhao, and C. Lin. Hierarchical focus+ context heterogeneous network visualization. In *Pacific Visualization Symposium (PacificVis), 2014 IEEE*, pages 89–96. IEEE, 2014.
- [45] H. Siirtola and K.-J. Räihä. Interacting with parallel coordinates. *Interacting with Computers*, 18(6):1278–1309, 2006.
- [46] M. Steinberger, M. Waldner, M. Streit, A. Lex, and D. Schmalstieg. Context-Preserving Visual Links. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2249–2258, 2011.
- [47] M. Streit, S. Gratzl, M. Gillhofer, A. Mayr, A. Mitterecker, and S. Hochreiter. Furby: fuzzy force-directed bicluster visualization. *BMC bioinformatics*, 15(Suppl 6):S4, 2014.
- [48] M. Sun, L. Bradel, C. L. North, and N. Ramakrishnan. The role of interactive biclusters in sensemaking. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 1559–1562. ACM, 2014.
- [49] M. Sun, C. North, and N. Ramakrishnan. A Five-Level Design Framework for Bicluster Visualizations. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):1713–1722, 2014.
- [50] A. Telea and O. Ersoy. Image-Based Edge Bundles: Simplified Visualization of Large Graphs. *Computer Graphics Forum*, 29(3):843–852, 2010.
- [51] T. Uno, T. Asai, Y. Uchida, and H. Arimura. An efficient algorithm for enumerating closed patterns in transaction databases. In *Discovery Science*, pages 16–31. Springer, 2004.
- [52] C. Viau and M. J. McGuffin. ConnectedCharts: Explicit Visualization of Relationships between Data Graphics. *Computer Graphics Forum*, 31(3pt4):1285–1294, June 2012.
- [53] M. Waldner, W. Puff, A. Lex, M. Streit, and D. Schmalstieg. Visual links across applications. In *Proceedings of Graphics Interface 2010*, pages 129–136. Canadian Information Processing Society, 2010.
- [54] H. Wu, J. Vreeken, N. Tatti, and N. Ramakrishnan. Uncovering the plot: detecting surprising coalitions of entities in multi-relational schemas. *Data Mining and Knowledge Discovery*, 28(5-6):1398–1428, 2014.
- [55] H.-M. Wu, Y.-J. Tien, and C.-h. Chen. Gap: A graphical environment for matrix visualization and cluster analysis. *Computational Statistics & Data Analysis*, 54(3):767–778, 2010.
- [56] M. J. Zaki and C.-J. Hsiao. Efficient algorithms for mining closed itemsets and their lattice structure. *Knowledge and Data Engineering, IEEE Transactions on*, 17(4):462–478, 2005.
- [57] H. Zhang, M. Sun, D. D. Yao, and C. North. Visualizing traffic causality for analyzing network anomalies. In *Proceedings of the 2015 ACM International Workshop on International Workshop on Security and Privacy Analytics*, pages 37–42. ACM, 2015.
- [58] H. Zhou, P. Xu, X. Yuan, and H. Qu. Edge bundling in information visualization. *Tsinghua Science and Technology*, 18(2):145–156, 2013.