Usability Challenges underlying Bicluster Interaction for Sensemaking

Maoyuan Sun
Virginia Polytechnic Institute
and State University
smaoyuan@cs.vt.edu

Chris North
Virginia Polytechnic Institute
and State University
north@cs.vt.edu

Peng Mi
Virginia Polytechnic Institute
and State University
mipeng@vt.edu

Naren Ramakrishnan
Virginia Polytechnic Institute
and State University
naren@cs.vt.edu

Hao Wu
Virginia Polytechnic Institute
and State University
wuhao723@vt.edu

Abstract

Exploring coordinated relationships (e.g., four people who all visit the same five cities), and understanding stories revealed from them is important to support sensemaking tasks (e.g., intelligence analysis). *Biclusters* can support this because they algorithmically bundle individual relations into coordinated sets. The computed, structural relations within biclusters enable analysts to leverage domain knowledge and intuition to determine the importance and relevance of extracted relationships for making hypotheses. However, to make biclusters usable, there are challenges in four key aspects: *algorithm comparison*, *algorithm design and parameter manipulation*, *bicluster visualization* and *bicluster evaluation*. These challenges raise usability oriented questions about usable biclusters, such as which algorithm(s) to use for bicluster discovery, how to design human-centered biclustering algorithms, and how to visualize and evaluate biclusters. In this paper, we present these usability challenges to inform future research directions, particularly focusing on visual analytics with biclusters.

Author Keywords

Bicluster; coordinated relationship; visual analytics.

ACM Classification Keywords

H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous.

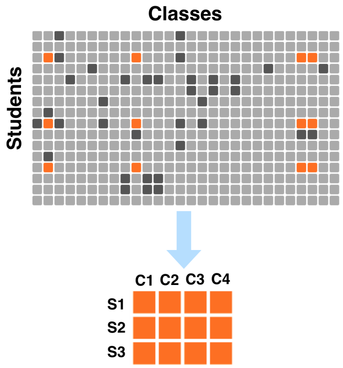


Figure 1: A bicluster, from a students-to-classes relationship, reveals three students took the same four classes. Dark cells indicate existing relations and orange cells represent relations belong to this bicluster.

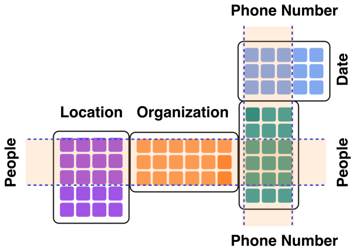


Figure 2: Chaining four biclusters through multiple relations by approximately matching sets of entities across common domains.

Introduction

Exploring meaningful coordinated relations is a common task in data analytics. Coordinated relationships are groups of shared relations between sets of entities from different domains (e.g., people, location, date, etc.). For example, intelligence analysts often investigate large unstructured textual datasets to identify coordinated activities that might be evidence for collusion [6]. Bioinformaticians explore coordinated relations from expression and interaction datasets to identify groups of genes and/or proteins that are commonly expressed or regulated conditions and species [1]. Cyber security analysts trace coordinated relations between processes, hosts and network domains to detect distributed coordinated attacks [14].

Coordinated relationship discovery needs significant cognitive effort. This process often includes three repetitious steps: 1) identify and extract meaningful entities, 2) check entities to verify whether a set of entities are related to the same entity or entities, and 3) group entities based on their shared relations. For example, to find four people who all visited the same five cities, analysts may have to read numerous documents, identify names and cities from them, compare many co-occurring people-city pairs among different scenarios, and test many possible combinatorial groupings of the pairs, before they finally get an answer.

As algorithmically identified coordinated relations, biclusters can ease this process. Biclusters are results of biclustering algorithms and have been applied in bioinformatics [3, 8] and intelligence analysis [5, 10]. A bicluster in a relation can be viewed as a bundling of individual relationships into a pair of sets. For instance, as is shown in Figure 1, from a relationship capturing attendance of students in classes, we can find a bicluster involving a set of students [S1, S2, S3] who all attend the same set of classes [C1, C2, C3, C4].

Biclusters and Bicluster Chains

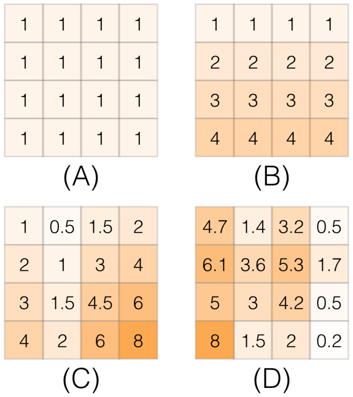
Biclustering finds both subsets of entities and subsets of dimensions and require that for each identified subset of entities, they identically behave within its corresponding subset of dimensions [8]. *Biclusters* are computational outcomes of biclustering algorithms that identify coordinated relations between two entity sets. An entity set refers to a set of unique elements 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. For example, in text analytics, R can be determined by word co-occurrence or semantic meanings identified with natural language processing. For instance, 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. A *bicluster* (E', F') on *R (E, F)* is defined 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'. $|E'| + |F'|$ denotes the *size* of a *bicluster* (E', F'), where $|E'|$ and $|F'|$ are the cardinality of E' and 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 $(e, f) \notin R$. In this paper, our notation of *biclusters* refers to *closed biclusters*.

Bicluster Types and Structures. Based on values of cells in a data matrix, there are four major types of biclusters [8],



(A)

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

(B)

1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4

(C)

1	0.5	1.5	2
2	1	3	4
3	1.5	4.5	6
4	2	6	8

(D)

4.7	1.4	3.2	0.5
6.1	3.6	5.3	1.7
5	3	4.2	0.5
8	1.5	2	0.2

Figure 3: *(A)* a bicluster with constant values. *(B)* a bicluster with constant values on rows. *(C)* a bicluster with row level of coherent values. *(D)* a bicluster with coherent evolutions on rows (the order of cells, based on values, in each row remains the same).

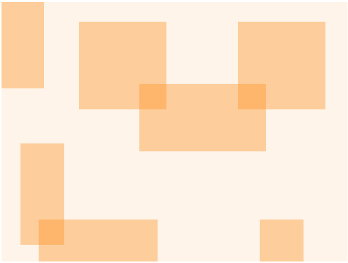


Figure 4: An example of arbitrarily positioned overlapping biclusters.

summarized as follows. Figure 3 shows examples of them.

- Constant values.
- Row/column level of constant values.
- Row/column level of coherent values.
- Row/column level of coherent evolutions.

Biclustering algorithms attempt to find $m(m \geq 1)$ biclusters from a data matrix in most cases [8], and the identified biclusters often overlaps each other with shared relation(s). Based on overlaps among discovered biclusters (composed by entities from two specific domains) from biclustering algorithms, there are, in total, eight different bicluster structures (more details can be found in [8]). Moreover, different biclustering algorithms may use and emphasize different criterion for bicluster discovery, so for the same data matrix, biclusters, generated from different algorithms, may have different (overlapping) structures. To make our discussion general, the notion of bicluster structure (or overlap), in this paper, refers to *arbitrarily positioned overlapping*. Figure 4 presents an example of this overlapping structure.

Bicluster-chains. Since every bicluster is discovered in a single relation, it is possible to compose separately identified biclusters across two relations by (approximately) matching biclusters with shared domains. This produces *bicluster-chains*, which can be identified from a dataset with compositional mining methods [7]. As is shown in Figure 2, four biclusters, indicating four different relations, can be chained together using common interfaces (e.g., use *people* to connect the purple bicluster with the orange one). By chaining biclusters across multiple relations, relationships from a diversity of domains can be bundled in a coherent manner. Moreover, results of such compositions can be read sequentially from one end to the other.

With above notations of biclusters and bicluster-chains, we reach a common ground about these key concepts used in this paper. Moreover, we summarize key attributes of biclusters with examples in Table 1. Two of these attributes (schema and size) determine how biclusters are algorithmically discovered, which are algorithm parameters controlled by users. Other attributes come from algorithm outputs. These attributes play a key role for users to control, explore and understand biclusters from algorithms.

Usability Challenges

With algorithm parameters (e.g., size), users can do some control on bicluster discovery. However, there are still usability challenges of interacting with biclusters for sensemaking, particularly in four key aspects: 1) algorithm comparison, 2) algorithm design and parameter manipulation, 3) bicluster visualization, and 4) bicluster evaluation. These challenges raise questions about usable biclusters from four different levels: *model* level, *parameter* level, *representation* level and *evaluation* level.

Algorithm Comparison

How to enable users to reasonably select biclustering algorithm(s) is the first challenge to make biclusters usable. As mentioned before, different algorithms may use different criterion for bicluster discovery, so different algorithms may find different biclusters for the same dataset. Thus, it is necessary for users to decide which algorithm(s) to use by leveraging algorithmic results with their domain knowledge and analysis tasks. Compared with arbitrary selections, making comparison among different algorithms can better help user decision making. However, *how to compare different biclustering algorithms* still remains a question. Specifically, *what aspects of biclustering algorithms (e.g., parameters, performance, results, etc.) are usable for comparison? Are these driven by specific user tasks?*

	Attribute	Example	Algorithm
Individual Biclusters	Schema	People - Location	Input
	Size	2×2	Input
	Members	{Alex, John} - {Boston, Seattle}	Output
	Individual Entity Frequency	Alex: 2, John: 3, Boston: 4, Seattle: 3	Output
	Individual Relation Frequency	Alex - Boston: 2, Alex - Seattle: 1, John - Boston: 2, John - Seattle: 2	Output
Multiple Biclusters	Overlaps between biclusters composed by two specific domains	{*Alex*, John} - {*Boston*, Seattle}, {*Alex*, Chris, Sarah} - {*Boston*, New York, Pittsburgh}	Output
	Overlaps between biclusters sharing one common domain	{Alex, John} - {Boston, Seattle} (*People* - Location), {Alex, John, Sarah} - {Apple, Google} (*People* - Organization)	Output
	Number	People - Location: 23, People - Organization: 15	Output

Table 1: A summary of bicluster attributes that are potentially usable to support exploring coordinated relations

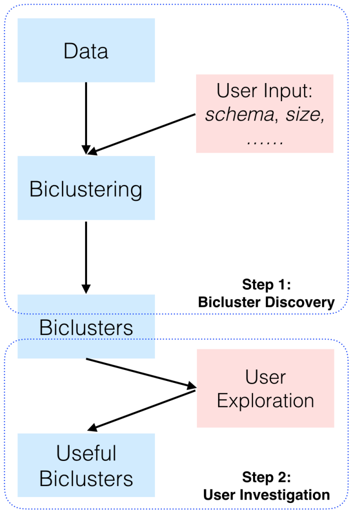


Figure 5: The existing paradigm of using biclusters for sensemaking.

Algorithm Design and Parameter Manipulation
Algorithm design and enabling (novel) user interactions to steer algorithms is another challenge of interacting with biclusters. Currently, the procedure of using biclusters for sensemaking includes two sequential steps: algorithmic bicluster discovery and user investigation. Figure 5 shows the paradigm of this process. This paradigm has been applied in recent visual analytics tools (e.g., BiSet [11], Bixplorer [5], Furby [9], etc.). In these tools, users interact with visual metaphors of biclusters to explore meaningful and useful ones from algorithmic results. However, user reasoning results (e.g., biclusters identified as meaningful ones) cannot be interpreted by the selected biclustering algorithm(s) and further impact the bicluster discovery process in future. Thus, users have to passively "accept" all algorithm results and then do explorations. This limits human investigations to a function of post-clustering filters.

Semantic interaction offers a novel way for users to interact with machine learning algorithms, and it uses (interpret) algorithms to enable the injection of user reasoning into a computational process [4]. *Can this concept be applied to inform human-centered biclustering algorithm design?* Particularly, in addition to existing algorithm parameters (e.g., schema and size), *what additional parameters can we add to biclustering algorithms to control bicluster discovery?* For example, if users identify locations, *Boston* and *Seattle*, as useful ones, how can we inform algorithms of this information, and further steer them to find biclusters containing these locations? Different from the sequential paradigm depicted in Figure 5, this requires iterative processes. Since user decisions and/or intentions can be inferred by algorithms, the role of user investigations becomes more active to steer algorithms, rather than just post-clustering filters.

How to enable user to manipulate algorithm parameters is another key question. Besides command line, there are two ways for users to adjust algorithm parameters: user interface widget (e.g., sliders, buttons, spinner, etc.), and direct manipulation (e.g., the "near - similar" metaphor in semantic interaction [4]). The latter is less precise in parameter adjustment than the former. For example, when users drag

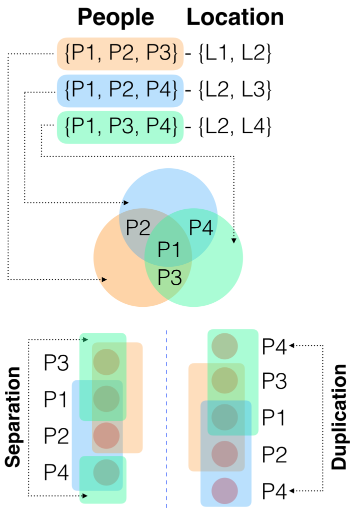


Figure 6: A example of the Euler diagram problem that arises when visually displaying the membership of entities in the domain of *people* shared by three biclusters.

one node closer to another, they may not know the exact distance between the two nodes. However, when they use a spinner, they can precisely change the distance between the two nodes. For existing biclustering algorithm parameters (e.g., size), user interface widgets may be a good choice due to the precise adjustment capability. For other parameters, possibly identified and added in future human-centered biclustering algorithms discussed above, *how can we provide usable ways for users to interact with them?*

Bicluster Visualization

How to visualize biclusters is the third key challenge to enable users to interact with biclusters. Algorithmically identified biclusters exist in a machine readable format. To make them usable, we need present biclusters in a human understandable manner. This raises a fundamental visualization problem, particularly when considering the overlap among biclusters. This is similar to the Euler diagram problem but even harder. Figure 6 shows an example of this problem. There are three biclusters displaying different coordinated relations between people and locations. Due to the overlap, it is difficult to clearly present both entities (without duplication) and biclusters (without separation). This is identified as the key design trade-off (*entity*-centric versus *relationship*-centric) for bicluster visualizaitons [11]. To balance this trade-off, BiSet [11] has been proposed. It uses list to show both entities and biclusters in a clearly organized way. However, due to lacking functions to aggregate (similar) biclusters, BiSet may show long lists of individual biclusters for large datasets, which overwhelms users. This leads to a layout challenge: *how to merge and/or split biclusters*.

To help users understand biclusters, five levels of relationships (*entity*-level, *group*-level, *bicluster*-level, *chain*-level and *schema*-level) have been identified [12]. These relations construct a relationship spectrum, which allows users to interpret biclusters from either low-level entities or high-level schemas. Biclusters locate in the middle of the spectrum. *How to visualize this relationship spectrum and enable users to traverse relations across different levels* is a challenge, worth further exploration. This offers two potential benefits. For one thing, such traverse reflects human reasoning process and strategies to explore different relations, which may be useful to inform future algorithm design by learning from human. For another thing, guiding users to traverse in the relationship spectrum for bicluster exploration, may simulate how biclustering algorithms work. This helps users to understand biclusters and how biclustering algorithms work, which are useful from an educational perspective (e.g., teaching biclustering algorithms).

Bicluster Evaluation

Besides visual aggregation and salient, computationally prioritizing biclusters helps to direct user attention to useful ones. This is useful especially when handling large datasets. However, *how to evaluate biclusters and prioritize them both computationally and visually* still remains a challenge. Using BiSet and maximum entropy model (MaxEnt), a preliminary attempt has been performed in [13]. In this exploration, biclusters are evaluated using MaxEnt, based on entity distribution. According to the score of each evaluated biclusters, given by MaxEnt, BiSet visually prioritize them with color codings. With a case study, this approach has been reported with promising results, but it still needs users to interpret semantic connections of statistically relevant biclusters. Besides such distribution based evaluation, *can biclusters be evaluated based on semantic meanings or meta data [2]?* Moreover, *how can we incorporate such computational evaluations into a user reasoning process to support sensemaking progressively, and how can we enable users interactively control these evaluations?*

Conclusion

As bundled sets of individual relations, biclusters can support exploring coordinated relations. To make biclusters usable, there are usability challenges in four key aspects: algorithm comparison, algorithm design and parameter manipulation, bicluster visualization, and bicluster evaluation. By discussing these challenges, we hope that it can help to inform future research directions of visual analytics with biclusters to support sensemaking tasks.

Acknowledgements

This research was partially supported by NSF grant IIS-1447416.

References

[1] David B Allison, Xiangqin Cui, Grier P Page, and Mahyar Sabripour. 2006. Microarray data analysis: from disarray to consolidation and consensus. *Nature Reviews Genetics* 7, 1 (2006), 55–65.

[2] Rich Caruana, Mohamed Elhawary, Nam Nguyen, and Casey Smith. 2006. Meta clustering. In *Data Mining, Sixth International Conference on*. IEEE, 107–118.

[3] Yizong Cheng and George M Church. 2000. Biclustering of expression data.. In *Ismb*, Vol. 8. 93–103.

[4] Alex Endert, Patrick Fiaux, and Chris North. 2012. Semantic interaction for visual text analytics. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 473–482.

[5] Patrick Fiaux, Maoyuan Sun, Lauren Bradel, Chris North, Naren Ramakrishnan, and Alex Endert. 2013. Bixplorer: Visual analytics with biclusters. *Computer* 8 (2013), 90–94.

[6] F Hughes and D Schum. 2003. 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).

[7] Ying Jin, T M Murali, and Naren Ramakrishnan. 2008. Compositional mining of multirelational biological datasets. *ACM Transactions on Knowledge Discovery from Data* 2, 1 (March 2008), 1–35.

[8] Sara C Madeira and Arlindo L Oliveira. 2004. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* 1, 1 (2004), 24–45.

[9] Marc Streit, Samuel Gratzl, Michael Gillhofer, Andreas Mayr, Andreas Mitterecker, and Sepp Hochreiter. 2014. Furby: fuzzy force-directed bicluster visualization. *BMC bioinformatics* 15, Suppl 6 (2014), S4.

[10] Maoyuan Sun, Lauren Bradel, Chris L North, and Naren Ramakrishnan. 2014. The role of interactive biclusters in sensemaking. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*. ACM, 1559–1562.

[11] Maoyuan Sun, Peng Mi, Chris North, and Naren Ramakrishnan. 2016. BiSet: Semantic Edge Bundling with Biclusters for Sensemaking. *Visualization and Computer Graphics, IEEE Transactions on* 22, 1 (2016), 310–319.

[12] Maoyuan Sun, Chris North, and Naren Ramakrishnan. 2014. A Five-Level Design Framework for Bicluster Visualizations. *Visualization and Computer Graphics, IEEE Transactions on* 20, 12 (2014), 1713–1722.

[13] Hao Wu, Maoyuan Sun, Peng Mi, Nikolaj Tatti, Chris North, and Naren Ramakrishnan. 2015. Interactive Discovery of Coordinated Relationship Chains with Maximum Entropy Models. *arXiv preprint arXiv:1512.08799* (2015).

[14] Chenfeng Vincent Zhou, Christopher Leckie, and Shanika Karunasekera. 2010. A survey of coordinated attacks and collaborative intrusion detection. *Computers & Security* 29, 1 (2010), 124–140.