Interactive Discovery of Coordinated Relationship Chains with Maximum Entropy Models

HAO WU*, Virginia Tech
MAOYUAN SUN*, University of Massachusetts Dartmouth
PENG MI, Virginia Tech
NIKOLAJ TATTI, Aalto University
CHRIS NORTH, Virginia Tech
NAREN RAMAKRISHNAN, Virginia Tech

Modern visual analytic tools promote human-in-the-loop analysis but are limited in their ability to direct the user toward interesting and promising directions of study. This problem is especially acute when the analysis task is exploratory in nature, e.g., the discovery of potentially coordinated relationships in massive text datasets. Such tasks are very common in domains like intelligence analysis and security forensics where the goal is to uncover surprising coalitions bridging multiple types of relations. We introduce new maximum entropy models to discover surprising chains of relationships leveraging count data about entity occurrences in documents. These models are embedded in a visual analytic system called MERCER that treats relationship bundles as first class objects and directs the user toward promising lines of inquiry. We demonstrate how user input can judiciously direct analysis toward valid conclusions whereas a purely algorithmic approach could be led astray. Experimental results on both synthetic and real datasets from the intelligence community are presented.

CCS Concepts: •**Mathematics of computing** →**Exploratory data analysis;** •**Human-centered computing** →**Visual analytics;** •**Computing methodologies** →**Maximum entropy modeling;** •**Information systems** →*Data mining;*

Additional Key Words and Phrases: Maximum entropy models, multi-relational pattern mining, interactive visual data exploration

ACM Reference format:
Hao Wu*, Maoyuan Sun*, Peng Mi, Nikolaj Tatti, Chris North, and Naren Ramakrishnan. 2010. Interactive Discovery of Coordinated Relationship Chains with Maximum Entropy Models. *ACM Trans. Knowl. Discov. Data.* 9, 4, Article 39 (March 2010), 36 pages.
DOI: 0000001.0000001

*These authors contributed equally to this work.

Author's addresses: H. Wu and N. Ramakrishnan, Discovery Analytics Center, Virginia Tech, Arlington, VA, USA; M. Sun, Computer and Information Science, University of Massachusetts Dartmouth, Dartmouth, MA, USA; P. Mi, VMWare Inc., Palo Alto, CA, USA; N. Tatti, Department of Information and Computer Science, Aalto University School of Science, Aalto, Finland; C. North, Discovery Analytics Center, Virginia Tech, Blacksburg, VA, USA.

DOI: 0000001.0000001

1 INTRODUCTION

Unstructured exploration of relationships from large text datasets is a crucial problem in many application domains, for example, intelligence analysis, biomedical discovery, analysis of legal briefs and opinions. The state-of-the-art today involves two broad classes of techniques. Visual analytic tools, for example Jigsaw [47], support the exploration of relationships extracted from large text datasets. While they promote human-in-the-loop analysis, identifying promising leads to explore is left to the creativity of the user. At the other end of the spectrum, text relationship exploration techniques such as storytelling [21] provide interesting artifacts (for example, stories, summaries) for analysis but are limited in their ability to incorporate user input to steer the discovery process.

Our goal here is to realize an amalgamation of algorithmic and human-driven techniques to support the discovery of coordinated relationship chains from document collections. A coordinated relationship (also called a bicluster) is one in which a group of entities are related to another group of entities via a common relation. It is thus a generalization of a relationship instance. A chain of such coordinated relationships enables us to bundle groups of entities across various domains and relate them through a succession of individual relationships. The primary artifact of interest are thus chains summarizing how entities in a document collection are related. We introduce new maximum entropy (MaxEnt) models to identify surprising chains of interest and rank them for inspection by the user. In intelligence analysis, such chains can reveal how hitherto unconnected people or places are related through a sequence of intermediaries. In biomedical discovery, such chains can reveal how proteins involved in distinct pathways are related through cross-talk via other proteins or signaling molecules. In legal briefs, one can use chains to determine how rationale for court opinions vary over the years and are buttressed by the precedence structure implicit in legal history.

As shown in Fig. 1 (left), we propose an interactive approach wherein user feedback is woven at each stage and used to rank the most interesting chains for further exploration. Such user feedbacks are also taken into account by the algorithm for further investigations of the data. We will demonstrate through case studies how such an approach gets users to their intended objectives compared to a purely algorithmic approach (Fig. 1 (right)). The work presented here is implemented in a system – Maximum Entropy Relational Chain ExploRer (MERCER) that uses a variety of visual exploration strategies and algorithmic means to foster user exploration.

Our key contributions are:

(1) MERCER is a marriage of two of our prior works [51, 61] but supercedes the state-of-the-art in these papers in significant, orthogonal, ways. MERCER is a significant improvement over the work presented by Sun et al. [51] because the authors provide support for only manual exploration of coordinated relationships. MERCER is also a significant improvement over the work presented by Wu et al. [61] because this work only presents approaches to rank chains involving a binary maximum entropy model whereas MERCER introduces more general maximum entropy approaches for real-valued data.

(2) We present two path strategies (full path and stepwise) to help analyze datasets. Using our proposed maximum entropy models, the full path strategy discovers the most surprising bicluster chains from all possible chains involving an analyst-selected bicluster. The stepwise strategy evaluates biclusters neighboring a user-specified one, and prioritizes possible connected information with the current pieces under

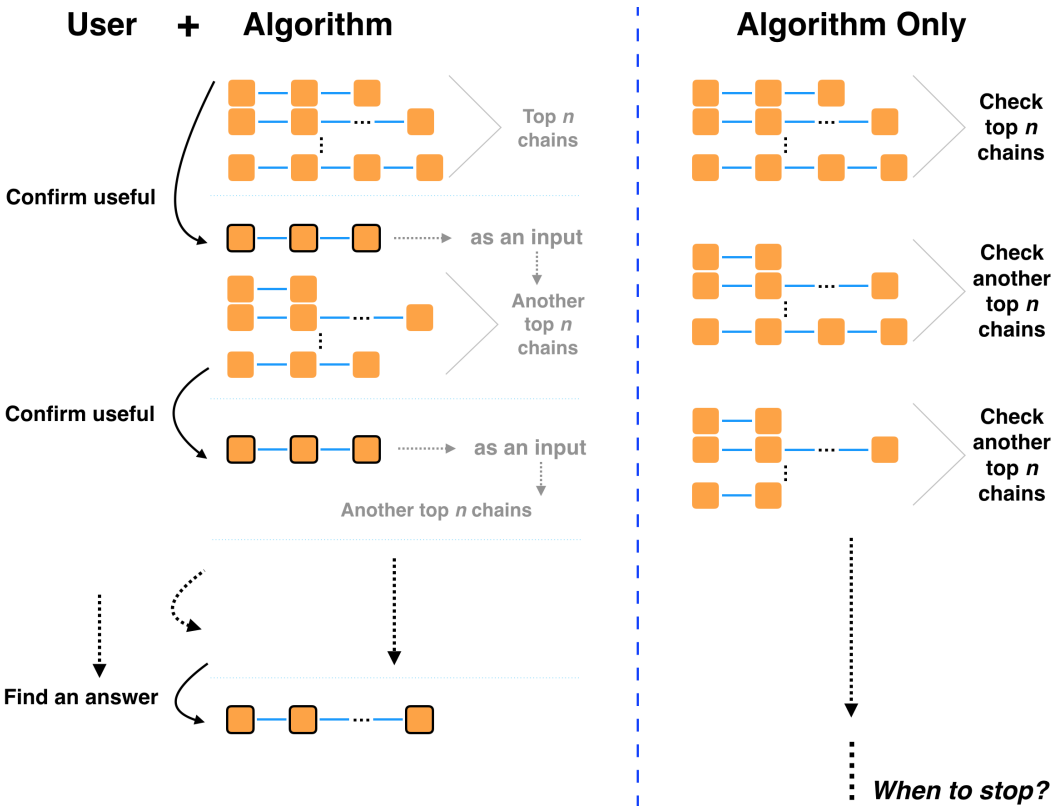

Fig. 1. Illustration of MERCER. (left) Discovery of coordinated relationship chains is aided by regular incorporation of user feedback. (right) Unaided algorithmic discovery of relationship chains leads to long lists of patterns that might not lead to the desired answer.

investigation. Both strategies directs analysts to reveal hidden plots involving surprising relational patterns.

(3) We describe new visual encodings and summary as well as detailed views to support user-guided exploration of coordinated relationships in massive datasets. Besides basic color codings (for example, *connection-oriented* highlighting [51]), MERCER offers highlighting mechanisms aimed at pointing out surprising information. Enhanced with the proposed maximum entropy models, this highlighting capability not only directs user's attention to important connected pieces of information, but also visually prioritizes them in a usable manner.

(4) We describe experimental results on both large, synthetic datasets (to illustrate efficiency and effectiveness of our algorithms) and small, real datasets (to illustrate how users can interactively explore a realistic text dataset). In particular, we show how MERCER enables the user to more quickly arrive at plots of interest than the traditional manual approach described in our previous work [51].

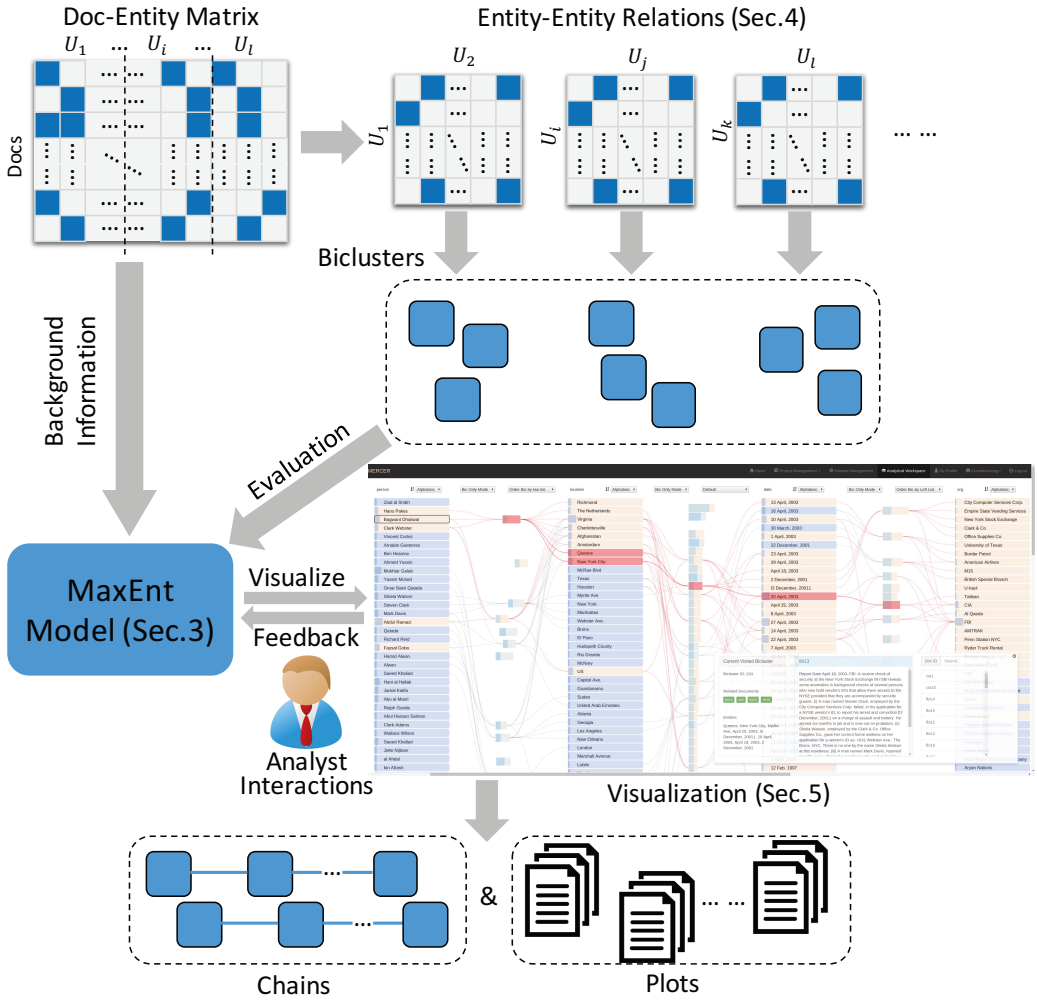

Fig. 2. MERCER system workflow.

2 PRELIMINARIES

Figure 2 illustrates the workflow in MERCER. By taking the background information from the document-entity transactional matrix, the MERCER system infers the maximum entropy model, which will be described in detail in Section 3. From the document-entity matrix, multiple entity-entity relations are extracted and surprisingness measure for relational patterns is defined based on the MaxEnt model (Section 4). By interacting with analysts, our visualization interface displays the surprising relational patterns discovered from the multiple entity-entity relations, and also provides analysts' feedback to the MaxEnt model, which will in turn help to further discover additional surprising patterns (Section 5). In this section, we introduce some preliminary concepts and notations that will be useful to understand the MERCER system and the rest of this paper.

Multi-relational schema. Suppose that we have l domains or *universes* which will be denoted by U_i, where $i = 1, 2, \ldots, l$, throughout the paper. An *entity* is a member of U_i and an *entity set* is just a subset of U_i. We use $R = R(U_i, U_j)$ to represent a binary relation between some domains U_i and U_j. Given a set of domains $\mathcal{U} = \{U_1, U_2, \ldots, U_l\}$ and a set of relations $\mathcal{R} = \{R_1, R_2, \ldots, R_m\}$, a *multi-relational schema* $S(\mathcal{U}, \mathcal{R})$ is defined as a connected bipartite graph whose vertex set is given by $\mathcal{U} \cup \mathcal{R}$ and edge set is the collection of edges each of which connects a relation $R_j \in \mathcal{R}$ and a domain $U_i \in \mathcal{U}$ that the relation R_j involves. In this paper, we will focus on binary relationships, e.g. each R_j is a binary relation. Thus, all the vertices in $\mathcal{R}$ in the bipartite graph will have degree of two. Binary relations usually can be represented as binary data matrices. In the rest of this paper, we will use these two terms interchangeably depending on which one is easier to use to present and explain our proposed model and algorithm.

Tiles. A *tile* [14] T is essentially a rectangle in a binary data matrix. Formally, it is defined as a tuple $T = (r(T), c(T))$ where $r(T)$ is a set of row identifiers (e.g., row IDs) and $c(T)$ is a set of column identifiers (e.g., column IDs) over the data matrix. This most general form definition imposes no constraints on values of the matrix entries identified by a tile. Thus, each element in a tile can be any valid value in the data matrix. In the binary case, when all entries within a tile T have the same value (i.e., either all 1s or all 0s), T is an exact tile. Otherwise we say it is a noisy tile.

Biclusters. As local patterns of interest over binary relations, we consider binary biclusters. Although the concept of biclusters was first introduced over real-valued data [5], as a generalization of this concept to binary data, we will use the word *biclusters* to refer to the local patterns over binary relations defined below in the rest of this paper. A *bicluster*, represented by $B = (E_i, E_j)$, on a relation $R = R(U_i, U_j)$, consists of two entity sets $E_i \subseteq U_i$ and $E_j \subseteq U_j$ such that $E_i \times E_j \subseteq R$. As such a bicluster is a special case of an exact tile, one in which all the elements are 1. Further, we say a bicluster $B = (E_i, E_j)$ is *closed* if for every entity $e_i \in U_i \setminus E_i$, there is some entity $e_j \in E_j$ such that $(e_i, e_j) \notin R$ and for every entity $e_j \in U_j \setminus E_j$, there is some entity $e_i \in E_i$ such that $(e_i, e_j) \notin R$. In other words, E_i is maximal (w.r.t. E_j) so that we cannot add more elements to E_i without violating the premise of a bicluster. If a pair of entities $e_i \in U_i, e_j \in U_j$ belongs to a bicluster B, we represent this fact by $(e_i, e_j) \in B$. In the rest of this paper, all the biclusters we mention refer to closed biclusters.

Redescriptions. Suppose that we have two biclusters $B = (E_i, E_j)$ and $C = (F_j, F_k)$, where $E_i \subseteq U_i$, $E_j, F_j \subseteq U_j$, and $F_k \subseteq U_k$. Note that E_j and F_j lie in the same domain. Assume that we are given a threshold $0 \leq \varphi \leq 1$. We define that B and C are *approximate redescriptors* of each other, which we represent by $B \sim_{\varphi,j} C$ if the Jaccard coefficient $|E_j \cap F_j| \,/\, |E_j \cup F_j| \geq \varphi$. The threshold φ is usually specified by users, consequently we often drop φ from the notation and write $B \sim_j C$. The index j indicates the common domain over which we should take the Jaccard coefficient. When this domain is clear from the context we often ignore the index j from the notation. If $B \sim_{1,j} C$, then we must have $E_j = F_j$ in which case we say that B is an *exact redescription* of C. This definition coincides with the definition given by Zaki and Ramakrishnan [64], who define redescriptions for itemsets over their mutual domain, transactions, such that the set E_j consists of transactions containing itemset E_i and the set F_j consists of transactions containing itemset F_k.

Bicluster Chains. We define a *bicluster chain* C as an ordered set of biclusters $\{B_1, B_2, \ldots, B_k\}$ and an ordered bag of domain indices $\{j_1, j_2, \ldots, j_{k-1}\}$ such that for each pair of adjacent biclusters they are redescriptions of each other, e.g. $B_i \sim_{j_i} B_{i+1}$. Note that this definition implicitly requires that two adjacent biclusters share a common domain. If a bicluster B_{R_i} is a member of a bicluster chain C, we will denote this by $B_{R_i} \in C$ in this paper.

Surprisingness. In the knowledge discovery tasks studied here, the primary goal is to extract novel, interesting, or unusual knowledge. That is, we aim to discover results that are highly informative compared to what we already know—we are not so much interested in what we already do know, or what can be trivially induced from such knowledge. To this end, we suppose a probability distribution p that represents the user's current beliefs about the data. When mining the data (e.g., for a bicluster or chain), we can use p to determine the likelihood of a result under our current beliefs: if the likelihood is high, this indicates that we probably already know about it, and thus, reporting it to users would provide little new information. In contrast, if the likelihood of a result is very low, the result would be quite interesting, thus potentially conveying a lot of new information. In Section 3, we will discuss how to infer such a probability distribution for both binary and real-valued data matrices.

Problem Statement. Given a multi-relational dataset, a bicluster chain across multiple relations describes a progression of entity coalitions. We are particularly interested in chains that are surprising w.r.t. what we already know since these could help to uncover the plots hidden in the multi-relational dataset. More formally, given a multi-relational dataset schema $S(\mathcal{U}, \mathcal{R})$, where $\mathcal{U} = \{U_1, U_2, \ldots, U_l\}$ and $\mathcal{R} = \{R_1, R_2, \ldots, R_m\}$, we aim to iteratively discover non-redundant bicluster chains that are most surprising with respect to each other and w.r.t. the background knowledge with the assistance of visual analysis techniques.

3 TILE-BASED MAXIMUM ENTROPY MODEL

Our problem statement is based on a notion of a multi-relational schema. In practice, one approach to infer such multi-relational datasets from a transactional dataset is to rely on the item co-occurrence information in transactions where these items involved in binary relations are from different domains (e.g., entities discovered from a document collection, and then subsequently related by co-occurrence). More specifically, we assume that our schema was generated from a transactional data matrix D (see Fig. 2). This data matrix can be viewed as a matrix of size N-by-M. We will introduce the method of obtaining a schema from D in Section 4. In this approach the columns of D correspond to the entities of the schema. Hence, we will refer to the columns of D as entities.

3.1 Maximum Entropy Model for Binary Data

In this section, we will formally define the maximum entropy (MaxEnt) model for binary data matrices using tiles as background knowledge—recall that a tile is a more general notion than a bicluster. In order to understand the model derivation in the context of binary data, we first introduce some notations for binary MaxEnt model. Then, MaxEnt theory for modeling binary data given tiles as background information will be reviewed, and finally, we will identify how we can estimate the model by maximizing the likelihood.

3.1.1 Notation for Tiles. Suppose we are given a binary data matrix D of size N-by-M and a tile T, we define the (relative) frequency of T in D, $fr(T;D)$, as

$$fr(T;D) = \frac{1}{|\sigma(T)|} \sum_{(i,j)\in\sigma(T)} D(i,j) \quad . \tag{1}$$

Here, $D(i,j)$ denotes the entry (i,j) in D, and $\sigma(T) = \{(i,j) \mid i \in r(T), j \in c(T)\}$ represents the cells covered by tile T in the data matrix D. Remember that a tile T is called 'exact' if the corresponding entries $D(i,j)$ are all 1 (resp. 0) for all $(i,j) \in \sigma(T)$. This indicates for exact tiles, $fr(T;D) = 1$ or $fr(T;D) = 0$. Otherwise, it is called a 'noisy' tile.

Let $\mathcal{D}$ be the space of all the possible binary data matrices of size N-by-M, and p be the probability distribution defined over the data matrix space $\mathcal{D}$. Then, the expected frequency of the tile T with respect to the data matrix probability distribution p is defined as

$$fr(T;p) = \mathbb{E}\left[fr(T;D)\right] = \sum_{D\in\mathcal{D}} p(D) fr(T;D). \tag{2}$$

By combining these definitions, we can derive the following lemma.

LEMMA 3.1 ([61]). *Given a dataset distribution p and a tile T, the expected frequency of tile T is*

$$fr(T;p) = \frac{1}{|\sigma(T)|} \sum_{(i,j)\in\sigma(T)} p\left((i,j) = 1\right) \quad ,$$

where $p((i,j) = 1)$ represents the probability of a data matrix having 1 at entry (i,j) under the data matrix distribution p.

Lemma 3.1 can be trivially proved by substituting $fr(T;D)$ in Equation (2) with Equation (1) and switching the summations.

3.1.2 Global MaxEnt Model from Tiles. Suppose we are given a set of tiles $\mathcal{T}$, and each tile $T \in \mathcal{T}$ is associated with a frequency γ_T—which typically can be trivially calculated from the data. This tile set $\mathcal{T}$ provides information about the data at hand, and we would like to estimate a distribution p over the space of all the possible data matrices $\mathcal{D}$ which conform with the information given in $\mathcal{T}$. In other words, we would like to be able to determine how probable is a data matrix $D \in \mathcal{D}$ given the tile set $\mathcal{T}$.

To derive a good statistical model, we adopt a principled approach and apply the maximum entropy principle [23] from information theory. Generally speaking, the MaxEnt principle identifies the best distribution given background knowledge as the unique distribution which represents the provided background information but is maximally random otherwise. MaxEnt modeling has recently attracted much attention in the realm of data mining as a tool for identifying *subjective* interestingness of results with respect to background knowledge [10, 29, 55, 60].

To formally define a MaxEnt distribution, we first specify the space of probability distribution candidates. Here, these are all the possible data matrix distributions which are consistent with the information given by the tile set $\mathcal{T}$. Hence, we define the data matrix distribution space as: $\mathcal{P} = \{p \mid fr(T;p) = \gamma_T, \forall T \in \mathcal{T}\}$. Among all these possible distribution candidates, we choose the distribution $p^*_{\mathcal{T}}$ that maximizes the entropy,

$$p^*_{\mathcal{T}} = \arg\max_{p\in\mathcal{P}} H(p) \quad .$$

Here, $H(p)$ denotes the entropy of the data matrix probability distribution p, which is defined as

$$H(p) = -\sum_{D \in \mathcal{D}} p(D) \log p(D) \quad .$$

Next, to infer the MaxEnt distribution $p^*_{\mathcal{T}}$, we rely on a classical theorem about how MaxEnt distributions can be factorized. In particular, Theorem 3.1 proved by Csiszar [6] states that for a given set of testable statistics $\mathcal{T}$ (background knowledge, here a tile set), a distribution $p^*_{\mathcal{T}}$ is the maximum entropy distribution if and only if it can be written as

$$p^*_{\mathcal{T}}(D) \propto \begin{cases} \exp\left(\sum_{T \in \mathcal{T}} \lambda_T \cdot |\sigma(T)| \cdot fr(T;D)\right) & D \notin \mathcal{Z} \\ 0 & D \in \mathcal{Z} \end{cases} ,$$

where λ_T is the weight for $fr(T;D)$ and $\mathcal{Z}$ is a collection of data matrices such that $p(D) = 0$, for all $p \in \mathcal{P}$.

De Bie [10] formalized the MaxEnt model for a binary matrix D given row and column margins—also known as a Rasch [37] model. Here, we consider a more general scenario of binary data and tiles. In this case, we additionally know [Theorem 2 in 29, 55] that given a tile set $\mathcal{T}$, with $\mathcal{T}(i,j) = \{T \in \mathcal{T} \mid (i,j) \in \sigma(T)\}$, we can further factorize the maximum entropy distribution $p^*_{\mathcal{T}}$ as

$$p^*_{\mathcal{T}} = \prod_{(i,j) \in D} p^*_{\mathcal{T}}((i,j) = D(i,j)) \quad ,$$

where

$$p^*_{\mathcal{T}}((i,j) = 1) = \frac{\exp\left(\sum_{T \in \mathcal{T}(i,j)} \lambda_T\right)}{\exp\left(\sum_{T \in \mathcal{T}(i,j)} \lambda_T\right) + 1} \text{ or } 0, 1 \ .$$

This result allows us to represent the MaxEnt distribution $p^*_{\mathcal{T}}$ of binary data matrices given background information in the form of a set of tiles $\mathcal{T}$ by a product of Bernoulli random variables, each of which denotes a single entry in the data matrix D. We need to emphasize here that this model is a different MaxEnt model compared to that when independence between rows in the data matrix D is assumed [see, e.g., 35, 54, 60]. Here, for example, in the special case where the given tiles are all exact ($\gamma_T = 0$ or 1), the resulting MaxEnt distribution will have a very simple form:

$$p^*_{\mathcal{T}}((i,j) = 1) = \begin{cases} \gamma_T & \text{if } \exists T \in \mathcal{T} \text{ such that } (i,j) \in \sigma(T) \\ \frac{1}{2} & \text{otherwise.} \end{cases}$$

3.1.3 Inferring the MaxEnt Distribution. To estimate the parameters of the Bernoulli random variables mentioned above, we follow a standard approach and apply the well known Iterative Scaling (IS) algorithm [7] to infer the tile based MaxEnt model over binary data matrices. Algorithm 1 illustrates the details of this IS algorithm for binary data. Briefly speaking, for each tile $T \in \mathcal{T}$, the algorithm updates the probability distribution p such that the expected frequency of 1s under the distribution p matches the given frequency γ_T. Clearly, during this iterative update procedure, we may change the expected frequencies of other tiles, and hence several iterations are required until the probability distribution p converges. For the proof of convergence, please refer to Theorem 3.2 proved by Csiszar [6]. In practice, the algorithm typically takes on the order of seconds to converge.

ALGORITHM 1: Iterative Scaling Algorithm (binary dataset)

input : a tile set $\mathcal{T}$, target frequencies $\{\gamma_T \mid T \in \mathcal{T}\}$.
output : maximum entropy distribution $p^*_{\mathcal{T}} \leftarrow p$.

$p \leftarrow$ a N-by-M matrix with all values of $\frac{1}{2}$;
for $T \in \mathcal{T}, \gamma_T = 0, 1$ **do**
  $p(i,j) \leftarrow \gamma_T$, for all $(i,j) \in \sigma(T)$;
end
while *not converged* **do**
  for $T \in \mathcal{T}, 0 < \gamma_T < 1$ **do**
    find x such that: $fr(T;p) = \sum_{(i,j)\in\sigma(T)} \frac{x \cdot p(i,j)}{1-(1-x)\cdot p(i,j)}$;
    $p(i,j) \leftarrow \frac{x \cdot p(i,j)}{1-(1-x)\cdot p(i,j)}$, for all $(i,j) \in \sigma(T)$;
  end
end

3.2 Maximum Entropy Model for Real-valued Data

In this section, we introduce the MaxEnt model for real-valued data with tiles as background knowledge. We first extend the concept of tiles from binary transactional matrix to a real-valued transactional matrix. Then, we formulate the global MaxEnt model over the real-valued transactional data, and finally, we provide an efficient algorithm to infer the real-valued MaxEnt distribution.

3.2.1 Notation for Tiles. As stated earlier, a document-entity transactional matrix D usually contains occurrence (count) information for each entity in every document of the corpus. Count data is integer valued but without loss of generality, the entries in the real-valued transactional matrix D is considered to be normalized into the range of $[0, 1]$ (e.g. each entry of D can be divided by the maximum entry of D).

A tile T over a real-valued matrix D is still defined as the tuple $T = (r(T), c(T))$ which identifies a sub-matrix from D. Compared to the frequency of a tile defined in the binary case, more descriptive statistical measures can be defined for real-valued tiles. In our scenario, we choose the sum of the values and sum of the squared values identified by a tile T, which are represented by f_m and f_v respectively. More specifically, f_m and f_v are defined as:

$$f_m(T \mid D) = \sum_{\forall (i,j)\in\sigma(T)} D(i,j), \tag{3}$$
$$f_v(T \mid D) = \sum_{\forall (i,j)\in\sigma(T)} D^2(i,j).$$

3.2.2 Global MaxEnt Model from Tiles. A real-valued MaxEnt model was first proposed by Kontonasios et al. [28]. Here, we are given a set of real-valued tiles $\mathcal{T}$ where for every entry (i,j) in the matrix D, there exists at least a tile $T \in \mathcal{T}$ such that $(i,j) \in \sigma(T)$. Each tile $T \in \mathcal{T}$ is associated with its basic statistics, here $\tilde{f}_m(T)$ and $\tilde{f}_v(T)$ which can be computed from the given real-valued data matrix. Then, the probability distribution space of real-valued data matrices can be defined as

$$\mathcal{P} = \{p \mid \mathbb{E}_p[f_m(T \mid D)] = \tilde{f}_m(T), \mathbb{E}_p[f_v(T \mid D)] = \tilde{f}_v(T), \forall T \in \mathcal{T}\} .$$

Here, $\mathbb{E}_p[\cdot]$ represents the expectation with respect to the probability distribution p. Among all the candidate distribution $p \in \mathcal{P}$, we choose the one that maximizes the entropy, that is,

$$p^*_{\mathcal{T}} = \operatorname*{argmax}_{p \in \mathcal{P}} \left\{ -\int_D p(D) \log p(D) dD \right\} .$$

To be more specific, inferring the MaxEnt distribution could be formulated as the following optimization problem:

$$\begin{aligned} p^*_{\mathcal{T}} = \operatorname*{argmax}_{p} & \left\{ -\int_D p(D) \log p(D) dD \right\} \\ \text{s.t.} \quad & \int_D p(D) f_m(T \mid D) dD = \tilde{f}_m(T), \ \forall T \in \mathcal{T} \\ & \int_D p(D) f_v(T \mid D) dD = \tilde{f}_v(T), \ \forall T \in \mathcal{T} \\ & \int_D p(D) dD = 1, \ p(D) \geq 0. \end{aligned} \tag{4}$$

Since the optimization problem defined above is convex, by applying the approach of Lagrange multipliers, we can derive that the MaxEnt distribution has the following exponential form:

$$p^*_{\mathcal{T}}(D) = \frac{1}{Z} \exp\left(-\sum_{T \in \mathcal{T}} \lambda^{(m)}_T f_m(T \mid D) - \sum_{T \in \mathcal{T}} \lambda^{(v)}_T f_v(T \mid D) \right) .$$

Substituting $f_m(T \mid D)$ and $f_v(T \mid D)$ with their definitions from Equation (3), the MaxEnt distribution could be simplified as

$$\begin{aligned} p^*_{\mathcal{T}} &= \frac{1}{Z} \prod_{(i,j) \in D} \exp\left(-\beta_{i,j} D^2(i,j) - \alpha_{i,j} D(i,j) \right) \\ &= \prod_{(i,j) \in D} p_{i,j}(D(i,j)), \end{aligned} \tag{5}$$

where

$$\begin{aligned} p_{i,j}(D(i,j)) &= \sqrt{\frac{\beta_{i,j}}{\pi}} \exp\left\{ -\frac{\left(D(i,j) + \frac{\alpha_{i,j}}{2\beta_{i,j}} \right)^2}{1/\beta_{i,j}} \right\} \\ \alpha_{i,j} = \sum_{\substack{(i,j) \in \sigma(T) \\ T \in \mathcal{T}}} \lambda^{(m)}_T, & \quad \beta_{i,j} = \sum_{\substack{(i,j) \in \sigma(T) \\ T \in \mathcal{T}}} \lambda^{(v)}_T. \end{aligned}$$

Equation (5) indicates that the real-valued MaxEnt distribution over the matrix D could be factorized into the product of the distributions of $D(i,j)$, where each $D(i,j)$ follows the Gaussian distribution,

$$D(i,j) \sim \mathbb{N}\left(-\frac{\alpha_{i,j}}{2\beta_{i,j}}, \frac{1}{2\beta_{i,j}} \right).$$

ALGORITHM 2: MaxEnt model inference (real-valued dataset)

input : a tile set $\mathcal{T}$, target tile statistics $\{f_m(T \mid D), f_v(T \mid D) \mid T \in \mathcal{T}\}$.
output : Maximum entropy distribution $p^*_{\mathcal{T}}$ parameterized by $\alpha_{i,j}$ and $\beta_{i,j}$.

```
Initialize λ_T^(m) and λ_T^(v) randomly ∀T ∈ T;
λ ← [λ_T^(m), λ_T^(v) | T ∈ T];
while not converged do
    updateAlphaBeta(λ);
    compute gradient using Equation (6) and (7);
    perform a conjugate gradient update on λ;
end
```

In addition, we can also compute the normalizing constant Z in Equation (5) as

$$Z = \oint_D \prod_{(i,j)\in D} \exp\left(-\beta_{i,j} D^2(i,j) - \alpha_{i,j} D(i,j)\right) dD$$
$$= \prod_{(i,j)\in D} \sqrt{\frac{\pi}{\beta_{i,j}}} \exp\left(\frac{\alpha_{i,j}^2}{4\beta_{i,j}}\right).$$

3.2.3 Inferring the MaxEnt Distribution. To infer the real-valued MaxEnt distribution, we need to estimate the values of the model parameters $\lambda_T^{(m)}$ and $\lambda_T^{(v)}$. We leverage the duality between maximum entropy and maximum likelihood formulations [41] by solving

$$\max_{\boldsymbol{\lambda}} : \mathcal{L}(\boldsymbol{\lambda}) = \log p(D) = \sum_{T\in\mathcal{T}} \left(-\lambda_T^{(m)} \tilde{f}_m(T) - \lambda_T^{(v)} \tilde{f}_v(T)\right) - \log Z$$
$$= -\sum_{(i,j)\in D} \left[\frac{1}{2}\log\left(\frac{\pi}{\beta_{i,j}}\right) + \frac{\alpha_{i,j}^2}{4\beta_{i,j}}\right] + \sum_{T\in\mathcal{T}} \left(-\lambda_T^{(m)} \tilde{f}_m(T) - \lambda_T^{(v)} \tilde{f}_v(T)\right)$$
$$\text{s.t.} \quad \beta_{i,j} > 0, \quad \forall (i,j) \in D.$$

The above optimization problem is convex and can be solved efficiently by state-of-the-art optimization algorithms. Here, we choose the conjugate gradient method to solve this problem, where the gradient of the objective function $\mathcal{L}(\boldsymbol{\lambda})$ is given by

$$\frac{\partial \mathcal{L}(\boldsymbol{\lambda})}{\partial \lambda_T^{(m)}} = -\sum_{(i,j)\in\sigma(T)} \left(\frac{\alpha_{i,j}}{2\beta_{i,j}}\right) - \tilde{f}_m(T), \tag{6}$$

$$\frac{\partial \mathcal{L}(\boldsymbol{\lambda})}{\partial \lambda_T^{(v)}} = \sum_{(i,j)\in\sigma(T)} \left(\frac{1}{2\beta_{i,j}} + \frac{\alpha_{i,j}^2}{4\beta_{i,j}^2}\right) - \tilde{f}_v(T). \tag{7}$$

4 SCORING BICLUSTERS AND CHAINS

We now turn our attention to using the above formalisms to help score our patterns, viz., biclusters and bicluster chains. But before we do so, we need to pay attention to the relational schema over which these patterns are inferred, as this influences how patterns can be represented as tiles, in order to be incorporated as knowledge in our maximum entropy models.

4.1 Entity-Entity Relation Extraction

In this section, we describe the approach to construct a multi-relational schema $S(\mathcal{U}, \mathcal{R})$ from a transaction data matrix D. Recall that whenever an element $D(r, e_i)$ has a non-zero value (e.g. 1 in the binary case or a fraction in the range of $[0, 1]$ in the real-valued case), this denotes that entity e_i appears in row r of D. As an example, when considering text data, an entity would correspond to a word or concept, and a row to a document in which this word occurs. (Thus, note that when considering text data we currently model occurrences of entities at the granularity of documents. Admittedly, this is a coarse modeling in contrast to modeling occurrences at the level of sentences, but it suffices for our purposes.)

To extract entity-entity relations from transaction data matrix D, we utilize the entity co-occurrence information. To be more specific, each binary relation in $\mathcal{R}$ stores the entity co-occurrences in data matrix D between two entity domains, e.g. for each $R = R(U_i, U_j)$ in $\mathcal{R}$, $(e, f) \in R$ for $e \in U_i$, $f \in U_j$, and e and f appear at least once together in a row in D.

4.2 Background Model Definition

Next, to discover non-trivial and interesting patterns, we need to incorporate some basic information about the multi-relational schema $S(\mathcal{U}, \mathcal{R})$ into the model. For such basic background knowledge over D we use the column marginals and the row marginals for each entity domain. To this end, following Wu et al. [61] we construct a tile set $\mathcal{T}_{col}$ consisting of a tile per column, a tile set $\mathcal{T}_{row}$ consisting of a tile per row per entity domain, and a tile set $\mathcal{T}_{dom}$ consisting of a tile per entity domain but spanning all rows. Formally, we have

$$\begin{aligned} \mathcal{T}_{col} &= \{(U_D, e) \mid e \in U, U \in \mathcal{U}\}, \\ \mathcal{T}_{row} &= \{(r, U) \mid r \in U_D, U \in \mathcal{U}\}, \text{ and} \\ \mathcal{T}_{dom} &= \{(U_D, U) \mid U \in \mathcal{U}\}. \end{aligned}$$

Here, U_D represents the domain of all the documents in the dataset (e.g. the set of all rows in the data matrix D). We refer to the combination of these three tile sets as the background tile set $\mathcal{T}_{back} = \mathcal{T}_{row} \cup \mathcal{T}_{col} \cup \mathcal{T}_{dom}$. Given the background tiles $\mathcal{T}_{back}$, the background MaxEnt model p_{back} can be inferred using iterative scaling (see Sect. 3.1.3) and the conjugate gradient method (see Sect. 3.2.3) for binary and real-valued cases, respectively.

4.3 Quality Scores

To assess the quality of a given bicluster B with regard to our background knowledge, we need to first convert it into tiles such that we can infer the corresponding MaxEnt model. Below we specify how we do this conversion for biclusters from entity-entity relations. For a given bicluster $B = (E_i, E_j)$, we construct a tile set $\mathcal{T}_B$ consisting of $|E_i|\,|E_j|$ tiles as

$$\mathcal{T}_B = \{(rows(X; D),\ X) \mid X = \{e_i, e_j\} \text{ with } (e_i, e_j) \in B\} \quad , \tag{8}$$

where $rows(X; D)$ is the set of rows that contain X in D, e.g. the corresponding entries for X in the matrix D that have non-zero values.

To evaluate the quality of a bicluster chain C, for each bicluster $B \in C$, we construct the set of tiles $\mathcal{T}_B$ as illustrated by Equation (8), and the tile set that corresponds to a bicluster chain C is then $\mathcal{T}_C = \bigcup_{B \in C} \mathcal{T}_B$.

Next, we describe the metrics that measure how much information a bicluster B (or the corresponding tile set $\mathcal{T}_B$) gives with regard to the background model p_{back}. Motivated

by De Bie [9], the global score is defined as

$$s_{global}(B) = KL(p_B||p_{back}) \ , \tag{9}$$

where p_B represents the MaxEnt distribution inferred over the background tile set $\mathcal{T}_{back}$ and the tile set $\mathcal{T}_B$ for the bicluster B.

For both of binary and real-valued MaxEnt model, the MaxEnt distribution $p(D)$ can be factorized as

$$p(D) = \prod_{(i,j)\in D} p(D(i,j)) \ .$$

Thus, this global score can be written as

$$\begin{aligned} s_{global}(B) &= \oint_D p_B(D) \log \frac{p_B(D)}{p_{back}(D)} dD \\ &= \oint_D \prod_{(i,j)\in D} p_B(D(i,j)) \sum_{(i,j)\in D} \log \frac{p_B(D(i,j))}{p_{back}(D(i,j))} dD \\ &= \sum_{(i,j)\in D} \int_{-\infty}^{+\infty} p_B(D_{i,j}) \log \frac{p_B(D(i,j))}{p_{back}(D(i,j))} dD(i,j) \\ &= \sum_{(i,j)\in D} KL(p_B(D(i,j))||p_{back}(D(i,j))) \ . \end{aligned} \tag{10}$$

For the binary MaxEnt model, $D(i,j)$ follows the Bernoulli distribution

$$D(i,j) \sim Bernoulli(q), \text{ where } q = \frac{\exp\left(\sum_{T\in\mathcal{T}(i,j)} \lambda_T\right)}{\exp\left(\sum_{T\in\mathcal{T}(i,j)} \lambda_T\right) + 1} \ ,$$

and the global score for binary MaxEnt model would be

$$s_{global}(B) = \sum_{(i,j)\in D} \left(q_B \log \frac{q_B}{q_{back}} + (1-q_B) \log \frac{1-q_B}{1-q_{back}} \right) \ .$$

For the real-valued MaxEnt model, $D(i,j)$ follows the Gaussian distribution

$$D(i,j) \sim \mathbb{N}\left(-\frac{\alpha_{i,j}}{2\beta_{i,j}}, \frac{1}{2\beta_{i,j}} \right) \ .$$

Given any two normal distribution $P_{\mathcal{N}_1} = \mathcal{N}(\mu_1, \sigma_1^2)$ and $P_{\mathcal{N}_2} = \mathcal{N}(\mu_2, \sigma_2^2)$, we can verify that the KL-divergence between these two normal distribution is

$$KL(P_{\mathcal{N}_1}||P_{\mathcal{N}_2}) = \log \frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2} \ . \tag{11}$$

Combining Equation (10) and (11), the global score for the real-valued maximum entropy model is

$$s_{global} = \sum_{(i,j)\in D} \left[\frac{1}{2} \log \frac{\beta_{i,j}^{(B)}}{\beta_{i,j}^{(back)}} + \frac{\beta_{i,j}^{(back)}}{2\beta_{i,j}^{(B)}} + \beta_{i,j}^{(back)} \left(\frac{\alpha_{i,j}^{(back)}}{2\beta_{i,j}^{(back)}} - \frac{\alpha_{i,j}^{(B)}}{2\beta_{i,j}^{(B)}} \right)^2 - \frac{1}{2} \right] . \tag{12}$$

However, using the global score defined above requires us to re-infer the MaxEnt model for every candidate bicluster that needs to be evaluated, which could be computationally expensive and thus not applicable to our interactive mining sitting. Moreover, s_{global}

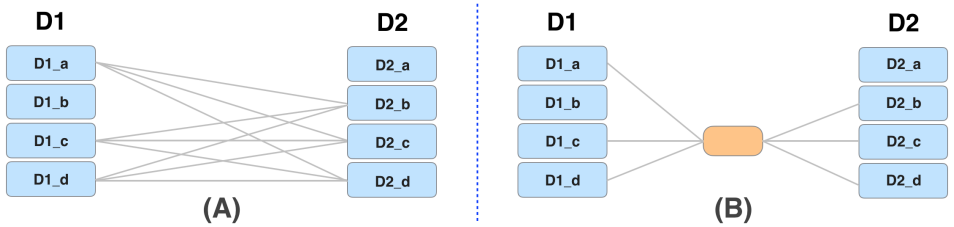

Fig. 3. Visual representations of a bicluster that includes three entities in *D1* and three entities in *D2*. (A) displays all individual relationships between the two sets of entities from the two domains, *D1* and *D2*. (B) Realtionships are aggregated as an edge bundle that represents a bicluster.

evaluates a candidate globally, whereas typically most information is *local*: at most a few entries in the maximum entropy distribution will be affected by adding B into the model. Making use of this observation and considering the ease of computation, to reduce the computational cost of candidate bicluster evaluation, we define the score $s_{local}(B)$ that measures the local surprisingness of a tile set as

$$s_{local}(B) = -\sum_{T \in \mathcal{T}_B} \sum_{(i,j) \in \sigma(T)} \log p_{back}(D(i,j)) \ , \tag{13}$$

which is an approximation of the local negative log-likelihood of the bicluster B. For both binary and real-valued MaxEnt model, $p_{back}(D(i,j))$ indicates the probability (or probability density) evaluated at the value $D(i,j)$ under the current background MaxEnt model. Notice that although the global and local scores are described using the notation of biclusters here, they can also be directly adopted to assess the quality of bicluster chains because fundamentally these scores are defined around the concept of tiles and bicluster chains (and can thus be trivially converted to a set of tiles as described at the beginning of this section).

5 MERCER

MERCER is a visual analytics system, supported by the maximum entropy model above, to support interactive exploration of coordinated relationships using biclusters. Coordinated relationships are groups of relations, connecting sets of entities from different domains (e.g., people, location, organization, etc.), which potentially indicate coalitions between these entities. MERCER extends a recently proposed bicluster visualization, BiSet [51], by incorporating MaxEnt models to support user exploration of entity coalitions for sensemaking purposes. In this section, we first briefly introduce BiSet, followed by the enhancements that MERCER provides.

5.1 BiSet Technique Overview

The key idea is that *BiSet visualizes the mined biclusters in context as edge bundles* between sets of related entities. BiSet uses lists as the basic layout to present entities and biclusters. Figure 3 shows an example of a visualized bicluster in BiSet. In Figure 3, (A) shows all individual edges between related entities and (B) presents the same bicluster as an edge bundle. BiSet enables both ways to show the coalition of entities with two modes: *link mode* and *bicluster mode*. *Link mode* displays the individual connections among entities in a dataset, while *bicluster mode* offers a more clear representation to show identified biclusters in the dataset. Based on these visual representations, BiSet can visually show

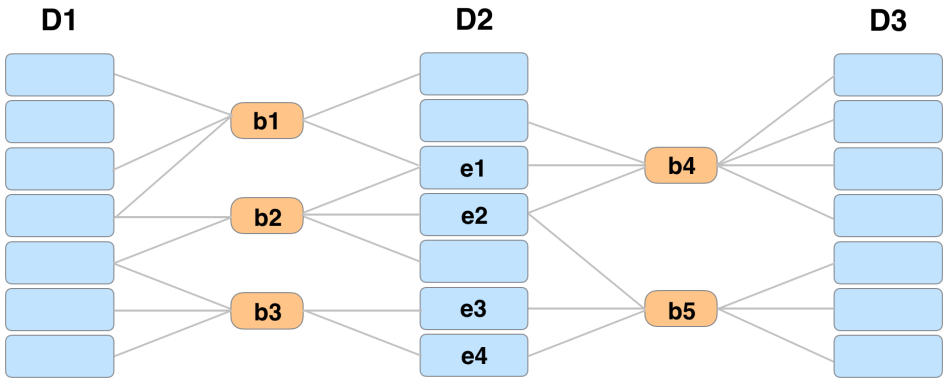

Fig. 4. An example of four bicluster-chains (*b1 - b4*, *b2 - b4*, *b2 - b5* and *b3 - b5*). These chains consist of entities from three domains, *D1*, *D2* and *D3*. *b1* and *b4* are connected through *e1*. *b2* and *b4* share *e1* and *e2*. *b2* and *b5* are linked by *e3*. *b3* and *b5* are connected by *e3* and *e5*.

bicluster-chains as connected edge bundles through their shared entities. Figure 4 shows four bicluster-chains (*b1 - b4*, *b2 - b4*, *b2 - b5* and *b3 - b5*) visualized using BiSet. Each of them consists of two different biclusters including entities from three domains. The two biclusters in each chain are visually connected through one or two shared entities. For example, bicluster *b2* and *b4* are connected by entity *e1* and *e2*. With edges, BiSet enables users to see members of bicluster-chains and how these biclusters are connected. This potentially guides users to interpret the coalition among sets of entities from multiple domains in an organized manner (e.g., checking connected biclusters from left to right).

To support exploratory analysis, BiSet treats *edge bundles as first class objects*, so users can directly manipulate them (e.g., drag and move) to spatially organize them in meaningful ways. BiSet also offers automatic ordering for entities and biclusters to help users organize them. For example, entities can be ordered based on their frequency in a dataset and biclusters can be ordered by size (i.e., the number of entities participating in a bicluster). Moreover, BiSet can highlight bicluster-chains as users select their members (e.g., entities and biclusters). This provides visual clues for users to follow in conducting their analysis.

5.2 Adaptions from BiSet to MERCER

Key adaptions, from BiSet to MERCER, lie in two levels: *representation-level* (specifically visual encoding), and *interaction-level*, (human-model interaction, in particular). MERCER shares the basic visual encodings in shape and size (to represent entities, biclusters and edges) with BiSet, but it introduces *surprisingness oriented highlighting*, which is not included in BiSet. Detailed visual encodings in MERCER is discussed in Section 5.3. This *surprisingness oriented highlighting* aims at supporting *human-model interactions* in MERCER. The capability of enabling *human-model interactions* is the essential difference between BiSet and MERCER. This capability helps to address recently identified usability challenges of using biclusters for sensemaking (e.g., bicluster evaluation and prioritization) [49, 52]. Without human-model interactions, in BiSet, users have to check biclusters or bicluster-chains (based on connection oriented highlighting) and manually figure out potentially useful ones. This may take much cognitive effort, especially when data is large. However, in MERCER, users

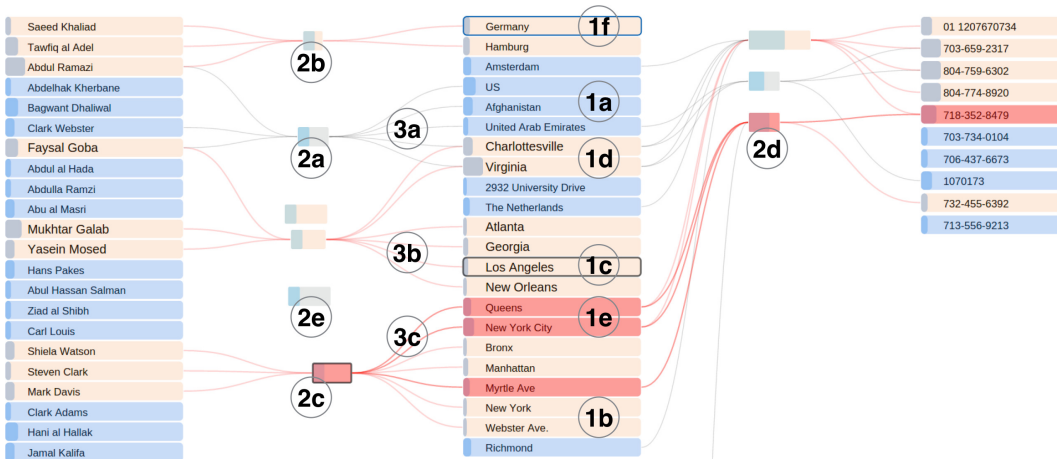

Fig. 5. Detailed visual encodings in MERCER. *1a*, *2a* and *3a* depict the *normal* state of an entity, a bicluster and edges, respectively. *1b*, *2b* and *3b* depict the *connection-oriented* highlighting state of an entity, a bicluster and edges, when users *select* bicluster *2c*, *hover* over entity *1f* and *select* entity *1c*. *1e*, *2d* and *3c* illustrate the *surprisingness-oriented* highlighting state of an entity, a bicluster and edges. *1d* demonstrates larger fonts of entities as users hover the mouse pointer over their previously selected entity *1c*. Moreover, *2e* represents a bicluster (in the *normal* state) with its edges chosen to be hidden by users.

can explicitly request computation to help them find potentially useful biclusters or bicluster-chains, by directly interacting with a bicluster. Moreover, by revealing the *surprisingness oriented highlighting*, MERCER helps to prioritize biclusters and bicluster-chains to support user explorations. The *human-model interaction* capability and evaluation strategies are discussed in Section 5.4 and Section 5.5, respectively.

5.3 MERCER Visual Encoding

5.3.1 Shape and Size. In MERCER, entities and biclusters are represented as rectangles (e.g., *1a* and *2a* in Figure 5), and edges are visualized as Bézier curves. We use Bézier curves because they can generate more smooth edges, compared with polylines [32]. Rectangles indicating entities are equal in length, while those representing biclusters are not. MERCER applies a linear mapping function to determine the length of a bundle based on the total number of its related entities. In a bicluster rectangle, MERCER uses two colored regions (light green and light gray) to indicate the proportion between its related entities in lists of both sides (left and right). In an entity rectangle, a small rectangle is displayed on the left to indicate its frequency in a dataset. The length of these rectangles is determined by the frequency of the associated entities with a linear mapping function. These helps users to visually discriminate entities from biclusters. Moreover, when users hover over a selected entity or bicluster (e.g., entity *1c* and bicluster *2c* in Figure 5), the font of its related entities is enlarged (e.g., comparing *1d* with *1b* in Figure 5). This helps users review relevant information of their previous selections.

5.3.2 Color Coding. MERCER applies color coding to entities, biclusters and edges to indicate their states and allows users to hide edges of biclusters to reduce visual clutter (see *2e* in Figure 5). In MERCER, entities, biclusters and edges have two basic states: *normal*

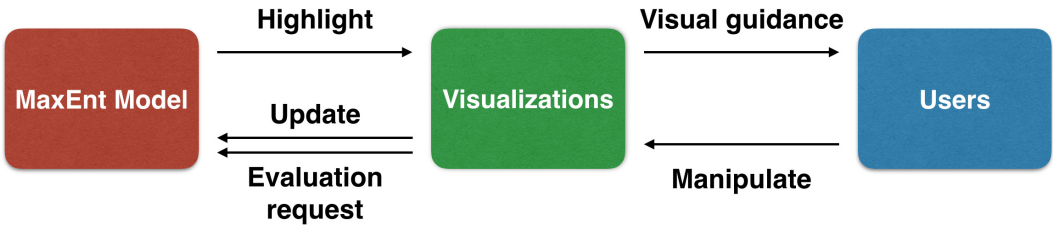

Fig. 6. The human-model interaction flow in MERCER. Visual representations in MERCER enable the interaction between users and the proposed maximum entropy models.

and *highlighted*. The normal state is the default state for entities, biclusters and edges. Examples of the normal state for them are shown as *1a*, *2a* and *3a*, respectively, in Figure 5. To encode surprisingness, MERCER supports two types of highlighting states: *connection oriented highlighting* (colored as *orange* in Figure 5) and *surprisingness oriented highlighting* (color as *red* in Figure 5), which encode two levels of information: *the coalition of entities* and *the surprisingness of the coalition*. The former indicates the linkage of entities, emphasizing the connections between entities. The latter reveals the model-evaluated surprisingness of different sets of entity coalitions. In Figure 5, examples of connection-oriented highlighting for entities, biclusters and edges are shown as *1b*, *2b* and *3b*, respectively; while examples of surprisingness-oriented highlighting are presented as *1e*, *2d* and *3c*.

The connection oriented highlighting state is triggered as users hover or select an entity or a bicluster. For example, when users hover the mouse pointer over the entity *1f*, its directly connected bicluster *2b* is highlighted and other entities that belong to this bicluster are also highlighted. The surprisingness oriented highlighting state is triggered by explicit user request of model evaluation. For instance, in Figure 5, as users request to find the most surprising chains with bicluster *2c* as the starting point, MERCER highlights entities and biclusters in a chain that has the highest score given by the proposed *maximum entropy model* (the approach to discover such a chain will be described in Section 5.5 below). With our color codings, MERCER empowers users to explore entity coalitions by directing them to computationally identified surprising chains.

5.4 Human-model Interaction

MERCER allows human-model interaction with visualizations to support visual analytics of entity coalitions. To enable this capability, we incorporate the proposed *maximum entropy models* into MERCER. Figure 6 illustrates the human-model interaction flow in MERCER. Visual representations in MERCER work as the bridge to enable the interaction between users and the proposed models. After inspecting the visualized biclusters and bicluster-chains, users can explicitly request model evaluations using right click menus on a bicluster. This further triggers the maximum entropy model to evaluate either all paths passing through the requested bicluster or its neighboring biclusters. Then, based on results of the model evaluation, MERCER highlights the most surprising bicluster-chain including the user requested bicluster or neighboring biclusters. We address this with a detailed discussion in Section 5.5. Moreover, users can mark highlighted bicluster(s), based on model evaluation, as useful one(s) by using a right click menu on the bicluster(s). This implicitly evokes a model update function, which informs the model that the information in a marked bicluster has been known by users. Then the model updates its background information to take the

marked bicluster(s) into account and prepare for further user requested evaluations. This human-model interaction flow in MERCER enables the combination the human cognition with computations for the exploration of entity coalitions.

5.5 Model Evaluation Strategies

MERCER offers two strategies to evaluate bicluster-chains, using the proposed maximum entropy models, based on explicit user requests: *full path evaluation* and *stepwise evaluation*. Both ways require users to explicitly specify a bicluster based on its visual information, e.g. size of a bicluster, frequency of corresponding entities, etc., to initiate the chain. The former evaluates all bicluster-chains passing through the bicluster that users request for evaluation, while the latter evaluates neighboring biclusters that satisfy a certain degree of overlap with the user-specified one. MERCER enables users to explicitly issue an evaluation request from a bicluster with a right click menu. From the menu, users can choose the desired way of evaluation.

5.5.1 Full Path Evaluation. The *full path evaluation* in MERCER includes three key steps: 1) path search, 2) path evaluation, and 3) path rank. In MERCER, a path, passing through a bicluster, refers to a set of biclusters (e.g., $\{b2, b4\}$ in Figure 4), which can be connected through certain entities to form a bicluster-chain. In the *path search* step, MERCER finds all possible paths passing through the bicluster that users request for evaluation. Similar to tree search, MERCER treats the user requested bicluster as a root node and applies depth-first search to find all paths starting from this bicluster. If the user requested bicluster is not from the left or right most relation in the user specified multi-relational schema, MERCER performs bidirectional search and then combines identified paths in the left and those in the right together to obtain all paths going through this bicluster. Then in the *path evaluation* step, MERCER converts each bicluster-chain, found in the previous step, into a unique set of tiles following the Equation (8) in Section 4.3, and applies the maximum entropy models to score them. Finally, based on the score from the model, in the *path rank* step, MERCER ranks these bicluster-chains and visually highlights the one that has the highest score (e.g., $\{2c, 2d\}$ in Figure 5). Thus, with the *full path evaluation* in MERCER, users can get the most surprising bicluster-chain for the bicluster requested for evaluation.

5.5.2 Stepwise Evaluation. The *stepwise evaluation* in MERCER examines neighboring biclusters for the one that users request for evaluation. *Neighboring biclusters* for a specific bicluster refers to those that can meet certain degree of overlaps, with respect to participated entities, with a user requested bicluster. MERCER uses the Jaccard coefficient to measure the degree of overlaps between two biclusters with a default threshold set as 0.1. Thus, for a specific bicluster, its potential neighboring biclusters are those sharing at least one domain (e.g., people, location, date, etc.) with this one.

Similar to the *full path evaluation*, the *stepwise evaluation* also has three key steps, including: 1) neighboring bicluster search, 2) neighboring bicluster evaluation, and 3) neighboring bicluster coloring. Based on a user specified bicluster for evaluation, MERCER first identifies its neighboring biclusters using the Jaccard coefficient. Then, MERCER converts the identified neighboring biclusters into different sets of tiles following Equation (8) and employs the maximum entropy models to score them. Based on the model evaluation score, BiSet applies a linear mapping function to assign the opacity value of *surprisingness* oriented highlighting color to these biclusters. The more red a color is, and the higher score this neighboring bicluster gets, which indicates more surprising information. Figure 7

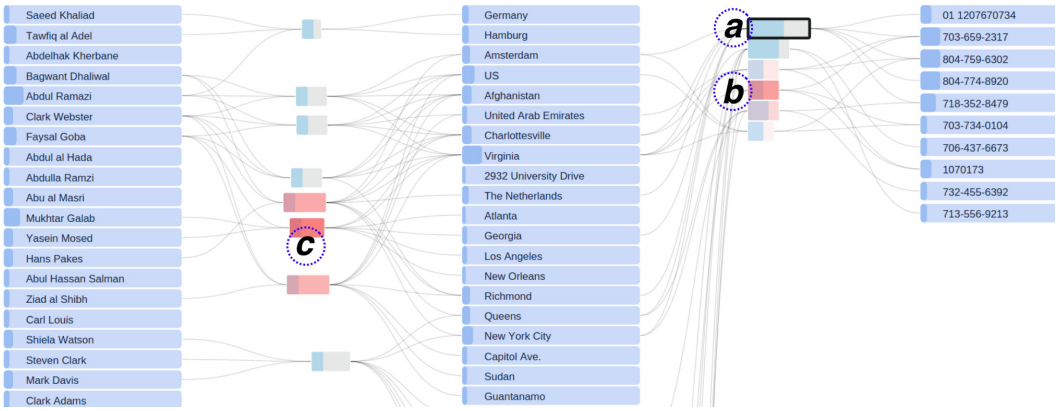

Fig. 7. Exampled results from the *stepwise* evaluation in MERCER. (a) shows the bicluster selected by a user to initiate the maximum entropy model evaluation. (b) represents the most surprising bicluster in the same bicluster list as the one requested for evaluation. (c) illustrates the most surprising bicluster in another bicluster list.

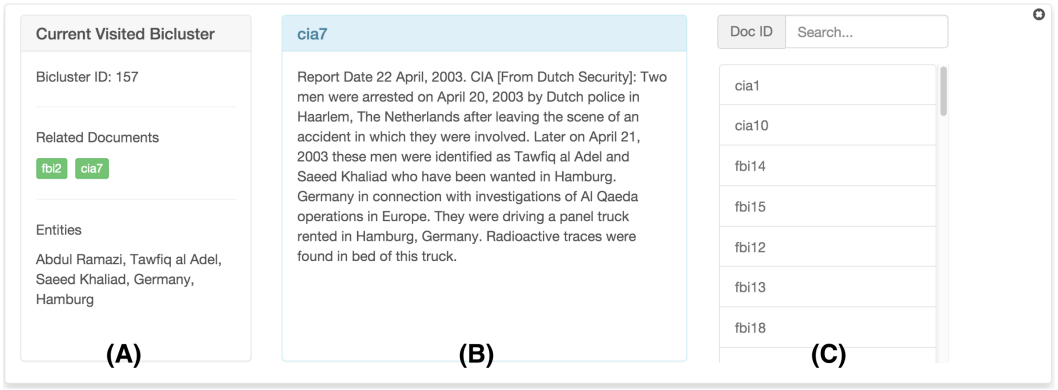

Fig. 8. Document view mode in MERCER. (A) depicts the bicluster ID, relevant document ID(s) and associated entities. (B) shows the content of a document. (C) lists all document IDs in a dataset with a search function.

gives an example of the *stepwise evaluation* in MERCER. In this example, users request to evaluate a bicluster (see a), MERCER highlights neighboring biclusters based on their model evaluation scores. Of these highlighted biclusters, bicluster b shows the most surprising one in the same bicluster list as that requested for evaluation, and bicluster c is the most surprising bicluster in the adjacent bicluster list. Although bicluster b here could not be used to extend the users selected bicluster a, it has the potential to reveal entities related to the bicluster a and the plots. Thus, we also take the most surprising bicluster from the same relation of the users selected bicluster into account. Such *stepwise evaluation* potentially enables to involve users in the process of building a meaningful bicluster-chain. Each time after a *stepwise evaluation*, users can investigate highlighted neighboring biclusters, identify and then select useful one(s) for further exploration. Users can iterate this process and build a bicluster-chain that is meaningful for them.

5.6 Bicluster based Evidence Retrieval

In MERCER, users can directly retrieve related documents from biclusters by using a right click menu. Users can use a right click menu to open a popup view, where relevant documents are listed, as is shown in Figure 8. This helps users review information relevant to this bicluster and verify computationally identified coalitions of entities. This document view is on top of the relationship exploration view with transparency, so users can simultaneously see both the visualized relationships and corresponding documents. Moreover, after reading documents, users can quickly return to the relationship exploration view by closing it.

6 EXPERIMENTS

We describe the experimental results over both synthetic and real datasets. For real datasets, we focus primarily on datasets from the domain of intelligence analysis. Through a case study, we demonstrate how the proposed maximum entropy models embedded in our visual analytics approach helps analysts to explore text datasets, such as used in intelligence analysis. All experiments described in this section were conducted on a Xeon 2.4GHz machine with 1TB memory. Performance results (for synthetic data) were obtained by averaging over 10 independent runs.

6.1 Results on Synthetic Data

To evaluate the runtime performance of the proposed maximum entropy models with respect to the data characteristics, we generate synthetic datasets. Since we focused on the runtime performance of the proposed models here, and the multi-relational schema of the dataset will not affect how the proposed models are inferred over the data matrix D, we will temporarily ignore the multi-relational schema of the dataset in the synthetic data for now. The synthetic datasets are parameterized as follows. The data matrix D consists of N rows and M columns, or entities, and β denotes the density of the data matrix D. For each entry in the data matrix D, we set its value to be non-zero with probability β. For the binary case, the non-zero values would naturally be one, and for the real-valued case, the non-zero values are generated from a standard uniform distribution. In order to avoid the scenario that too many rows or columns in D contains only zeros, a non-zero value is placed randomly in a row or column if it only contains zeros.

In our experiments, we explore data matrix D sizes of $(N = 1000, M = 1000)$, $(N = 2000, M = 2000)$, and $(N = 3000, M = 3000)$, and varied the density β of the data matrix D from 0.01 to 0.05 in steps of 0.01. To infer the maximum entropy models, we use column margin and row margin tiles as the set of constraint tiles for the proposed model (see Sect. 3). We first investigate the time needed to infer the maximum entropy models. Figure 9 shows the model inference time for the binary and real-valued maximum entropy formulations. As expected, model inference increases with dataset size and requires more time for the real-valued model. Since the real-valued maximum entropy model adopts the conjugate gradient method, model inference time heavily depends upon the structure of the given dataset, the number of constraint tiles, and how fast the model converges to the optimal solution along the gradient direction. For example, in our experiments we used the row and column margin tiles as the constraints for the real-valued maximum entropy model, the dimension of the gradient could be $2(M + N)$ (that would be 4,000 dimension when $N = 1000, M = 1000$ for our synthetic datasets).

Another interesting phenomenon we observed here is that as the density β of the data matrix D increases, the inference time required by the real-valued maximum entropy model

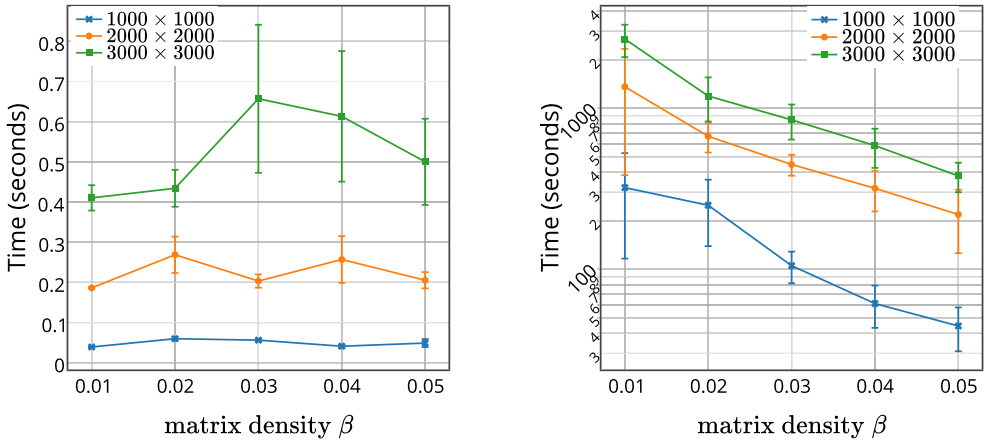

Fig. 9. Time to infer the binary (left) and real-valued (right, Y-axis is in log scale) maximum entropy model on synthetic datasets. The error bars represent the standard deviation

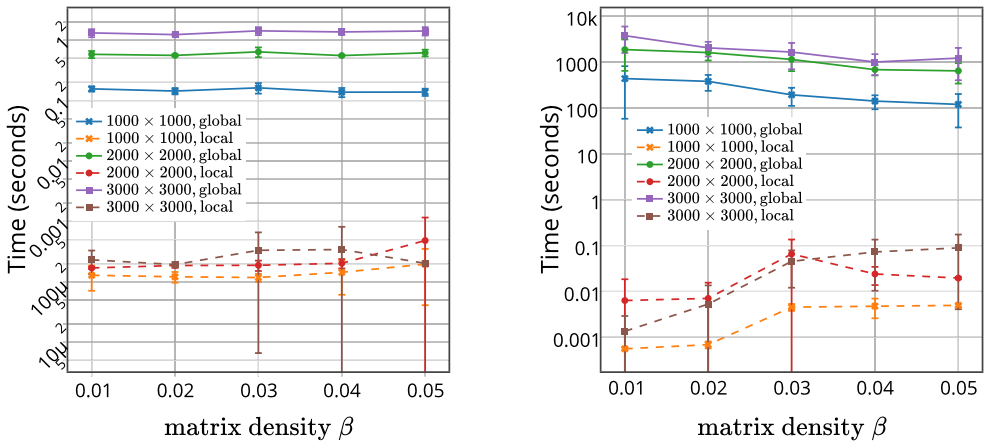

Fig. 10. Time to evaluate a set of tiles with the binary (left) and real-valued (right) maximum entropy model on synthetic datasets. The set of solid lines on the top represents the results of global score, and the set of dash lines at the bottom represents the results of local score. The error bars represent the standard deviation, and the Y-axis is in log scale.

decreases. One explanation for this phenomenon is that denser data matrices provide more information to the maximum entropy model about the underlying data generation distribution through the constraint tiles. This aids the model in rapidly learning the structure of the data space and search for the optimal solution with fewer iterations of the conjugate gradient algorithm.

We also measured the runtime performance of evaluating tile sets with the proposed binary and real-valued maximum entropy models since the patterns (biclusters or bicluster chains) whose qualities we would like to assess will eventually be converted into a set of tiles in our framework. To be more specific, we randomly generated a set of tiles over the synthetic data matrix, and compared the time required to evaluate this tile set with both global score and local score using converged binary and real-valued models, and Figure 10 illustrates the results. As we can see from this figure, in both binary and real-valued maximum entropy model, evaluating tile sets using the global score requires more time than the local score, which is expected since the global score requires a complete re-inference of the model. The difference of runtime performance between global and local scores is significant in the real-valued model due to this model inference step. When applying the real-valued maximum entropy model in practical applications, such as the one here necessitating real-time interaction, we can employ an asynchronized model inference scheme, e.g. creating a daemon process to infer the model when the system is idle, and adopt the local score to evaluate tile sets.

6.2 Evaluation on Real Dataset: A Usage Scenario

In this section, we walk through an intelligence analysis scenario to demonstrate how MERCER, particularly incorporating the proposed maximum entropy models for identifying surprising entity coalitions, can support an analyst to discover a coordinated activity via visual analysis of entity coalitions. For ease of description, we use a small dataset, viz. *The Sign of the Crescent* [22], which includes 41 fictional intelligence reports about three coordinated terrorist plots in three cities. Each plot involves at least four suspicious people. 24 of these reports are relevant to the plots. We use LCM [58] to identify *closed* biclusters from this dataset with the *minimum support* parameter set to 3. This assures that each bicluster has at least three entities from one domain. This generates 337 biclusters from 284 unique entities and 495 individual relationships (based on entity co-occurrence in the reports).

In order to try to discover all the possible plots hidden in the *Crescent* dataset, in MERCER, we set the threshold for the Jaccard coefficient as 0.05, which is a loose constraint. This enables the model to evaluate those neighboring biclusters that has a few entity overlaps with user specified biclusters for assessment. Although MERCER fully supports pattern evaluations with the real-valued *maximum entropy model*, we observed that the model evaluation results of a given bicluster were similar when using the binary and the real-valued *maximum entropy models* in our experiments over the *Crescent* dataset. Thus, we only present the use case study using the binary *maximum entropy model* here to demonstrate the effectiveness of the proposed MERCER technique when assisting analysts in conducting intelligence analysis tasks.

To illustrate the benefits of integrating the maximum entropy models into visual analytic tools, in this intelligence analysis scenario, we use BiSet [51] as the baseline approach for comparison purposes. Notice that BiSet does not has the capability of model evaluations, and thus it just provides the *connection* oriented highlighting function for users to manually explore entity coalitions. We begin our discussions with the use case of BiSet, and then discuss the use case of MERCER.

In our scenario, suppose that Linda is an intelligence analyst. She has a task to read intelligence reports and identify potential terrorist threats and key persons from the *Crescent* dataset. She opens BiSet, picks four identified domains (people, location, phone number and

date), and starts her analysis. Figure 11 to Figure 16 demonstrate Linda's key analytical steps using BiSet. Figure 17 and Figure 18 show the key steps of Linda's analytical process using MERCER. The BiSet use case and Figure 17 are informed by the previous publication of the BiSet technique [51].

6.2.1 BiSet Use Case. Linda starts analysis by checking people's names. When she hovers the mouse over an entity, BiSet highlights its related bundles and entities. Immediately she notices that *A. Ramazi* is active in three bundles. This indicates that he may be involved in three coordinated activities. Linda selects it (Figure 11) to focus on the highlighted entities of these bundles. She finds that *A. Ramazi* is involved in two cells with five people (*S. Khallad*, *T. al Adel*, *B. Dhaliwal*, *C. Webster* and *F. Goba*). One cell is in Germany and the other cell is more broadly located in four countries. *A. Ramazi* is the only person connecting the two cells. Moreover, two overlapped groups of people (sharing *A. Ramazi* and *C. Webster*) are involved in the broader cell, and each group has its unique person, *B. Dhaliwal* and *F. Goba*, respectively.

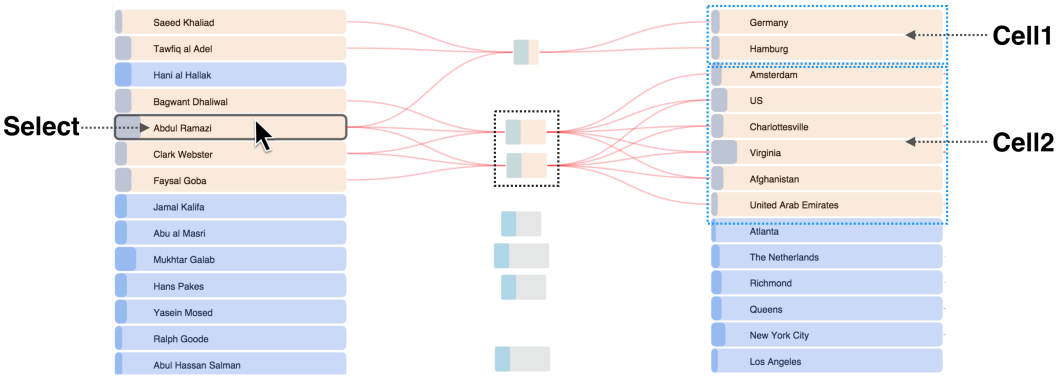

Fig. 11. Selecting *A. Ramazi* and finding that there are two similar bundles and two cells.

Then Linda decides to investigate the two overlapped groups, since she aims to know what brings the unique people to them. She checks *B. Dhaliwal* first. After hovering the mouse over it, two bundles are highlighted. Following their edges, Linda finds that two people's names (*B. Dhaliwal* and *C. Webster*) and four locations (*Charlottesville*, *Virginia*, *Afghanistan* and *Richmond*) are shared by them, and the bigger one is connected with a new name, *H. Pakes* (see Figure 12). Then she checks *F. Goba* in the same way. This time three names (*M. Galab*, *Y. Mosed* and *Z. al Shibh*) and three bundles are highlighted, and one name, *M. Galab*, has a high frequency (see Figure 13).

Linda quickly notices this, so she decides to temporarily pause her analysis of *B. Dhaliwal*, and moves on with *F. Goba*. Linda hovers the mouse over *M. Galab* to check what additional information it can lead to. However, no additional bundles or names are highlighted. Linda realizes that people potentially connected with *M. Galab* have already been highlighted in the current view. The bundle (shown in Figure 13 as the black box in the middle) reveals two people (*F. Goba* and *Y. Mosed*) related with *M. Galab*, and all their activities are in the US, including *Charlottesville*, *Virginia*, *Atlanta*, *Los Angeles*, *New Orleans* and *Georgia*. Linda get this key insight based on the group of locations in this bundle. The relations from this bundle are important, and Linda hypothesizes that the three people (*M. Galab*, *Y.*

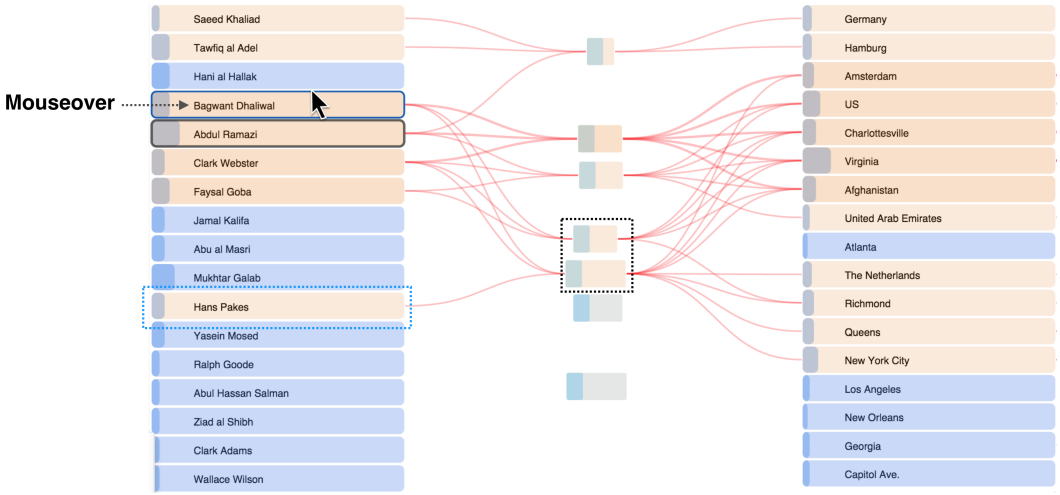

Fig. 12. When hovering the mouse over *B. Dhaliwal*, one name and two bundles are highlighted.

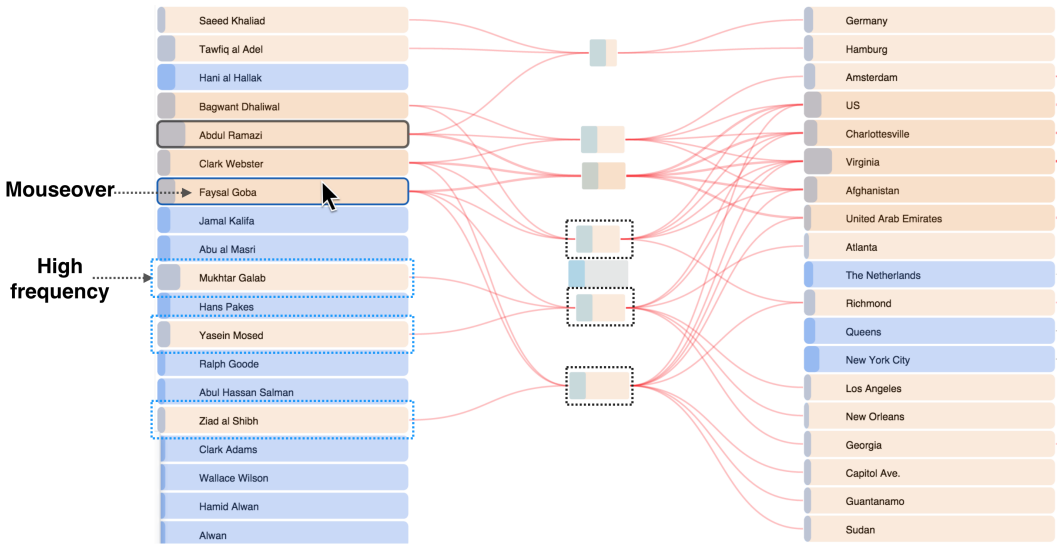

Fig. 13. When exploring *F. Goba*, three names and three bundles are highlighted.

Mosed and *Z. al Shibh*) may work on something together in the US. Thus, by following this tail [25], she wants to find more related information.

After Linda selects this useful bundle, BiSet highlights its related bundles that can form bicluster chains. Five bundles, between the *location* list and the *phone number* list, are highlighted (Figure 14), and two bundles, between the *phone number* list and the *date* list, are highlighted (Figure 15). Relevant entities in these lists are also highlighted. For these newly highlighted bundles, Linda finds that there are two big ones (relatively longer shown in Figure 14 and Figure 15). These two bundles seem useful since they have more relations. Linda decides to check them and find how bundles from different relationship lists

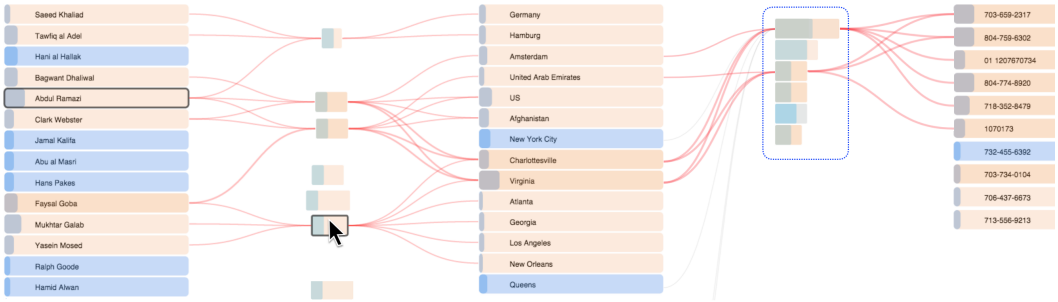

Fig. 14. After selecting a useful bundle, five bundles (in-between the list of *location* and *phone number*) are highlighted. Checking connected entities of the first and the third bundles.

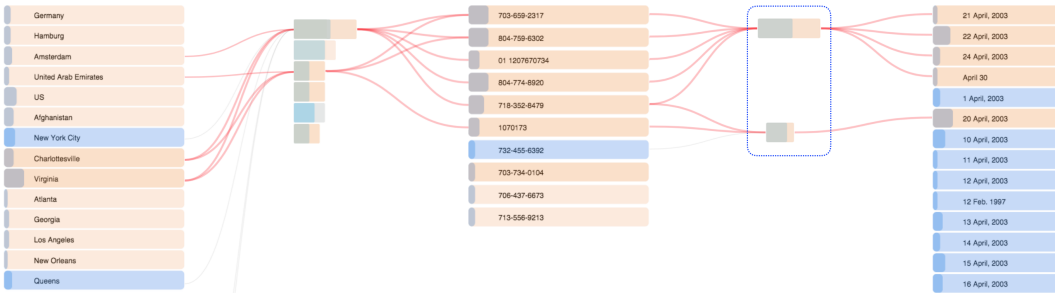

Fig. 15. After selecting a useful bundle, two bundles (in-between the list of *phone number* and *date*) are highlighted.

are connected. For bundles between the *location* list and the *phone number* list (from top to bottom in Figure 14), Linda finds that the first and the third bundle share two locations (*Charlottesville* and *Virginia*) with the selected bundle, and other highlighted bundles just share one location with the selected one. Compared with the first bundle, the third one is related with less locations that are not associated the selected bundle. Linda chooses to focus on information highly connected with the selected bundle, instead of additional information, so she considers the third bundle a useful one. Using a similar strategy in another bicluster list (between the *phone number* list and the *date* list), she finds that the bigger bundle (the top listed one in Figure 15) is more useful.

After this, Linda uses the right click menu to hide edges of other bundles for creating a clear view (see Figure 16). Then, in her workspace, there are three bundles connecting with each other through two shared locations (*Charlottesville* and *Virginia*) and three shared phone numbers (*703-659-2317* and *804-759-6302* and *1070173*). Linda feels that she has found a good number of relations, connecting four groups of entities, which may reveal a suspicious activity. Therefore, she decides to read relevant documents to collect details about these connections and make her hypothesis.

These three connected bundles direct Linda to eight reports, and all of them are relevant to the plot. By referring to the entities with bright shading in the four connected groups (shown in Figure 16), Linda reads these reports. The darker shading indicates that an entity is shared more times. This information helps to direct her attention to more important

entities in the reports. After reading these reports, she identifies four key persons involved in a potential threat 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*.

In this use case, Linda has to manually check details about shared entities to determine which biclusters are meaningful and useful, because BiSet does not provide the function of model based bicluster or chain evaluation. With just *connection* oriented highlighting, Linda has to verify many connected biclusters to find potentially useful ones. This limits her analysis strategy as stepwise search, and such search focuses on checking the shared entities of investigated biclusters. Thus, it takes Linda significant effort to work at the entity-level information to identify a meaningful bicluster chain.

6.2.2 MERCER Use Case. Similar to the previous case, Linda begins analysis by hovering individual entities in the list of people. MERCER highlights related bundles and entities as she hovers the mouse over an entity. Immediately she finds that *A. Ramazi* is active in three bundles (Figure 17 (1)), which indicates that this person is involved in three coordinated activities. Based on edges, Linda finds that two bundles are similar (see the black dotted box in Figure 17 (1)) due to the number of their shared entities. Thus, she decides to further investigate them.

With the right click menu on the two bundles, Linda uses the *stepwise* evaluation function, provided by MERCER, to find their neighboring bundles that contain the most surprising information (Figure 17 (2) and (3)). Based on evaluated scores from the *maximum entropy model*, MERCER highlights their most surprising neighboring bundles. She finds that the most surprising bundles connected with the two investigated bundles are the same. This indicates that the model-suggested most surprising bundle may be important and worthy of further inspection, and so Linda decides to find more relevant information from it.

Linda chooses the *full path* evaluation function on this model-suggested bundle to find the most surprising bicluster-chain. MERCER highlights the path (Figure 17 (4)) passing through this bundle having the highest evaluation score from the maximum entropy model. This provides four connected sets of entities from all the selected domains (people, location, phone and date). Linda feels that she has discovered enough information for a story, so she checks entities involved in this chain and reads documents from the three connected bundles. The three bundles directs Linda to nine reports in total, and eight of them are relevant to each other. After reading these relevant reports, she identifies a potential threat with four key persons 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*.

Linda is satisfied with this finding and marks the bundles in this model suggested chain as useful, using the right click menu. This informs the integrated maximum entropy model in MERCER that the information in these bundles has been known to the analyst, and so the model updates its background information for further evaluations.

The content of one report, from the bundle in the middle of the surprising chain (*a* in Figure 17 (4)), is irrelevant to that of the other eight, but the entities extracted from this report are connected with those in the identified threat. Thus, Linda considers the information in this report as potentially useful clues, which may lead to some other threat plot(s). In order to check what new information it can bring in, she uses the *full path* evaluation function on the bundle in the middle of the surprising chain (*a* in Figure 17

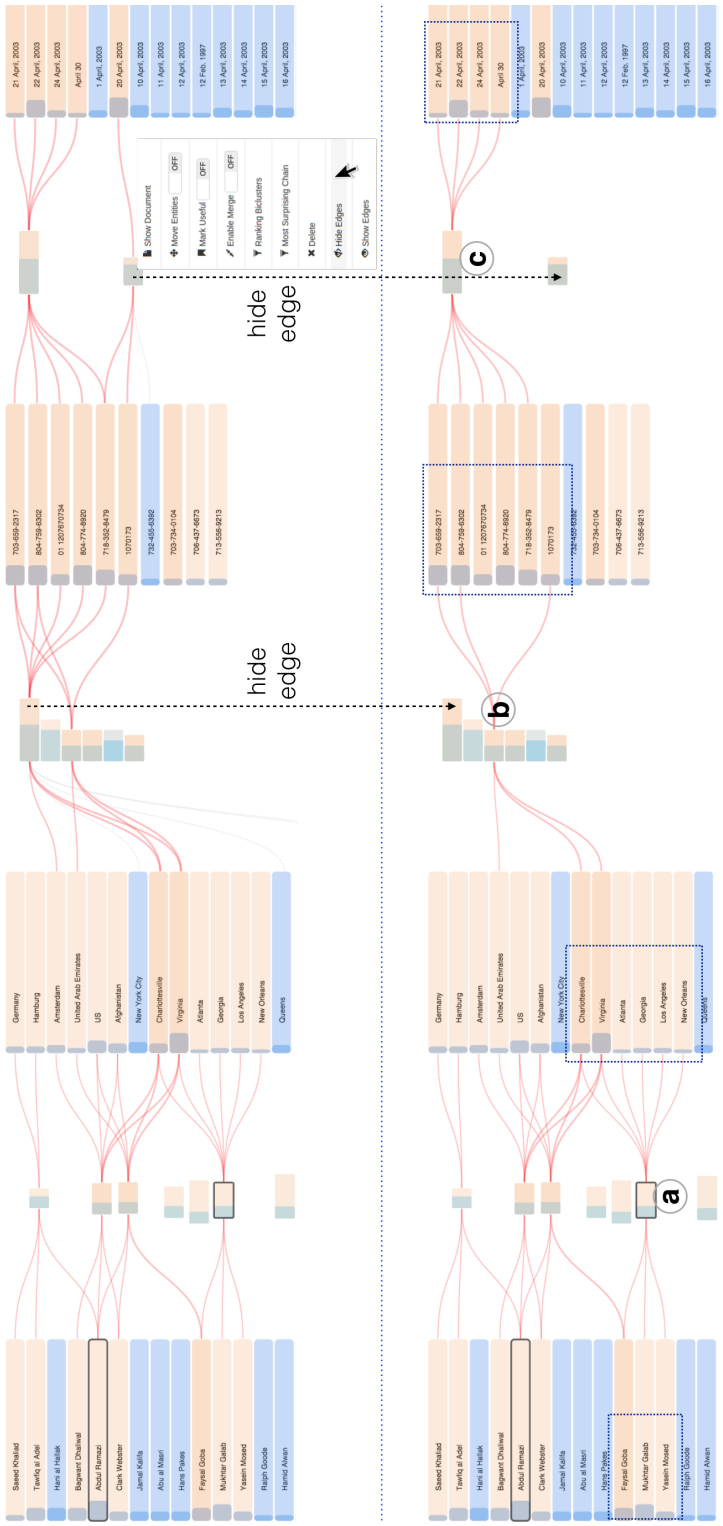

Fig. 16. Using a right-click menu from edge bundles to hide edges. Referring to the four connected groups of useful entities, through three bundles (*a*, *b* and *c*), for hypothesis generation. These entities are those inside the boxes with dotted borders.

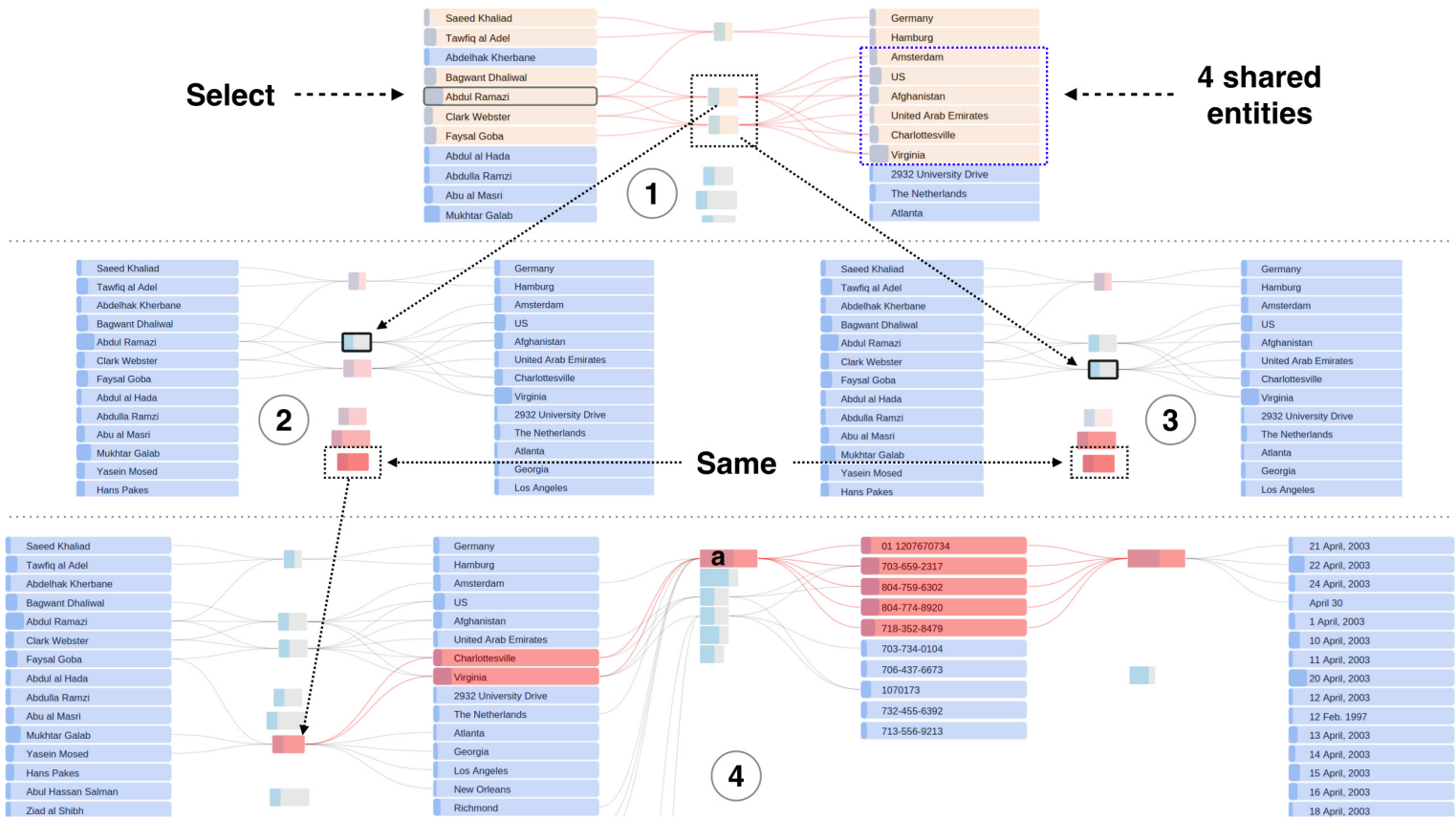

Fig. 17. A process of finding one major threat plot with key steps. (1): Based on *A. Ramazi*, finding that there are two similar bundles. (2): Finding the most surprising bundle evaluated by the *maximum entropy model* for one of the two similar bundles. (3): Finding the most surprising bundle evaluated by the *maximum entropy model* for the other one of the two similar bundles. (4). The most surprising bicluster-chain suggested by the *maximum entropy model* for the most surprising bundle identified in previous steps.

ACM Transactions on Knowledge Discovery from Data, Vol. 9, No. 4, Article 39. Publication date: March 2010.

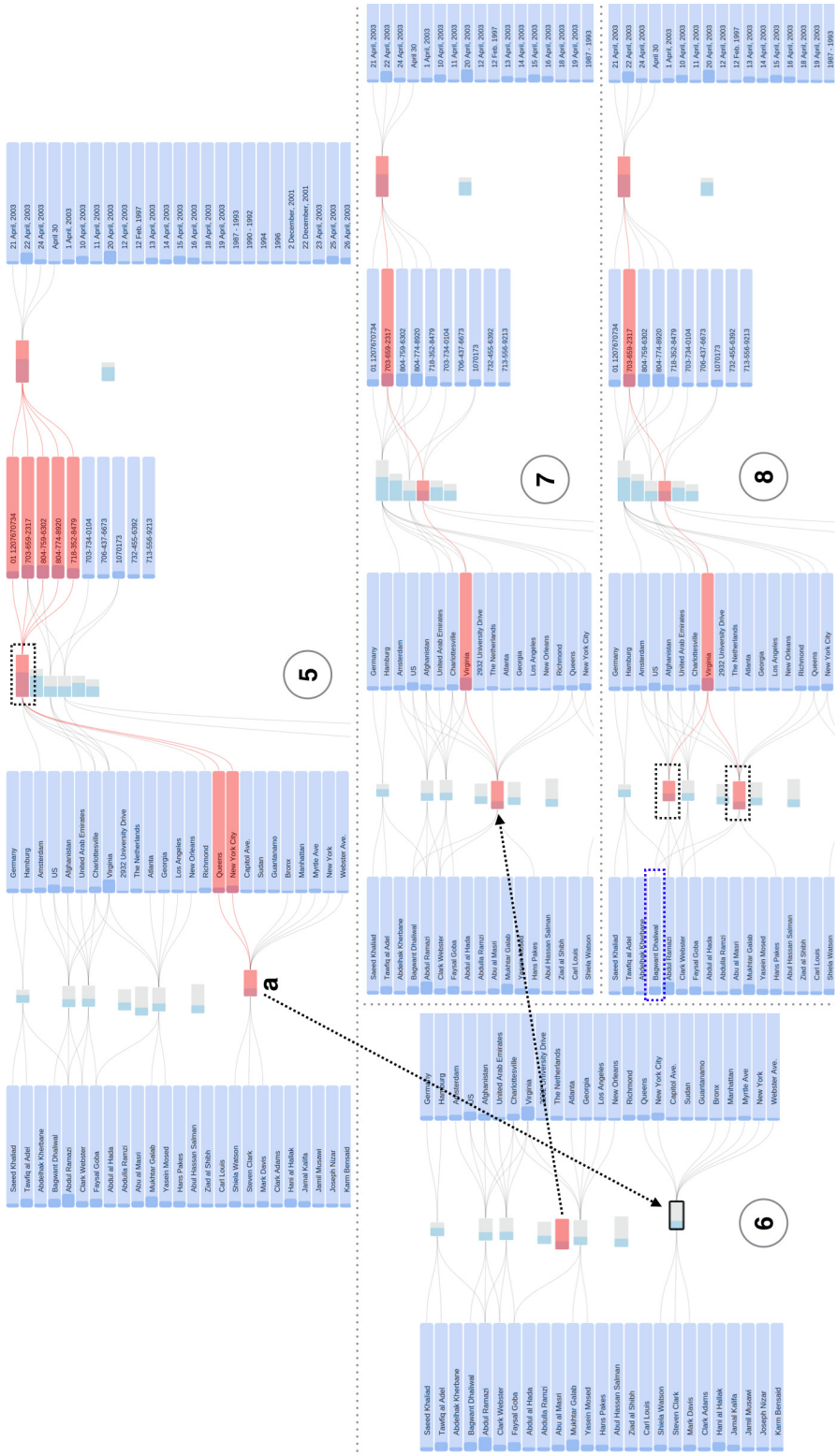

Fig. 18. A process of finding another key threat plot. (5): Finding the most surprising bicluster-chain by the *maximum entropy model* from the bundle marked in the black dotted box. (6): Requesting the *stepwise* evaluation on the bundle *a* in (5). (7): Based on the most surprising bundle shown in (6), requesting to find its most surprising bicluster-chain. (8): Based on the shared entity, *B. Dhaliwal*, requesting to find the most surprising bicluster-chain from the bundle that includes *B. Dhaliwal* and *A. Ramazi*. The chain from this step and that from the previous step merge together.

(4)). Based on this request, MERCER highlights another chain (Figure 18 (5)). This newly highlighted chain has one new bundle (*a* in Figure 18 (5)), and this chain merged with previously suggested surprising chain (comparing Figure 17 (4) with Figure 18 (5)). By checking this newly brought in bundle, Linda finds that all its entities are different from those in previously investigated bundles. In order to connect this new piece of information with previously examined pieces, Linda decides to use the *stepwise* evaluation function on this bundle.

After this stage, MERCER highlights just one bundle (Figure 18 (6)), which is the most surprising one suggested by the model. From this bundle, Linda finds that it includes the person, *B. Dhaliwal*. This quickly catches her attention since she remembers that *B. Dhaliwal* is connected with *A. Ramazi* (Figure 17 (1)). Because of this connection, Linda decides to find more information from this bundle and another bundle that includes *B. Dhaliwal* and *A. Ramazi* (the bundle on top in the black dotted box in Figure 17 (1)), so she requests the *full path* evaluation from them. Based on the request from the newly highlighted bundle shown in Figure 18 (6), MERCER highlights a new bicluster-chain (Figure 18 (7)). Then based on the evaluation request from the bundle including *B. Dhaliwal* and *A. Ramazi*, MERCER highlights another chain. Linda finds these two chains merge together (Figure 18 (8)). The two merged chains both include new pieces of information which connects with the previous findings. Thus, Linda decides to read the reports that are related to these four bundles.

From the four bundles, in the document view of MERCER, Linda finds in total ten unique reports. Of the ten reports, six show evidences about a new threat and three are those relevant to previously identified threat plot. Based on the six reports, Linda identifies the potential threat as:

> *B. Dhaliwal* and *A. Ramazi* plan to attack the *New York Stock Exchange* at *9:00 am* on *April 30*.

Considering the connections between this plot and the previously identified one (e.g., they share some people's names and date), Linda also confirms that *A. Ramazi* is the key person who coordinates the two planned attacks.

With the capability of model evaluations, in this use case, MERCER effectively directs Linda to discover potentially meaningful biclusters or bicluster-chains. Using colors to visually indicate the model evaluation scores in MERCER, Linda can easily see the most surprising bicluster or bicluster-chain, evaluated by the maximum entropy model. Compared with the previous use case of BiSet, following the model-suggested biclusters or chains saves Linda significant time in checking entity-level overlaps for meaningful bicluster identification. In this use case, the maximum entropy model shares the burden of Linda for foraging information (e.g., finding potentially useful biclusters or chains). Thus, compared with the first use case, Linda can spend more time and effort to synthesize the visualized structured information for hypothesis generation.

6.2.3 Comparison between BiSet and MERCER. Both BiSet and MERCER can highlight entities and biclusters based on connections, and visually present entities and biclusters (algorithmically identified structured information) in an organized manner. However, compared with BiSet, MERCER also enables the highlighting entities and biclusters based on identified surprising coalitions from the maximum entropy model. Comparing the two cases discussed above, we find that MERCER better supports the user's sensemaking process of exploring entity coalitions, than BiSet does, from two key aspects: 1) efficiency and 2) exploring new analytical paths.

Compared with BiSet, MERCER more effectively directs users' attention to potentially useful biclusters or bicluster-chains by visually prioritizing them with colors based on their maximum entropy model evaluation scores. The model evaluation function provided in MERCER eases the process for users to find useful biclusters, particularly compared with manually entity overlap investigation. For example, in the first use case, a user has to examine in total 9 biclusters (4 in the left most bicluster list, 5 in the middle bicluster list and 2 in the right most bicluster list as shown in Figure 16), before she finally identifies a meaningful bicluster-chain that covers the information of a potential threat. However, in the second use case, MERCER directs the user to a bicluster-chain after she investigates 3 biclusters (in the left most bicluster list shown in Figure 17). Although this chain is slightly different from the manually identified one in the first use case, it covers the same amount of information as the other one does. Thus, in the second use case, MERCER saves the user from checking highlighted biclusters in the other two lists, and effectively provides a useful bicluster-chain for users to explore.

Based on the four user selected domains (visualized as a fixed schema), it is hard to identify all three threat plots in the *Crescent* dataset because not all pre-identified biclusters can be shown. However, from the two cases, we can find that MERCER can direct users from one identified plot to a new plot via a surprising bicluster-chain. However, when users manually forage relevant information, it is not easy for them to make such transitions due to cognitive tunneling [56]. In the first use case, the key bundle that can lead to a new plot is actually identified not as useful as the one shown as b in Figure 16. Thus, MERCER significantly aids in identifying coalitions of entities worthy of further exploration.

7 RELATED WORK

In this section we survey related work. In particular, we discuss work related with regard to mining surprising patterns, iterative data mining, mining multi-relational datasets, finding plots in data, and bicluster visualizations for data exploration.

7.1 Mining Biclusters

Mining biclusters is an extensively studied area of data mining, and many algorithms for mining biclusters from varied data types have been proposed, e.g. [2, 5, 39, 42, 57, 59, 63]. Bicluster mining, however, is not the primary aim in this paper; instead it is only a component in our proposed framework. Moreover, the above mentioned studies do not assess whether the mined clusters are subjectively interesting. A comprehensive survey of biclustering algorithms was given by Madeira and Oliveira [34].

7.2 Mining Surprising Patterns

There is, however, significant literature on mining representative/succinct/surprising patterns [e.g., 27] as well as on explicit summarization [e.g., 8]. Wang and Parthasarathy [60] summarized a collection of frequent patterns by means of a row-based MaxEnt model, heuristically mining and adding the most significant itemsets in a level-wise fashion. Tatti [54] showed that querying such a model is PP-hard. Mampaey et al. [35] gave a convex heuristic, allowing more efficient search for the most informative set of patterns. De Bie [10] formalized how to model a binary matrix by MaxEnt using row and column margins as background knowledge, which allows efficient calculation of probabilities per cell in the matrix. Kontonasios et al. [28] first proposed a real-valued MaxEnt model for assessing patterns over real-valued rectangular databases. These papers all focus on mining surprising

patterns from a single relation. They do not explore the multi-relational scenario, and can hence not find connections among surprising patterns from different relations—the problem we focus on.

7.3 Iterative Data Mining

Iterative data mining as we study was first proposed by Hanhijärvi et al. [18]. The general idea is to iteratively mine the result that is most significant given our accumulated knowledge about the data. To assess significance, they build upon the swap-randomization approach of Gionis et al. [15] and evaluate empirical p-values. With the help of real-valued MaxEnt model, Kontonasios et al. [30] proposed a subjective interestingness measure called *Information Ratio* to iteratively identify and rank the interesting structures in real-valued data. Mampaey et al. [35] and Kontonasios et al. [30] show that ranking results using a static MaxEnt model leads to redundancy in the top-ranked results, and that iterative updating provides a principled approach for avoiding this type of redundancy. Tatti and Vreeken [55] discussed comparing the informativeness of results by different methods on the same data. They gave a proof-of-concept for single binary relations, for which results naturally translate into tiles, and gave a MaxEnt model in which tiles can be incorporated as background knowledge. In this work we build upon this framework, translating bicluster chains (over multiple relations) into tiles to measure surprisingness with regard to background knowledge using a maximum entropy model.

7.4 Multi-relational Mining

Mining relational data is a rich research area [12] with a plethora of approaches ranging from relational association rules [11] to inductive logic programming (ILP) [33]. The idea of composing redescriptions [64] and biclusters to form patterns in multi-relational data was first proposed by Jin et al. [24]. Cerf et al. [4] introduced the DataPeeler algorithm to tackle the challenge of directly discovering closed patterns from n-ary relations in multi-relational data. Later, Cerf et al. [3] refined DataPeeler for finding both closed and noise-tolerant patterns. These frameworks do not provide any criterion for measuring subjective interestingness of the multi-relational patterns. Ojala et al. [36] studied randomization techniques for multi-relational databases with the goal to evaluate the statistical significance of database queries. Spyropoulou and De Bie [43] and Spyropoulou et al. [46] proposed to transform a multi-relational database into a K-partite graph, and to mine maximal complete connected subset (MCCS) patterns that are surprising with regard to a MaxEnt model based on the margins of this data. Spyropoulou et al. [45] extended this approach to finding interesting local patterns in multi-relational data with n-ary relationships. Bicluster chains and MCCS patterns both identify redescriptions between relations, but whereas MCCS patterns by definition only identify exact pair-wise redescriptions (completely connected subsets), bicluster chains also allow for approximate redescriptions (incompletely connected subsets). All except for the most simple bicluster chains our methods discovered in the experiments of Section 6 include inexact redescriptions, and could hence not be found under the MCCS paradigm. Another key difference is that we iteratively update our MaxEnt model to include all patterns we mined so far. Later, Spyropoulou and De Bie [44] further extended MCCS patterns to support discovering approximate multi-relational patterns (α-CCS), where dense local patterns (e.g. dense tiles but not biclusters in our scenario) from binary relations are allowed to be used to construct the α-CCS patterns. Compared to our bicluster chain approach proposed in this paper, these are two different fundamental approaches to formulate and

solve the same problem. However, the differences between these two approaches from the theoretical perspective and whether they perform similarly or not over the same dataset need to be further investigated in future work.

7.5 'Finding Plots'

The key difference between finding plots, and finding biclusters or surprising patterns is the notion of chaining patterns into a chain, or plot. Commercial software such as *Palantir* provide significant graphic and visualization capabilities to explore networks of connections but do not otherwise automate the process of uncovering plots from document collections. Shahaf and Guestrin [40] studied the problem of summarizing a large collection of news articles by finding a chain that represents the main events; given either a start or end-point article, their goal is to find a chain of intermediate articles that is maximally coherent. In contrast, in our setup we know neither the start nor end points. Further, in intelligence analysis, it is well known that plots are often loosely organized with no common all-connecting thread, so coherence cannot be used as a driving criterion. Most importantly, we consider data matrices where a row (or, document) may be so sparse or small (e.g., 1-paragraph snippets) that it is difficult to calculate statistically meaningful scores. Storytelling algorithms [e.g., 20, 21, 31] are another related thread of research; they provide algorithmic ways to rank connections between entities but do not focus on entity coalitions and how such coalitions are maintained through multiple sources of evidence. Wu et al. [61] proposed a framework to discover the plots by detecting non-obvious coalitions of entities from multi-relational datasets with maximum entropy principle and further support iterative, human-in-the-loop, knowledge discovery. However, no visualization framework was developed to enable analysts to be involved when discovering the surprising entity coaliations in that work. Moreover, we also propose the full path and step-wise chain search strategies and combine them together to help analyts to explore the data.

7.6 Bicluster Visualizations

Finally, we give an overview of work on bicluster visualization techniques. Biclusters offer a usable and effective way to present coalitions among sets of entities across multiple domains. Various visualizations have been proposed to present biclusters for sensemaking of data in different fields. One typical application domain of bicluster visualizations is bioinformatics, where biclusters are visualized to help bioinformaticians to identify groups of genes that have similar behavior under certain groups of conditions (e.g., BicAt [1], Bicluster viewer [19], BicOverlapper 2.0 [38], BiGGEsTS [16], BiVoc [17], Expression Profiler [26], GAP [62] and Furby [48]). In addition, Fiaux et al. [13] and Sun et al. [50] applied biclusters in Bixplorer, a visual analytics tool, to support intelligence analysts for text analytics. Evaluations of these tools show promising results, which indicates that using visualized bicluster to empower data exploration is beneficial.

In order to systematically inform the design of bicluster visualizations, a five-level design framework has been proposed [53] and the key design trade-off to visualize biclusters has been identified: *Entity-centric* and *relationship-centric* [51]. This design framework highlights five levels of relationships that underlie the notions of biclusters and bicluster chains. The design trade-off suggests that bicluster visualizations should visually represent both the membership of entities and the overlap among biclusters in a human perceptible and usable manner.

8 CONCLUSION

Our approach to discover multi-relational patterns with maximum entropy models in a visual analytics tool is a significant step in formalizing a previously unarticulated knowledge discovery problem and supporting its solution in an interactive manner. We have primarily showcased results in intelligence analysis; however, the theory and methods presented are applicable for analysis of unstructured or discrete multi-relational data in general—such as for biological knowledge discovery from text. The key requirement to apply our methods is that the data should be transformed into our data model.

Some of the directions for future work include (i) obviating the need to mine all biclusters prior to composition, (ii) improving the scalability of the proposed models and framework to be able to deal with even larger datasets, (iii) enabling dynamic and flexible multi-relational schema generation to support better sensemaking and hidden plot discovery, (iv) incorporating weights on relationships to account for differing veracities and trustworthiness of evidence. Ultimately, the key is to support more expressive forms of human-in-the-loop knowledge discovery.

ACKNOWLEDGMENTS

This research has been supported by US National Science Foundation grants CCF-0937133, IIS-1447416, DGE-1545362, IIS-1633363 and the Institute for Critical Technology and Applied Science, Virginia Tech. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

REFERENCES

[1] Simon Barkow, Stefan Bleuler, Amela Prelić, Philip Zimmermann, and Eckart Zitzler. 2006. BicAT: a biclustering analysis toolbox. *Bioinformatics* 22, 10 (2006), 1282–1283.
[2] Andrea Califano, Gustavo Stolovitzky, and Yuhai Tu. 2000. Analysis of Gene Expression Microarrays for Phenotype Classification. In *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 75–85.
[3] Loïc Cerf, Jérémy Besson, Kim-Ngan T. Nguyen, and Jean-François Boulicaut. 2013. Closed and noise-tolerant patterns in n-ary relations. *Data Min. Knowl. Discov.* 26, 3 (2013), 574–619.
[4] Loïc Cerf, Jérémy Besson, Céline Robardet, and Jean-François Boulicaut. 2009. Closed Patterns Meet N-ary Relations. *TKDD* 3, 1, Article 3 (March 2009), 36 pages.
[5] Yizong Cheng and George M. Church. 2000. Biclustering of Expression Data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, 93–103.
[6] I. Csiszar. 1975. *I*-Divergence geometry of probability distributions and minimization problems. *The Annals of Probability* 3, 1 (1975), pp. 146–158.
[7] J. N. Darroch and D. Ratcliff. 1972. Generalized Iterative Scaling for Log-Linear Models. *The Annals of Mathematical Statistics* 43, 5 (1972), pp. 1470–1480.
[8] Warren L. IV Davis, Peter Schwarz, and Evimaria Terzi. 2009. Finding representative association rules from large rule collections.. In *SDM'09*. SIAM, 521–532.
[9] Tijl De Bie. 2011. An Information Theoretic Framework for Data Mining. In *KDD '11*. ACM, 564–572.
[10] Tijl De Bie. 2011. Maximum entropy models and subjective interestingness: an application to tiles in binary databases. *Data Min. Knowl. Discov.* 23, 3 (2011), 407–446.
[11] Luc Dehaspe and Hannu Toironen. 2000. Discovery of relational association rules. In *Relational Data Mining*, Sašo Džeroski (Ed.). Springer-Verlag New York, Inc., 189–208.
[12] S. Dzeroski and N. Lavrac (editors). 2001. *Relational Data Mining*. Springer, Berlin.
[13] Patrick Fiaux, Maoyuan Sun, Lauren Bradel, Chris North, Naren Ramakrishnan, and Alex Endert. 2013. Bixplorer: Visual analytics with biclusters. *Computer* 46, 8 (2013), 90–94.
[14] Floris Geerts, Bart Goethals, and Taneli Mielikainen. 2004. Tiling databases. In *Discovery Science '04*. Springer, 278–289.
[15] Aristides Gionis, Heikki Mannila, Taneli Mielikäinen, and Panayiotis Tsaparas. 2007. Assessing data mining results via swap randomization. *TKDD* 1, 3 (2007), 167–176.

[16] Joana P Gonçalves, Sara C Madeira, and Arlindo L Oliveira. 2009. BiGGEsTS: integrated environment for biclustering analysis of time series gene expression data. *BMC Research Notes* 2, 1 (2009), 124.
[17] Gregory A Grothaus, Adeel Mufti, and TM Murali. 2006. Automatic layout and visualization of biclusters. *Algorithms for Molecular Biology* 1, 1 (2006), 15.
[18] Sami Hanhijärvi, Markus Ojala, Niko Vuokko, Kai Puolamäki, Nikolaj Tatti, and Heikki Mannila. 2009. Tell me something I don't know: randomization strategies for iterative data mining. In *KDD'09*. ACM, 379–388.
[19] Julian Heinrich, Robert Seifert, Michael Burch, and Daniel Weiskopf. 2011. BiCluster viewer: a visualization tool for analyzing gene expression data. In *Advances in Visual Computing*. Springer, 641–652.
[20] M.S. Hossain, J. Gresock, Y. Edmonds, R. Helm, M. Potts, and N. Ramakrishnan. 2012. Connecting the Dots between PubMed Abstracts. *PLoS ONE* 7, 1 (2012).
[21] M. Shahriar Hossain, Patrick Butler, Arnold P. Boedihardjo, and Naren Ramakrishnan. 2012. Storytelling in entity networks to support intelligence analysts. In *KDD'12*. ACM, 1375–1383.
[22] 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).
[23] E. T. Jaynes. 1957. Information Theory and Statistical Mechanics. *Phys. Rev.* 106, 4 (1957), 620–630.
[24] Ying Jin, T. M. Murali, and Naren Ramakrishnan. 2008. Compositional mining of multirelational biological datasets. *TKDD* 2, 1, Article 2 (April 2008), 35 pages.
[25] Youn-ah Kang, C Gorg, and John Stasko. 2009. 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*. IEEE, 139–146.
[26] Misha Kapushesky, Patrick Kemmeren, Aedín C Culhane, Steffen Durinck, Jan Ihmels, Christine Körner, Meelis Kull, Aurora Torrente, Ugis Sarkans, Jaak Vilo, and others. 2004. Expression Profiler: next generation-an online platform for analysis of microarray data. *Nucleic acids research* 32, suppl 2 (2004), W465–W470.
[27] Jerry Kiernan and Evimaria Terzi. 2008. Constructing comprehensive summaries of large event sequences. In *KDD'08*. 417–425.
[28] K. Kontonasios, J. Vreeken, and T. De Bie. 2011. Maximum Entropy Modeling for Assessing Results on Real-Valued Data. In *ICDM' 11*. 350–359.
[29] Kleanthis-Nikolaos Kontonasios and Tijl DeBie. 2012. Formalizing Complex Prior Information to Quantify Subjective Interestingness of Frequent Pattern Sets. In *IDA'12*. Springer-Verlag, 161–171.
[30] Kleanthis-Nikolaos Kontonasios, Jilles Vreeken, and Tijl De Bie. 2013. Maximum Entropy Models for Iteratively Identifying Subjectively Interesting Structure in Real-Valued Data. In *ECMLPKDD'13*. Springer, 256–271.
[31] Deept Kumar, Naren Ramakrishnan, Richard F. Helm, and Malcolm Potts. 2006. Algorithms for storytelling. In *KDD'06*. 604–610.
[32] A Lambert, R Bourqui, and D Auber. 2010. Winding Roads: Routing edges into bundles. *Computer Graphics Forum* 29, 3 (Aug. 2010), 853–862.
[33] N. Lavrac and P.A. Flach. 2001. An Extended Transformation Approach to Inductive Logic Programming. *ACM Transactions on Computational Logic* Vol. 2, 4 (Oct 2001), pages 458–494.
[34] Sara C. Madeira and Arlindo L. Oliveira. 2004. Biclustering Algorithms for Biological Data Analysis: A Survey. *IEEE/ACM Trans. Comput. Biol. Bioinformatics* 1, 1 (Jan. 2004), 24–45.
[35] Michael Mampaey, Jilles Vreeken, and Nikolaj Tatti. 2012. Summarizing Data Succinctly with the Most Informative Itemsets. *TKDD* 6 (2012), 1–44. Issue 4.
[36] Markus Ojala, Gemma C. Garriga, Aristides Gionis, and Heikki Mannila. 2010. Evaluating Query Result Significance in Databases via Randomizations. In *SDM'10*. 906–917.
[37] G. Rasch. 1960. *Probabilistic Models for Some Intelligence and Attainnment Tests*. Danmarks paedagogiske Institut.
[38] Rodrigo Santamaría, Roberto Therón, and Luis Quintales. 2014. BicOverlapper 2.0: visual analysis for gene expression. *Bioinformatics* (2014), btu120.
[39] Eran Segal, Ben Taskar, Audrey Gasch, Nir Friedman, and Daphne Koller. 2001. Rich probabilistic models for gene expression. *Bioinformatics* 17, suppl 1 (2001), S243–S252.
[40] Dafna Shahaf and Carlos Guestrin. 2012. Connecting Two (or Less) Dots: Discovering Structure in News Articles. *TKDD* 5, 4, Article 24 (Feb. 2012), 31 pages.

[41] Amnon Shashua. 2008. Introduction to Machine Learning - 67557 Lecture Notes. http://arxiv.org/pdf/0904.3664.pdf. (2008).
[42] Qizheng Sheng, Yves Moreau, and Bart De Moor. 2003. Biclustering microarray data by Gibbs sampling. *Bioinformatics* 19, suppl 2 (2003), ii196–ii205.
[43] Eirini Spyropoulou and Tijl De Bie. 2011. Interesting Multi-relational Patterns. In *ICDM'11*. 675–684.
[44] Eirini Spyropoulou and Tijl De Bie. 2014. Mining approximate multi-relational patterns. In *DSAA'14*. 477–483.
[45] Eirini Spyropoulou, Tijl De Bie, and Mario Boley. 2013. Mining Interesting Patterns in Multi-relational Data with N-ary Relationships. In *Discovery Science*. Lecture Notes in Computer Science, Vol. 8140. Springer Berlin Heidelberg, 217–232.
[46] Eirini Spyropoulou, Tijl De Bie, and Mario Boley. 2014. Interesting pattern mining in multi-relational data. *Data Min. Knowl. Discov.* 28, 3 (2014), 808–849.
[47] John Stasko, Carsten Görg, and Zhicheng Liu. 2008. Jigsaw: supporting investigative analysis through interactive visualization. *Information visualization* 7, 2 (2008), 118–132.
[48] 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.
[49] Maoyuan Sun. 2016. *Visual Analytics with Biclusters: Exploring Coordinated Relationships in Context*. Ph.D. Dissertation. Virginia Tech.
[50] Maoyuan Sun, Lauren Bradel, Chris L North, and Naren Ramakrishnan. 2014. The role of interactive biclusters in sensemaking. In *Proceedings of the Conference on Human Factors in Computing Systems*. ACM, 1559–1562.
[51] Maoyuan Sun, Peng Mi, Chris North, and Naren Ramakrishnan. 2016. BiSet: Semantic Edge Bundling with Biclusters for Sensemaking. *IEEE transactions on visualization and computer graphics* 22, 1 (2016), 310–319.
[52] Maoyuan Sun, Peng Mi, Hao Wu, Chris North, and Naren Ramakrishnan. 2016. Usability Challenges Underlying Bicluster Interaction for Sensemaking. In *Human Centered Machine Learning Workshop at ACM CHI 2016*.
[53] Maoyuan Sun, Chris North, and Naren Ramakrishnan. 2014. A Five-Level Design Framework for Bicluster Visualizations. *IEEE transactions on visualization and computer graphics* 20, 12 (2014), 1713–1722.
[54] Nikolaj Tatti. 2006. Computational complexity of queries based on itemsets. *IPL* 98, 5 (2006), 183–187.
[55] Nikolaj Tatti and Jilles Vreeken. 2012. Comparing Apples and Oranges - Measuring Differences between Exploratory Data Mining Results. *Data Min. Knowl. Discov.* 25, 2 (2012), 173–207.
[56] Lisa C Thomas and Christopher D Wickens. 2001. Visual displays and cognitive tunneling: Frames of reference effects on spatial judgments and change detection. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 45. SAGE Publications, 336–340.
[57] Robert Tibshirani, Trevor Hastie, Mike Eisen, Doug Ross, David Botstein, and Pat Brown. 1999. *Clustering methods for the analysis of DNA microarray data*. Technical Report. Stanford University.
[58] Takeaki Uno, Tatsuya Asai, Yuzo Uchida, and Hiroki Arimura. 2004. An efficient algorithm for enumerating closed patterns in transaction databases. In *Discovery Science*. Springer, 16–31.
[59] Takeaki Uno, Masashi Kiyomi, and Hiroki Arimura. 2005. LCM Ver.3: Collaboration of Array, Bitmap and Prefix Tree for Frequent Itemset Mining. In *Proceedings of the 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations (OSDM '05)*. ACM, New York, NY, USA, 77–86.
[60] Chao Wang and Srinivasan Parthasarathy. 2006. Summarizing itemset patterns using probabilistic models. In *KDD'06*. 730–735.
[61] Hao Wu, Jilles Vreeken, Nikolaj Tatti, and Naren Ramakrishnan. 2014. Uncovering the Plot: Detecting Surprising Coalitions of Entities in Multi-relational Schemas. In *ECMLPKDD'14*. Springer.
[62] Han-Ming Wu, Yin-Jing Tien, and Chun-houh Chen. 2010. GAP: A graphical environment for matrix visualization and cluster analysis. *Computational Statistics & Data Analysis* 54, 3 (2010), 767–778.
[63] M.J. Zaki and C.-J. Hsiao. 2005. Efficient algorithms for mining closed itemsets and their lattice structure. *TKDE* 17, 4 (2005), 462–478.
[64] Mohammed J. Zaki and Naren Ramakrishnan. 2005. Reasoning about sets using redescription mining. In *KDD'05*. ACM, 364–373.

Received December 2015; revised August 2016; revised Janunary 2017; accepted January 2017