Data Mining and Knowledge Discovery manuscript No.
(will be inserted by the editor)

Uncovering the Plot: Detecting Surprising Coalitions of Entities in Multi-Relational Schemas

Hao Wu · Jilles Vreeken · Nikolaj Tatti · Naren Ramakrishnan

Received: date / Accepted: date

Abstract Many application domains such as intelligence analysis and cybersecurity require tools for the unsupervised identification of suspicious entities in multi-relational/network data. In particular, there is a need for automated semi-automated approaches to 'uncover the plot', i.e., to detect non-obvious coalitions of entities bridging many types of relations. We cast the problem of detecting such suspicious coalitions and their connections as one of mining surprisingly dense and well-connected chains of biclusters over multi-relational data. With this as our goal , we model data by the Maximum Entropy principle, such that in a statistically well-founded way we can gauge the surprisingness of a discovered bicluster chain with respect to what we already know. We design an algorithm for approximating the most informative multi-relational patterns, and provide strategies to incrementally organize discovered patterns into the background model. We illustrate how our method is adept at discovering the hidden plot in multiple synthetic and real-world intelligence analysis datasets. Our approach naturally generalizes traditional attribute-based

H. Wu (✉)
Department of Electrical and Computer Engineering, Virginia Tech, Arlington, VA, and
Discovery Analytics Center, Virginia Tech, Arlington, VA, USA.
E-mail: wuhao723@vt.edu

J. Vreeken
Max Planck Institute for Informatics, Saarbrücken, Germany, and
Cluster of Excellence MMCI, Saarland University, Saarbrücken, Germany
E-mail: jilles@mpi-inf.mpg.de

N. Tatti
HIIT, Department of Information and Computer Science, Aalto University, Finland, and
Department of Computer Science, KU Leuven, Leuven, Belgium
E-mail: nikolaj.tatti@aalto.fi

N. Ramakrishnan
Department of Computer Science, Virginia Tech, Arlington, VA, USA, and
Discovery Analytics Center, Virginia Tech, Arlington, VA, USA.
E-mail: naren@cs.vt.edu

maximum entropy models for single relations, and further supports iterative, human-in-the-loop, knowledge discovery.

Keywords Multi-relational data · Maximum entropy modeling · Subjective interestingness · Pattern mining · Biclusters

1 Introduction

Knowledge discovery from multi-relational datasets is a crucial task that arises in many domains, e.g., intelligence analysis, biological knowledge discovery. In the domain of intelligence analysis, questions such as 'How is a suspect connected to the passenger manifest on this flight?', 'How do distributed terrorist cells interface with each other?,' are common conundrums faced by analysts. Similarly, 'how do these two pathways influence each other?', 'what is the signal transduction cascade from an extracellular molecule to an intracellular protein?' are typical questions posed by biologists. A pervasive task performed by analysts is thus the laying out of evidence from multiple sources, identifying coalitions of entities, incrementally building connections between them, and chaining these connections to create stories that either serve as end hypotheses or as templates of reasoning that can then be prototyped.

Our work here focuses exclusively on multi-relational datasets, either available in native form or obtained through straightforward 'relationalization' of unstructured text datasets. We focus on discovering patterns that tie together three inter-related aspects: *coalitions* (which groups of entities come together?), *connections* (how do they interface with other groups?), and *chains* (how do such connections form a chain of evidence?). Fig. 1 illustrates a pattern example from the popular *Crescent* dataset used in intelligence analysis. The hidden plot in this dataset is a distributed and loosely organized network of terrorist cells that plan to attack multiple cities. As shown, the plot involves four coalitions (of phone numbers, dates, people, and places), three relations (who called which number, from where, and when), and the combined chain reveals an odd group of people that turns out to be central to communication and coordination between the terrorist cells. Note the multiple phone numbers that were used by the group to coordinate with each other, and by chaining patterns across the phone number interface we are able to discover the distributed terrorist network. Note also that the chains do not necessarily have to be perfect to be informative; here there are phone numbers (e.g, 732-455-6392, 706-437-6673) not perfectly related to other pieces of evidence, yet the entire chain is surprising enough to alert the analyst to the overall pattern. Similarly, in the domain of bioinformatics, it is easy to imagine coalitions of genes, proteins, and other molecules, and where the connections straddle relations such as transcriptional regulation, signal transduction, and small molecule binding, and the combined chain would indicate a cascade of how extracellular inputs get propagated into downstream cellular responses.

Finding surprising chains as shown in Fig. 1 from multi-relational data has thus far been considered somewhat of a black art and requires significant trial-

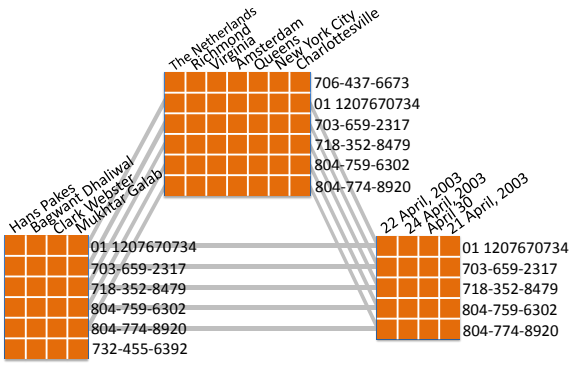


Fig. 1 Uncovering the plot in the *Crescent* dataset. A network of terrorist cells is discovered by a surprising multi-relational pattern involving phone numbers–who called these numbers, where was the call made from, and when was the call made.

and-error on the part of the analyst. Our goal is to formalize these notions so that algorithmic support can complement (and sometimes even supplant and supercede) painstaking manual analysis of large-scale multirelational datasets. In particular, the objective of knowledge discovery is not to extract a unique answer from the dataset but rather to guide an expert into deeper consideration of key process elements.

Our contributions are:

1. We present a formalization of data mining patterns that encapsulate surprising coalitions of entities in multi-relational schemas. More importantly, our approach can help gauge the surprisingness of such patterns w.r.t. prior knowledge on the data and its patterns – a key requirement in intelligence analysis to not mine obvious or pervasive patterns.
2. We develop an algorithm for approximating the most informative multi-relational patterns, with specific consideration to two different data models. These data models represent the two most common ways in which connections are made in domains such as intelligence analysis and bioinformatics.
3. Using results on both synthetic and real-world datasets, we demonstrate how our approach is adept at discovering coalitions, connections, and chains. In particular, we show how our approach fosters human-in-the-loop knowledge discovery whereby an analyst can provide feedback to steer the discovery of patterns.

The rest of this paper is organized as follows. In Section 2, we introduce some preliminaries that will be used in the following sections and formally state the problem studied in this paper. A quick refresher on maximum entropy modeling theory follows in Section 3. Sections 4 and 5 describe in detail our proposed framework and an algorithm to find surprising bicluster chains, respectively. Section 6 outlines experiment results on both synthetic and real datasets. Related work is surveyed in Section 7. Advantages and limitations of the proposed framework are discussed in Section 8, and we round up with conclusions in Section 9.

2 Preliminaries and Problem Statement

Before formalizing our problem statement, we introduce some preliminary concepts and notations.

Multi-relational schema We assume that we are given m domains or *universes* which we will denote throughout the paper by U_i. An entity is a member of U_i and an entity set E_i is simply a subset of U_i. We write $R = R(U_i, U_j)$ to be a binary relation between some 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 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, without loss of generality, all vertices in $\mathcal{R}$ are assumed to have degree of two, i.e., only binary relationships are considered. As is well known, ternary and higher-order relations can be converted into sets of binary relationships. (No such degree constraint exists for $\mathcal{U}$; a domain can participate in many relationships.) Figure 2 shows a toy example of a multi-relational schema involving four domains (i.e., Phone Numbers, Organizations, People, and Places) and three binary relations (Person–Phone Number, Person–Organization , and Organization–Place) between these domains.

We now introduce mechanisms to relate entity sets. *Redescriptions* relate entity sets in the same domain whereas *biclusters* are ways to relate entity sets across domains.

Tiles A *tile* T on binary relationship, a notion introduced by Geerts et al (2004), is essentially a rectangle in a data matrix. Formally, it is defined as a tuple $T = (r(T), c(T))$ where $r(T)$ is a set of row identifier (e.g., row IDs) and $c(T)$ is a set of column identifier (e.g., column IDs) on the matrix representation of the binary relationship. In this most general form, it imposes no constraints on values of the matrix elements identified by a tile. So, each element in a tile could be either 1 or 0. In Figure 2, T_1 is an example of a tile. When all elements within a tile T have the same value (i.e., either all 1s or all 0s) we say it is an exact tile. Otherwise we call it a noisy tile.

Biclusters As local patterns of interest over binary data, we consider biclusters. A *bicluster*, denoted by $B = (E_i, E_j)$, on 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, we cannot expand E_i without modifying E_j and vice versa. If a pair of entities $e_i \in U_i, e_j \in U_j$ belongs to a bicluster B, we denote it by $(e_i, e_j) \in B$.

In Figure 2, B_1, B_2 and B_3 are three biclusters from relation R_1, R_2 and R_3, respectively—T_1 is not a bicluster, as not all its elements are 1s.

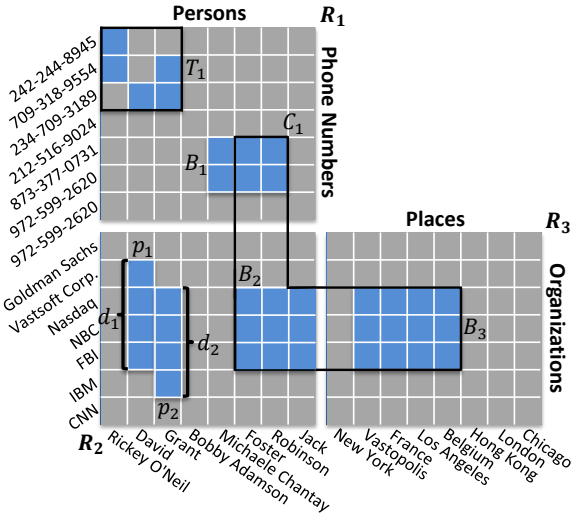


Fig. 2 An example multi-relational schema involving four entity domains (People, Places, Organizations, and Phone Numbers) and three relations (R_1, R_2, R_3). Blue squares indicate presence in the binary relations, and gray squares indicate absence. Examples of redescription ($(p_1, d_1) \sim_{\text{org}} (p_2, d_2)$), biclusters ($B_1$, B_2, B_3), tiles (T_1), and bicluster chains (C_1) are shown.

Redescriptions Assume that we are given 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 say that B and C are *approximate redescriptors* of each other, which we denote 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 a user parameter, 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. If this domain is clear from the context we often drop j from the notation. For example, in Figure 2, we have $(p_1, d_1) \sim_{\varphi,\text{org}} (p_2, d_2)$ for $\varphi \leq 3/5$. 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 is a variant of the definition given by Zaki and Ramakrishnan (2005), whom 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 A *bicluster chain* C consists of 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 we have $B_i \sim_{j_i} B_{i+1}$. Note that this implicitly require that two adjacent biclusters share a common domain.

In Figure 2, C_1 is an example of a bicluster chain comprising three biclusters, viz., B_1, B_2 and B_3. If a bicluster B_{R_i} is a part of the bicluster chain C, we will represent this by $B_{R_i} \in C$ in this paper.

Surprisingness In data mining the main goal is to extract novel knowledge. That is, we aim to find results that are highly informative with regard to what we already know—we are not so much interested in what we already do know, or what we can trivially induce 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 it is high, this indicates that we most likely already know about it, and thus, reporting it would provide little new information. In contrast, if the likelihood of a result is very low, the result is very surprising, which means it conveys a lot of new information. In Section 3, we will discuss how to infer such a probability distribution for binary data. First, we formally define the problem we consider in this paper.

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, as 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 are interested to find K non-redundant bicluster chains that are most surprising with regard to each other and w.r.t. the background knowledge:

Given: Multi-relational Dataset Schema $S(\mathcal{U}, \mathcal{R})$
Find: K Bicluster Chains $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$ such that $\mathcal{C}$ is non-redundant and surprising with respect to the background knowledge of $S(\mathcal{U}, \mathcal{R})$.

Next, we first discuss how to probabilistically model binary data using the Maximum Entropy principle. In Section 4 we will use this model to develop scores for bicluster chains, while in Section 5 we will discuss strategies for efficiently mining surprising bicluster chains from data.

3 MaxEnt Models for Binary Data

Our problem statement is based on a notion of multi-relational schema. For technical reasons we will base our score, however, on binary datasets. More specifically, we assume that our schema was generated from a transactional binary data matrix D. This data matrix can be viewed as a binary matrix of size N-by-M. We will introduce two ways of obtaining a schema from D in Section 4.1. In both of these approaches the columns of D correspond to the entities of the schema. Hence, we will refer to the columns of D as entities.

In this section, we will define the Maximum Entropy (MaxEnt) model for binary data using tiles as background knowledge—recall that a tile is a more general notion than a bicluster. We will first introduce notation that will be

useful to understand the model derivation. Then, we will recall MaxEnt theory for modelling binary data given tiles as background information, and finally, identify how we can fit the model to the data by maximizing the likelihood.

3.1 Notation for Tiles

Given a binary dataset D of size N-by-M and a tile T, the frequency of T in D, $fr(T;D)$, is defined as

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

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

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

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

the expected frequency of tile T under the dataset probability distribution.

Combining these definitions, we have the following lemma.

Lemma 1 *Given a dataset distribution p and a tile T, the 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 dataset having 1 at entry (i,j) under the dataset distribution p.

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

3.2 Global MaxEnt Model from Tiles

Here, we will construct a global statistical model based on 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 obtained from the data. This tile set $\mathcal{T}$ provides information about the data at hand, and we would like to infer a distribution p over the space of possible datasets $\mathcal{D}$ that conforms with the information given in $\mathcal{T}$. That is, we want to be able to determine how probable is a dataset $D \in \mathcal{D}$ given the tile set $\mathcal{T}$.

To derive a good statistical model, we take a principled approach and employ the Maximum Entropy principle (Jaynes, 1957) from information theory. Loosely speaking, the MaxEnt principle identifies the best distribution given background knowledge as the unique distribution that represents the provided background information but is maximally random otherwise. MaxEnt modelling has recently become popular in data mining as a tool for identifying *subjective* interestingness of results with regard to background knowledge (Wang and Parthasarathy, 2006; De Bie, 2011; Tatti and Vreeken, 2012).

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

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

Here, $H(p)$ represents the entropy of the dataset 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 in (Csiszar, 1975) 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

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

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

De Bie (2011) formalized the MaxEnt model for a binary matrix D given row and column margins—also known as a Rasch (1960) model. Here, we consider the more general scenario of binary data and tiles, for which we additionally know (Theorem 2 in Tatti and Vreeken, 2012) that given a tile set $\mathcal{T}$, with $\mathcal{T}(i,j) = \{T \in \mathcal{T} \mid (i,j) \in \sigma(T)\}$, we can write the distribution $p^*_{\mathcal{T}}$ as

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

where

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

This result allows us to factorize the MaxEnt distribution $P^*_{\mathcal{T}}$ of binary dataset given background information in the form of a set of tiles $\mathcal{T}$ into a product of Bernoulli random variables, each of which represents a single entry in the dataset D. We should emphasize here that this model is different MaxEnt model than when we assume independence between rows in the dataset D (see, e.g., Tatti, 2006; Wang and Parthasarathy, 2006; Mampaey et al, 2012). 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:

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

3.3 Inferring the MaxEnt Distribution

To discover the parameters of the Bernoulli random variable mentioned above, we follow a standard approach and apply the well known Iterative Scaling (IS) algorithm (Darroch and Ratcliff, 1972) to infer the tile based MaxEnt distribution on binary dataset. Basically, for each tile $T \in \mathcal{T}$, the algorithm updates the probability distribution p such that the expected frequency of 1s under distribution p for that will match the given frequency γ_T. Clearly, during this update we may change the expected frequency for other tiles, and hence several iterations are needed until the probability distribution p converges. For the proof of algorithm convergence, please refer to Theorem 3.2 in (Csiszar, 1975). In practice, it typically takes on the order of seconds for the algorithm to converge.

4 Scoring Bicluster Chains

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

4.1 Data Model Specification

In this section, we describe two approaches to construct multi-relational schemas $S(\mathcal{U}, \mathcal{R})$ from binary transaction data D. Whenever an element $D(r, e_i)$ has value 1, 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.)

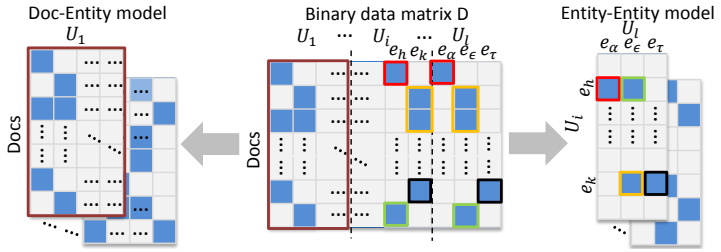


Fig. 3 Illustration of the two data models. The entities marked with red, orange, green, and black squares in matrix D appear together in some documents, thus, induce the, resp., red, orange, green, and black squares in the Entity–Entity model on the right, respectively. The data marked with large brown rectangular in matrix D involves only one type of entities (U_1), which results the Doc–Entity model on the left. See Figure 4 for a toy example.

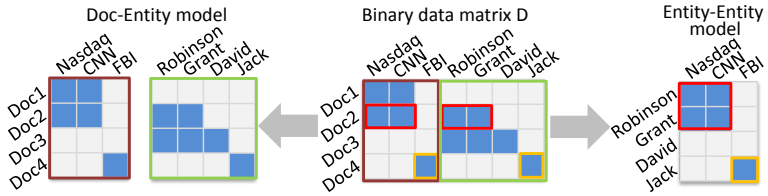


Fig. 4 Toy example of how to construct our two data models from a binary matrix D.

Entity–Entity Model. In the Entity–Entity data model, each binary relation in $\mathcal{R}$ stores the entity co-occurrences in data matrix D between two entity domains. More specific, 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. The right-hand side of Figure 3 illustrates this data model for the example transactional binary data matrix D depicted in the middle.

Example 1 We illustrate how to construct this model using the toy example depicted in Figure 4. We show the Entity–Entity model (right) for the toy data matrix D (middle). The binary data matrix D consists of 4 documents (rows) over 7 entities (columns) which belong to two entity domains: Organizations and Persons. We observe entities *Robinson* and *Grant* of entity domain *Persons* appear together with entities *Nasdaq* and *CNN* of domain *Organizations* in document *Doc2* in D (marked with red squares). For the corresponding Entity–Entity model this hence induces the four relation instances $(Robinson, Nasdaq)$, $(Robinson, CNN)$, $(Grant, Nasdaq)$ and $(Grant, CNN)$ (also marked with red square) in the Persons-Organizations relation on the right-hand side of the figure. Analogue, the person entity *Jack* and organization entity *FBI* (marked with yellow squares) appear together in document *Doc4*, which induces the relation instance $(Jack, FBI)$ (marked with yellow square) in the bottom-right corner of Persons-Organizations relation.

Doc–Entity Model. In the Doc–Entity data model, we treat the rows in D as a special entity domain, U_D, in the multi-relational schema $S(\mathcal{U}, \mathcal{R})$.

More in particular, rows in D are considered the common interconnecting entity domain relating the rest of the other entity domains, and leads to a 'unidirectional' schema—unlike, say the schema shown in Fig. 2. In this data model, each binary relation in $\mathcal{R}$ contains the entity occurrence information of each entity domain in binary data matrix D. For every $R = R(U_D, U_i)$ in $\mathcal{R}$, $(r, e) \in R$ for $r \in U_D$, $e \in U_i$, and transaction r contains entity e in the data matrix D, $D(r, e_k) = 1$. Figure 3 illustrates the concepts of the Doc–Entity (left) model for the given data matrix D (middle).

Example 2 Let us also illustrate the Doc–Entity model using the toy data depicted in Figure 4. We show the Doc–Entity model (left) for the given data matrix D (middle). As D consists of two entity types (Persons and Organizations) the Doc–Entity model consists of two relations, i.e., Document-Organization and Document-Person—which contain the exact same entity occurrence information as left (marked with brown square) and right (marked with green square) hand sides of data matrix D.

Here, every pair of biclusters naturally shares a common domain, the documents U_D. Hence, any pair of biclusters $B = (E_i, E_D)$ and $C = (F_j, F_D)$ here redescribe each other iff $B \sim_{\varphi,D} C$. Further, in practice we only consider chaining biclusters that share a single domain, and hence for this model have that the bag of domain indices associated with the chain consists of only D's.

4.2 When is What Model Applicable?

The choice of whether to use the Entity–Entity versus the Doc–Entity model carries many ramifications. At first glance, it might appear that a bicluster in the Entity–Entity model must be equivalent to a bicluster chain (of two biclusters) in the Doc–Entity model. To see why this is not so, consider what it means to be a bicluster in each of these models. In the Entity–Entity model, a bicluster (E_i, E_j) captures two sets of entities that co-occur, but the co-occurrence could be derived from *many different* documents. In contrast, in the Doc–Entity model, a bicluster chain relating entity sets E_i and E_j must do so using the *same* or a significantly overlapping set of documents. It is hence instructive to view the Entity–Entity model as a 'multiple source of evidence' model, whereas the Doc–Entity model is a 'common domain' model. The former better integrates disparate sources of evidence, each of which may not be significant in itself. The latter provides stronger evidence for inference and is consequentially stricter in evidence integration. As we will show in our results, both models have uses in intelligence analysis.

4.3 Background Model Definition

Next, to discover non-trivial and interesting bicluster chains, we need to incorporate some basic information about the multi-relational schema $S(\mathcal{U}, \mathcal{R})$

into the model. As such basic background knowledge over D we use the column marginals, and the row marginals per entity domain. To this end, following Tatti and Vreeken (2012) 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

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

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. 3.3).

Example 3 Using again the toy data of Figure 4, a tile in $\mathcal{T}_{col}$ would be $(\{Doc1, Doc2, Doc3, Doc4\}, \{Nasdaq\})$, a tile in $\mathcal{T}_{row}$ would be $(\{Doc1\}, \{Nasdaq, CNN, FBI\})$, and a tile in $\mathcal{T}_{dom}$ would be $(\{Doc1, Doc2, Doc3, Doc4\}, \{Nasdaq, CNN, FBI\})$.

4.4 Assessing the Quality of a Bicluster Chain

To assess the quality of a given bicluster chain C 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 each of our two data models.

Entity–Entity Model: For each bicluster $B \in C$ in a chain C, with $B = (E_i, E_j)$, we construct a tile set $\mathcal{T}_B$, consisting of $|E_i|\,|E_j|$ tiles, as follows

$$\mathcal{T}_B = \{(rows(X; D),\ X) \mid X = \{e_i, e_j\} \text{ with } (e_i, e_j) \in B\} \quad , \tag{3}$$

where $rows(X; D)$ is the set of rows that contain X in D. The tile set that corresponds to a bicluster chain C is then $\mathcal{T}_C = \bigcup_{B \in C} \mathcal{T}_B$.

Example 4 Considering the example Entity–Entity bicluster $B = (\{Robinson, Grant\}, \{Nasdaq, CNN\})$ in Figure 4, the entity *Robinson* from *Person* domain and entity *Nasdaq* from *Organization* domain appear together in *Doc2* in the data matrix D. Thus, $rows(\{Robinson, Nasdaq\}; D) = \{Doc2\}$, and the related tile would be $(\{Doc2\}, \{Robinson, Nasdaq\})$. Following the similar logic, the tile set $\mathcal{T}_B$ corresponding to the example bicluster B would be

$$\begin{aligned}\mathcal{T}_B = \{&(\{Doc2\}, \{Robinson, Nasdaq\}), \\ &(\{Doc2\}, \{Robinson, CNN\}), \\ &(\{Doc2\}, \{Grant, Nasdaq\}), \\ &(\{Doc2\}, \{Grant, CNN\})\} \quad .\end{aligned}$$

Doc–Entity Model: For the Doc–Entity model, we only have to construct a single tile T_C to represent a bicluster chain C, with

$$T_C = \left(\bigcup_{(E_D,E_i)\in C} E_D \ , \ \bigcup_{(E_D,E_i)\in C} E_i \right) \quad , \tag{4}$$

where each E_D is a set of documents, $E_D \subseteq U_D$, each entity set $E_i \subseteq U_i$, and $B = (E_D, E_i)$ is a bicluster in chain C. Note that unions are well defined in Eq. 4 since E_D are subsets of U_D, the domain corresponding to the transactions, while the domains of E_i together consistute the items of the dataset. Trivially, the tile set corresponding to the bicluster chain C is $\mathcal{T}_C = \{T_C\}$.

Example 5 Considering the example bicluster chain $C = \{(\{Doc1, Doc2\}, \{Nasdaq, CNN\}), (\{Doc2, Doc3\}, \{Robinson, Grant\})\}$ in the Doc–Entity model in Figure 4 as an example, the tile representing this bicluster chain for the Doc–Entity model is

$$T_C = (\{Doc1, Doc2, Doc3\}, \{Nasdaq, CNN, Robinson, Grant\}) \quad .$$

4.5 Measuring Relative Quality

To determine the relative quality of a bicluster chain $C_i \in \mathcal{C}$ in a set of bicluster chains $\mathcal{C}$—for example, to rank results—we propose to calculate its relative importance: how much novel information does C_i provide in contrast to our background knowledge and the rest of the chain? We will do this by inferring two probability distributions using MaxEnt, and then calculating the divergence between these distributions.

More formally, we first infer a MaxEnt model P_{full} based on all available information. That is, P_{full} is based on $\mathcal{T}_{back} \cup \mathcal{T}_{\mathcal{C}}$, where $\mathcal{T}_{\mathcal{C}} = \{\mathcal{T}_{C_1}, \mathcal{T}_{C_2}, \ldots, \mathcal{T}_{C_K}\}$ and $K = |\mathcal{C}|$. Next, we infer a second MaxEnt model P_{C_i} based on all information *except* C_i; that is, we infer P_{C_i} based on T_{back} and $\mathcal{T}_{\mathcal{C}\setminus C_i} = \{\mathcal{T}_{C_1}, \mathcal{T}_{C_2}, \ldots, \mathcal{T}_{C_{i-1}}, \mathcal{T}_{C_{i+1}}, \ldots, \mathcal{T}_{C_K}\}$. Finally, we measure the divergence between these two models. To this end, we choose the Kullback-Leibler (KL) divergence (Cover and Thomas, 2006), that is $KL(P_{full} \mid\mid P_{C_i})$, as it is well understood, and fits our setup both in goal as well as with regard to ease of computation. In theory, however, other divergence measures could be considered.

5 Searching For Good Chains

In this section, we will describe the strategy to find interesting bicluster chains from multi-relational dataset schema. In theory, to discover a set of interesting bicluster chains $\mathcal{C}$ we could exhaustively explore the search space and evaluate each and every bicluster chain. Clearly, however, the number of possible bicluster chains is prohibitively large; their number is in the order of $O(\prod_{i=1}^{|\mathcal{R}|} |\mathcal{B}_i|)$ where $\mathcal{B}_i$ is the set of biclusters, which are eligible to form bicluster chains,

Algorithm 1: Greedy Inference of Bicluster Chains

input : background MaxEnt model P_{back};
$\mathcal{B}_{all} = \{\mathcal{B}_i\}$ where $\mathcal{B}_i$ is a set of biclusters from R_i in $\mathcal{R}$;
K, the desired number of bicluster chains.
output: set of bicluster chains $\mathcal{C}$.

$\mathcal{C} \leftarrow \emptyset$;
while $(|\mathcal{C}| < K)$ **do**
  $R_{start} \leftarrow$ `mostBiclusterRelation(`$\mathcal{R}$`)`;
  $B^*_{start} \leftarrow$ `findStartBicluster(`R_{start}`)`;
  $C \leftarrow B^*_{start}$;
  $\mathcal{B} \leftarrow$ `eligibleBiclusters(`$\mathcal{B}_{all}$`,` C`)`;
  while $|\mathcal{B}| \neq 0$ **do**
    $B^* \leftarrow \arg\max_{B \in \mathcal{B}} s_{global}(B)$;
    $C \leftarrow$ `addToChain(`C`,` B^*`)`;
    $\mathcal{B} \leftarrow$ `eligibleBiclusters(`$\mathcal{B}_{all}$`,` C`)`;
  end
  $P_{back} \leftarrow$ `UpdateMaxEntModel(`P_{back}`,` C`)`;
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{C\}$;
end
$\mathcal{C} \leftarrow$ `SortChains(`$\mathcal{C}$`)`;

from relation R_i in $\mathcal{R}$. This number explodes if either the number of relations $|\mathcal{R}|$, or the number of candidate biclusters $|\mathcal{B}_i|$ is non-trivial. Moreover, the search space does not exhibit structure (e.g. monotonicity) that we can exploit for efficient search. Hence, we resort to heuristics.

We employ a simple iterative greedy search, that, as we will see, works well in practice. First we define how to greedily choose bicluster candidates making use of our MaxEnt model. Given a bicluster B, we generate a corresponding set of tiles $\mathcal{T}_B$. This conversion is different between our two data models.

Entity–Entity Data Model. When considering the Entity–Entity model, we generate the tile set $\mathcal{T}_B$ according to the definition in Equation (3).

Doc–Entity Data Model. When considering the Doc–Entity model, the bicluster B already represents the tile and hence we use $\mathcal{T}_B = \{B\}$.

Next, we measure how much information $\mathcal{T}_B$ gives with regard to the background model P_{back}. We define the following score,

$$s_{global}(B) = KL(P_B || P_{back}) \quad , \tag{5}$$

where P_B is the MaxEnt model inferred on both tile sets $\mathcal{T}_{back}$ and $\mathcal{T}_B$. As this score measures the amount of information B adds with regard to the full data, we refer to this score as the *global* score. The larger $s_{global}(B)$ is, the more new information B contains—and, we say, the more likely B is a good candidate to form an interesting bicluster chain.

Next, we explain our search strategy for mining interesting bicluster chains. Algorithm 1 illustrates the algorithm. Starting from the relation in $\mathcal{R}$ that has

the most biclusters (Line 3), say R_{start}, we choose the bicluster B^*_{start} from R_{start} as starting bicluster for a bicluster chain such that the score defined in Equation (5) is maximized (Line 4), that is,

$$B^*_{start} = \arg \max_{B \subseteq R_{start}} s_{global}(B) \quad .$$

Given a starting bicluster B^*_{start} that serves as the first bicluster in our chain C, we proceed as follows. From all relations in $\mathcal{R}$ not yet covered by a $B \in C$, we select the set of biclusters $\mathcal{B} \subset \mathcal{B}_{all}$ that we can add to C. That is, we consider every $B(E_i, E_j) \in \mathcal{B}_{all}$ over relation $R(U_i, U_j)$ for which a) the current chain C does not already include a bicluster over relation $R(U_i, U_j)$, and b) which are redescriptions of either the first bicluster $B_{first} \in C$, or the last bicluster $B_{last} \in C$ in chain C. We extend C with the most informative bicluster $B^* \in \mathcal{B}$, with

$$B^* = \arg \max_{B \in \mathcal{B}} s_{global}(B) \quad .$$

We then iterate; we re-calculate the set of eligible biclusters $\mathcal{B}$ (Line 10), and continue to find the next bicluster B^* to add to C. The search stops when the chain cannot be extended further, i.e., when $\mathcal{B}$ is empty.

During this search process, we need to score every candidate bicluster $B \in \mathcal{B}$. When using s_{global} this implies we have to infer a MaxEnt model for each and every candidate, which is computationally expensive. Moreover, s_{global} evaluates a candidate globally, whereas typically most information is *local*: only few entries in MaxEnt distribution will be affected by adding B into the model. Making use of this observation, to reduce the computation cost of the chain search procedure, we define the score $s_{local}(B)$ that measures the local surprisingness of a tile set as

$$s_{local}(B) = - \sum_{T \in \mathcal{T}_B} \sum_{(i,j) \in \sigma(T)} \log p^*((i,j) = D(i,j)) \quad , \tag{6}$$

where $p^*((i,j) = D(i,j))$ is the MaxEnt probability defined by the background model. Compared with the previous score defined in Equation (5), this local score does not require re-infer MaxEnt model for each bicluster $B \in \mathcal{B}$, and will hence greatly reduce the time needed by the search.

Once a chain C is completed, we update the background MaxEnt model P_{back} to not re-discover C or chains with close resemblance again (Line 12).[1] We repeat the search process until we have completed finding the top-K bicluster chains, where K is specified by the user. Finally, the set of bicluster chains $\mathcal{C}$ mined from the multi-relational dataset is reordered based on the quality measure defined in Section 4.5 (Line 15).

[1] Note that, to save computation, we do not update the MaxEnt model after adding each B^*. However, in line with the local score, we know that adding a bicluster typically only changes the distribution locally, and as we never re-visit the same relation R in a single chain C the information by B_i is unlikely to influence much the informativeness of B_{i+1}.

Table 1 Dataset Statistics

Dataset	Number of Documents	Number of Entities	**Doc–Entity**	**Entity–Entity**	
			%1s	min %1s	max %1s
Synthetic 1k	1000	1000	0.01 — 0.05	0.01	0.05
Synthetic 2k	2000	2000	0.01 — 0.05	0.01	0.05
Synthetic 3k	3000	3000	0.01 — 0.05	0.01	0.05
Synthetic 5k	5000	5000	0.01 — 0.05	0.01	0.05
Synthetic 10k	10000	10000	0.01 — 0.05	0.01	0.05
Atlantic Storm	111	716	0.0179	0.0261	0.0608
Crescent	41	284	0.0425	0.0357	0.136
Manpad	47	143	0.0299	0.0385	0.0714

Computation Complexity To mine a single chain, at worst case we evaluate every eligible bicluster in every $R_i \in \mathcal{R}$ against our MaxEnt model—i.e., every bicluster in $\mathcal{B} = \bigcup_{i=1}^{|\mathcal{R}|} \mathcal{B}_i$. That is, we have a worst case complexity of $O(v \cdot |\mathcal{B}|)$, where v is the time needed to evaluate the surprisingness of a single bicluster and depends on whether global or local score is used. When using the global score, Eq. (5), v is characterized by the convergence time of the MaxEnt model P_B when evaluating bicluster B. When using the local score, Eq. (6), v statistically relies on the average size of the biclusters under consideration (which implicitly depends on the density of data matrix D). To mine K bicluster chains, the number of bicluster evaluations is $O(K \cdot |\mathcal{B}|)$. While constructing the chain, determining whether a bicluster is an admissible redescription takes $O(1)$ time. All combined, the computation complexity of the greedy search process is $O(v \cdot K \cdot |\mathcal{B}|)$. Using the global score we have to re-infer the MaxEnt model for every eligible bicluster B, while for the local score only after completing a chain.

6 Experiments

We describe experimental results over both synthetic and real datasets. For real datasets, we focus primarily on datasets from the domain of intelligence analysis, although as stated in the introduction, our framework is broadly applicable to many relational data mining contexts. The focuses of our experimental investigations are to answer the following questions:

i. Can the proposed framework discover planted bicluster chains from multi-relational datasets? (Section 6.1)
ii. How does the framework's runtime scale with increasing size and density of the dataset? (Section 6.2)
iii. Does the approach help reveal plots hidden in real data? (Section 6.3)
iv. How does the proposed framework perform against a baseline method? (Section 6.4)
v. Can our approach foster human-in-the-loop knowledge discovery? (Section 6.5)

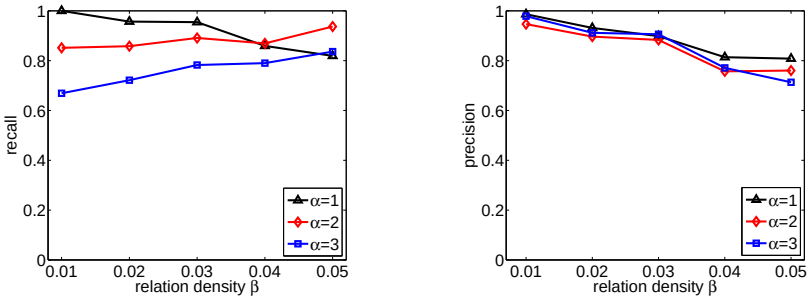


Fig. 5 Recall (left) and Precision (right) scores for the Entity–Entity model on *Synthetic* data of $N = 1000$ by $M = 1000$, of $l = 5$ different types of 200 entities each.

All the experiments described were conducted on a Xeon 2.0 Ghz machine with 528 GB memory. Performance results were obtained by averaging over 10 independent runs. We give the basic statistics of the datasets we use in Table 1. We make the implementation and real datasets publically available for research purposes.[2]

6.1 Synthetic Data

To evaluate against known ground truth and with control over the data characteristics, we generate synthetic data. The synthetic datasets are parameterized as follows. The binary data matrix D consists of N rows and M columns, or entities, which we divide into l different domains. For the Doc–Entity data model, we then have $\mathcal{U} = \{U_D, U_1, U_2, \ldots, U_l\}$ and $\mathcal{R} = \{R_i = R(U_D, U_i) \mid i = 1, 2, \ldots, l\}$. For the Entity–Entity data model, we set the dataset schema $S(\mathcal{U}, \mathcal{R})$ to contain $l - 1$ inter-entity relationships such that $R_i = R(U_i, U_{i+1}), i = 1, 2, \ldots, l - 1$. That is, two adjacent inter-relationships R_i and R_{i+1} share a common entity domain U_{i+1}. These relationships can be extracted from D based on entity co-occurrences.

To verify whether our proposed framework can indeed discover true bicluster chains, we generate the synthetic datasets as follows. The binary data matrix D is first constructed with $N = 1000$ and $M = 1000$, with $l = 5$ entity domains of $\tau = 200$ entities each. As ground truth, α bicluster chains across all the binary relations $R_i \in \mathcal{R}$ are constructed. The rows and columns of each bicluster in the chain are randomly sampled from the domains that $R_i \in \mathcal{R}$ involves. A Jaccard coefficient of $\varphi = 0.75$ is maintained between the adjacent biclusters in the chain. In this experiment, we varied α from 1 to 3. Then, these bicluster chains are planted into the binary data matrix D. Finally, for each entry in D that is not covered by the planted true bicluster chains, we sample its values from a Bernoulli distribution with parameter β' dependent

[2] `http://dac.cs.vt.edu/projects`

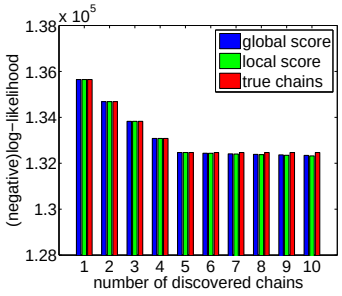


Fig. 6 [Lower is better] Negative log-likelihood scores of the data under the MaxEnt model, for *Synthetic* data of 1000-*by*-1000, of 5 different types of entities with 200 entities per type, and 5 planted bicluster chains. The relation density is $\beta = 0.03$. The decrease of the (negative) log-likelihood indicates the MaxEnt model is more certain about the data.

on R_i—which allows us to control the density β of the binary relations. In this set of experiments, the binary relation density β was varied from 0.01 to 0.05.

To discover the planted chains from the synthetic data, we first consider the global score as defined in Equation (5). As input to our algorithm we first mine candidate biclusters by applying the LCM algorithm (Uno et al, 2005) to each binary relation—note that any bicluster mining algorithm for binary data is applicable. When constructing the chains, we use a Jaccard coefficient threshold of $\varphi = 0.75$ to determine whether two biclusters are approximate redescriptions, and as we are aiming to recover all platned bicluster chains we set $K = \alpha$. We find that for each of the scenarios described above our approach correctly recovers all planted bicluster chains.

Figures 5 shows the recall and precision with regard to entities—the proportion of entities of the *planted* chains discovered by our approach, resp. the proportion of entities in the discovered chains also in the planted chains—for Entity–Entity modeling. We observe that precision decreases slightly with increased density, which stems from cliques being less detectable in more dense data. Due to the random data generation process, and the greedy search, we may observe effects that recall locally increases for higher density. Overall the trend we observe corresponds with intuition: for higher density data it is more difficult to find correct chains, as dense areas are more likely to be created at random. For the Doc–Entity model, the results are perfect, in that both precision and recall are always 1 over all parameter settings, as our framework finds the exact planted true bicluster chains in each of these scenarios. (As these figures are visually not very interesting, we omit them.)

To further investigate how well our approach works in general, and to evaluate both the global and local scores, we run our method on synthetic data as above, planting 5 bicluster chains. We show in Figure 6 the (negative) log-likelihood scores of this data under the MaxEnt models per discovered

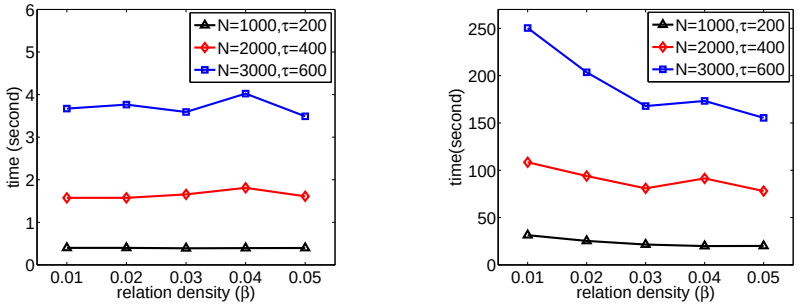


Fig. 7 Time to infer the MaxEnt model (left) resp. time to search for chains (right) on *Synthetic* data over 5 entity domains, 1 planted chain, and of differing density β.

chain. We calculate the likelihood as follows

$$-\sum_{(i,j)\in D} \log p^*_{\mathcal{T}}((i,j) = D(i,j)) \quad . \tag{7}$$

We find that the chains discovered using both the global and local scores very closely match the true chains, and hence do the likelihood scores. Moreover, for both scores the likelihood converges at the true number of chains; this means both that we can determine whether all significant bicluster chains have been discovered, as well as that standard model selection techniques, such as BIC (Schwarz, 1978) or MDL (Rissanen, 1978) are applicable for automatically identifying the correct number of chains in the data.

As a negative result, we report that for *Synthetic* data with a relation density $\beta > 0.1$, we discover only partial chains; likely the (relatively small) planted biclusters do not sufficiently stand out from the dense background. We did not encounter this problem for real data, as they are typically (very) sparse.

6.2 Runtime and Scalability

To evaluate the scalability of our method with regard to different data characteristics, we applied our approach on synthetic datasets of 1000 up to 10 000 rows and columns, and varying the relation densities β from 0.01 to 0.05. We keep the number of entity domains and planted bicluster chains fixed at 5 and 1, respectively. We first evaluate only using our global score.

We first investigate the time needed to infer the MaxEnt model, and the time to search for chains (Figure 7) for datasets of resp. 1000, 2000, and 3000 rows and columns. The figure shows that these aspects are stable with regard to relation density. Figure 8 (left) shows that, as expected, computation mostly depend on the size of the data, and that less time is needed to rank the final chains for more dense. In Figure 8 (right) we summarize the total run time of our approach. Most importantly, in Figure 9 we show that the local score

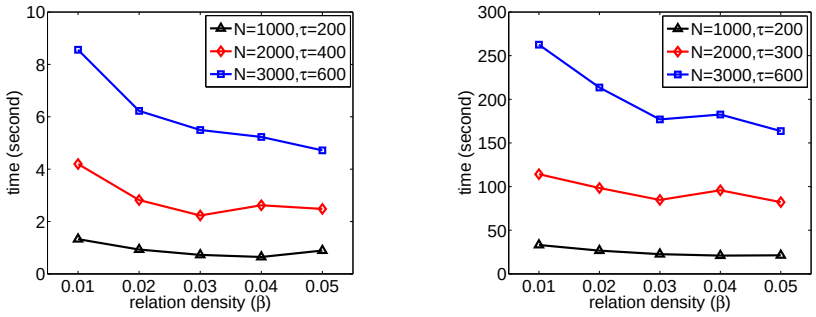


Fig. 8 Time to rank discovered chains (left) resp. total run-time (right) on *Synthetic* data of 5 entity domains, 1 planted chain and of differing density β.

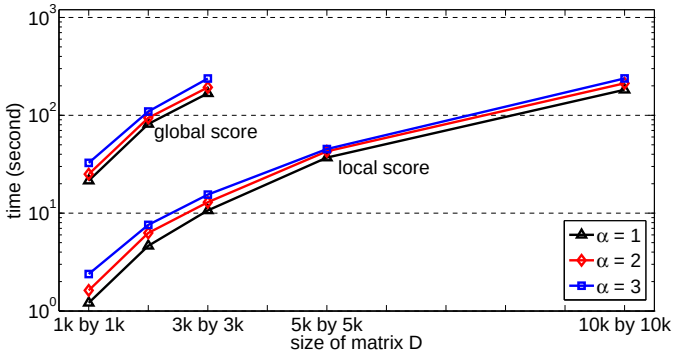


Fig. 9 Scalability comparison between the global and local scores on *Synthetic* data of density $\beta = 0.03$, over 5 entity domains, for resp. $\alpha = 1$, 2, and 3 planted chains.

is more than one order of magnitude faster than the global score; while it generally attains results of equally high quality.

Overall run times are reasonable and within the order of minutes. Moreover, the framework allows for trivial parallelization, e.g., over calculating the scores for all candidates (Line 8 in Algorithm 1).

6.3 Real Data

Next we investigate the performance on real data. To be able to evaluate the discovered chains qualitatively we consider three intelligence analysis datasets: *AltanticStorm* (Hughes, 2005), *Manpad*, and *Crescent* (Hughes, 2005). The task for these datasets is to discover the plots of any imminent threat, arms dealing, or possible terrorist attacks. Note the highly unstructured nature of knowledge discovery in these datasets. We pre-process these datasets in three steps: (i) co-reference resolution, i.e., resolving entities refering to the same object (e.g., a proper noun upon introduction of the entity and a pronoun at a

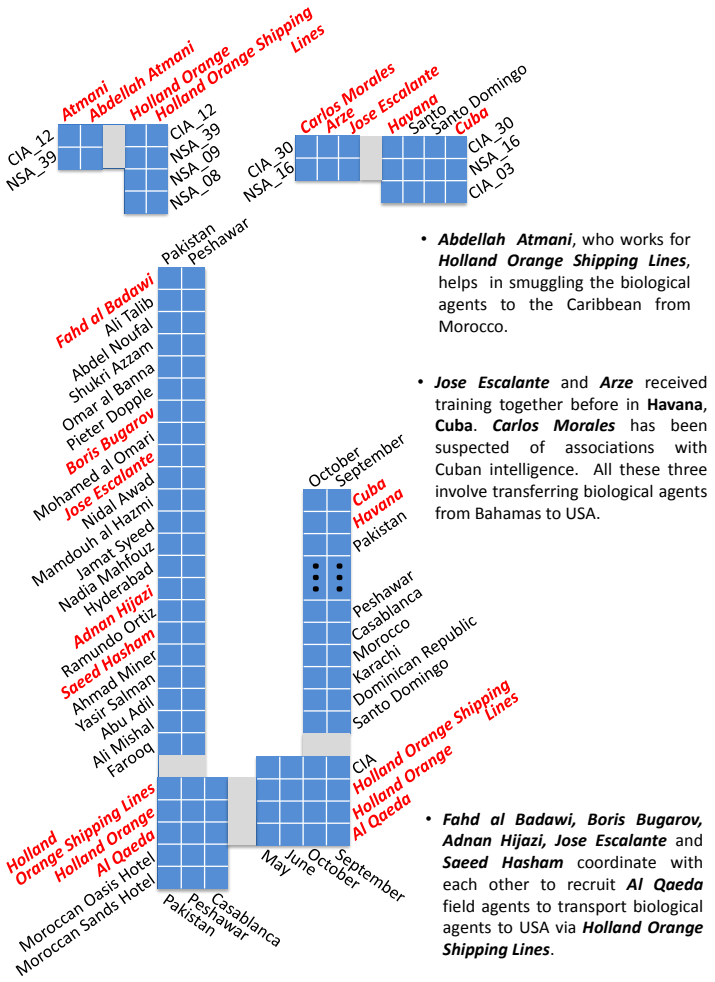
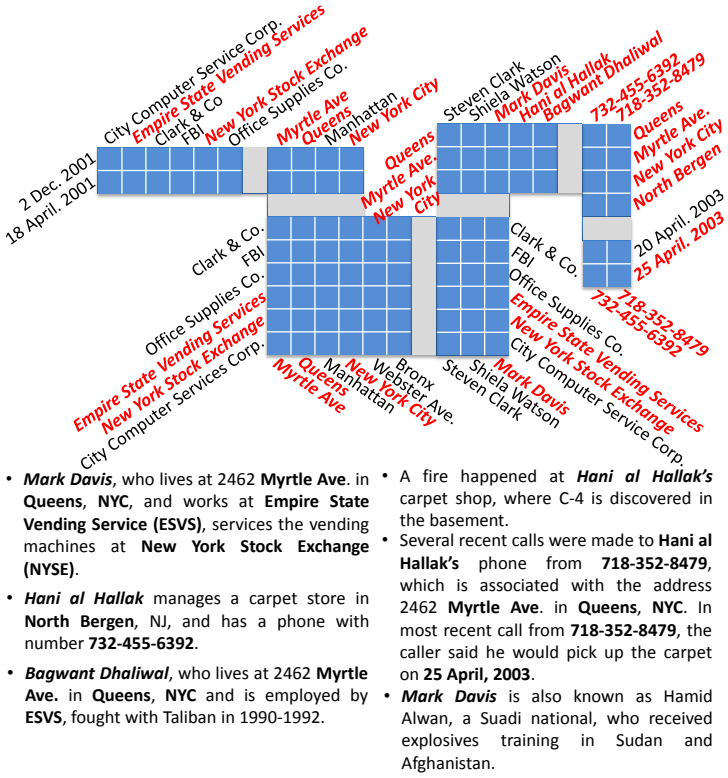


Fig. 10 Top-ranked bicluster chains and related plots for the *AtlanticStorm* dataset. The bottom chain was found using the Entity–Entity model, the top two using the Doc–Entity model. Entities in bold (red) are part of the true solution.

later reference), (ii) entity extraction and classification into categories, using the standard NLP tool AlchemyAPI,[3] and (iii) transforming the data into our Doc–Entity and Entity–Entity data models. To select candidate biclusters in the chain construction process, we here use a Jaccard coefficient of $\varphi = 0.5$, which ensures a wide variety of chains can be discovered from the noisy real datasets.

Figure 10 shows the top few bicluster chains found by our proposed framework on the *AtlanticStorm* dataset. The two small bicluster chains at the top, from the Doc–Entity data model, reveal connections between three persons:

[3] `http://www.alchemyapi.com/`



- ***Mark Davis***, who lives at 2462 **Myrtle Ave**. in **Queens**, **NYC**, and works at **Empire State Vending Service (ESVS)**, services the vending machines at **New York Stock Exchange (NYSE)**.
- ***Hani al Hallak*** manages a carpet store in **North Bergen**, NJ, and has a phone with number **732-455-6392**.
- ***Bagwant Dhaliwal***, who lives at 2462 **Myrtle Ave.** in **Queens**, **NYC** and is employed by **ESVS**, fought with Taliban in 1990-1992.
- A fire happened at ***Hani al Hallak's*** carpet shop, where C-4 is discovered in the basement.
- Several recent calls were made to **Hani al Hallak's** phone from **718-352-8479**, which is associated with the address 2462 **Myrtle Ave**. in **Queens**, **NYC**. In most recent call from **718-352-8479**, the caller said he would pick up the carpet on **25 April, 2003**.
- ***Mark Davis*** is also known as Hamid Alwan, a Suadi national, who received explosives training in Sudan and Afghanistan.

Fig. 11 Top-ranked bicluster chain and related plots for the *Crescent* dataset. Entities in bold (red) are part of the true solution.

Carlos Morales, *Arze* and *Jose Escalante*, and the connections between the person *Abdellah Atmani* and the company *Holland Orange Shipping Lines*. These two bicluster chains lead us to the central plots of the *AtlanticStorm* dataset: namely, that *Jose Escalante*, *Arze*, *Carlos Morales* and *Abdellah Atmani* are involved in transferring biological agents to United States. The large bicluster chain at the bottom of Fig. 10, discovered using the Entity–Entity data model, reveals the connections among the persons *Fahd al Badawi*, *Boris Bugarov*, *Jose Escalante*, *Adnan Hijazi* and *Saeed Hasham* and the organizations of *Holland Orange Shipping Lines* and *Al Qaeda*, which identifies the plot of these five persons to recruit *Al Qaeda* members to transport biological agents to the USA via *Holland Orange Shipping Lines*.

The bicluster chain in Figure 11 is the top one discovered from the *Crescent* dataset under the Entity–Entity data model. This bicluster chain shows the connections among three persons (*Mark Davis*, *Hani al Hallak* and *Bagwant Dhaliwal*), two Companies (*Empire State Vending Services (ESVS)* and *New York Stock Exchange (NYSE)*), and an address (*Myrtle Ave. in Queens, New York City*). It turns out one plot related to a terrorist action in *Crescent* dataset is *Mark Davis*, who has received explosive training before, and *Bagwant*

Table 2 Recall scores on three real datasets for three different chain discovery methods. For these datasets the plots do not form coherent chains, and hence we report and discuss the precision, and the subtle issues regarding this, in Section 6.4.

		Our Approach	
Dataset	BigCluster	InfPair	InfCluster
Atlantic Storm	0.32	0.16	**0.32**
Crescent	0	0.29	**0.86**
Manpad	0.14	0.21	**0.36**

Dhaliwal, who fought with Taliban in 1990-1992, will pick up C-4 bombs from *Hani al Hallak* on *April 25, 2003*, and plan to install the C-4 onto vending machines at *NYSE*. Thus, the bicluster chain shown here helps to uncover this hidden plot in the *Crescent* dataset.

6.4 Baseline Comparisons

Though there exist studies on the general topic of 'finding plots', e.g.,Shahaf and Guestrin (2010, 2012); Hossain et al (2012b); Kumar et al (2006); Hossain et al (2012a), we consider a quite distinct problem setting for which to the best of our knowledge no existing approach is applicable. (See Section 7 for a complete discussion of related work.) To demonstrate the effectiveness of our approach, we hence compare to two baselines.

The first baseline method, BigCluster, follows Algorithm 1 to find bicluster chains yet iteratively chooses to extend the chain with the *largest* bicluster instead of determining subjective interestingness using MaxEnt. As a second baseline we consider a simplified version of our method, we consider InfPair, which performs like InfCluster but iteratively finds the most informative entity pair to add to the chain, as opposed to the most informative bicluster.

We apply the three methods to each of the three real datasets, and consider the recall with regard to entities and the precision with regard to chains—the proportion of the discovered chains related to the dataset solution—for the top-3 discovered chains. We use this definition here, as unlike for the synthetic data in these real datasets the 'plots' do not (necessarily) from one coherent chain. (See Section 8 for a more in-depth discussion of the difficulties of evaluating our problem setting.) We first discuss the recall scores, which we give in Table 2. Overall, our main approach InfCluster consistently identifies the best chains. With its simplified variant InfPair in second place this shows that a good bicluster chain is *not* simply the combination of large biclusters—and that surprisingness is an important aspect to obtain interesting results. As we only consider the top-3 chains discovered in a single iteration, precision is a less interesting metric here—the scores possible per dataset are resp. 0, $\frac{1}{3}$, $\frac{2}{3}$, and 1. Averaged over the three datasets, we find that BigCluster has a precision of 0.33, InfPair a precision of 0.78, and InfCluster a precision

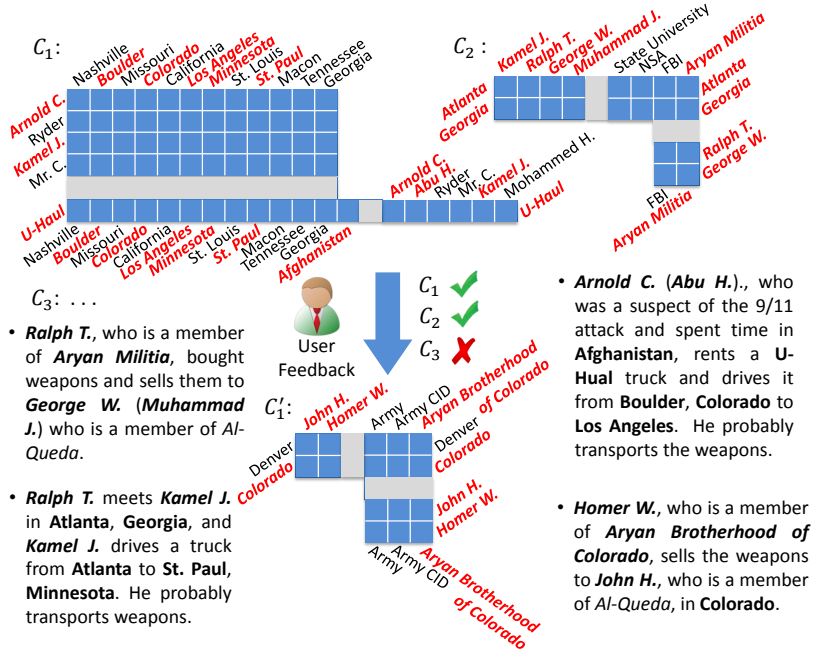


Fig. 12 Iterative mining on the *Manpad* Dataset. Entities in bold (red) are part of the true solution. The top two bicluster chains were selected by the analyst from the top-ranked discovered chains, and then incorporated as background knowledge, leading to discovery of a new bicluster chain at the bottom revealing a sub-plot in the dataset.

of 0.56. The difference between InfPair and InfCluster is explained by InfPair considering much smaller biclusters than InfCluster. Considering both precision and recall InfCluster performs best with a wide margin.

6.5 Case Study for Iterative Human-in-the-Loop Data Mining

Having verified the quality of our framework and InfCluster in particular, we now investigate its performance for iterative human-in-the-loop discovery of interesting bicluster chains in a small case study. To this end we consider the *Manpad* dataset and ask an in-house domain expert to analyze the data with our tool. We give the key results for the first iterations in Figure 12.

We mined the top-3 of bicluster chains and presented these to the expert, whom selected those she finds interesting; here, the two top-ranked bicluster chains, depicted at the top of the figure are the two top-ranked chains discovered in the first iteration, were selected. After identifying that these two bicluster chains reveal the plots of two separate arms dealings in *Boulder*, *Colorado* and *Atlanta*, *Georgia*, and that the weapons are transported to *Los Angeles* in *California* and *St. Paul* in *Minnesota* for potential terrorist attacks, the analysts add these two chains back into the model as part of the

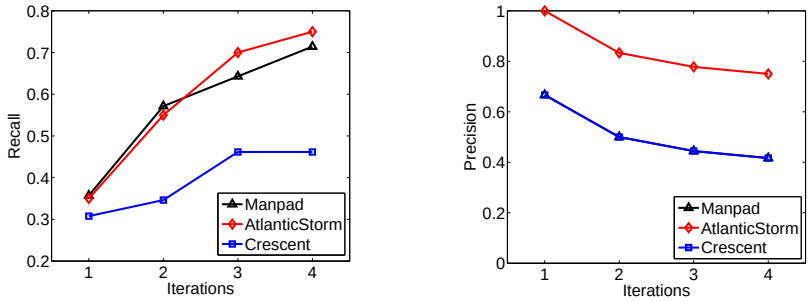


Fig. 13 Recall (left) and precision (right) of true plot entities vs. number of iterations for resp. the *Manpad*, *AtlanticStorm*, and *Crescent* datasets. Please note that in the right panel the lines of *Manpad* and *Crescent* datasets overlap.

background knowledge, and further investigate the dataset to discover other related plots. With the updated background information, our framework recomputes the bicluster chains, and discovers as the top-ranked bicluster chain the one shown at the bottom of Figure 12, revealing a plot involving another arm dealing between the persons *John H.* and *Homer W.* in *Colorado*.

As the iterative discovery process continues, the intelligence analyst finds more and more entities involved in the plots of the intelligence dataset. We give the recall with regard to important entities and precision with regard to chains over the first four iterations in Figure 13 for each the *Manpad*, *AtlanticStorm* and *Crescent* datasets. From the figure we see that the recall of important entities steadily increases with further iterations, indicating that more and more of the entire plots are discovered. Note that while desirable, it is not realistic to expect 100% recall—unless the plot is trivial, stands out very strongly from the background knowledge, and no other significant structures are present in the dataset. Investigating the relatively modest recall scores for *Crescent* we find these are due to key structure of the data is reported in the first few iterations that is unrelated to the plot yet unexplained by the background knowledge.

7 Related Work

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

7.1 Mining Biclusters

Mining biclusters is an extensively studied area of data mining, and many algorithms for mining biclusters from all sorts of data types have been proposed. Tibshirani et al (1999) proposed a backward pruning method to find biclusters

with constant values. While, Califano et al (2000) aimed to find biclusters with constant values on rows or columns, which they defined as σ-valid ks-patterns. Segal et al (2001) and Sheng et al (2003) investigated mining biclusters within a Bayesian framework and using probabilistic relational models. Cheng and Church (2000) proposed an algorithm to find the δ-biclusters—which is defined by the mean squared residue between rows and columns of the cluster. Regarding the biclustering algorithms on binary data, Zaki and Hsiao (2005) proposed an efficient algorithm called CHARM-L for mining frequent closed itemsets and their lattice structure. Uno et al (2005) developed the LCM algorithm that combines the data structures of array, bitmap and prefix trees to efficiently discover frequent as well as closed itemsets. A comprehensive survey of biclustering algorithms was given by Madeira and Oliveira (2004).

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.

7.2 Mining Surprising Patterns

There is, however, significant literature on mining representative/succinct/surprising patterns (e.g., Kiernan and Terzi, 2008) as well as on explicit summarization (e.g., Davis et al, 2009). Wang and Parthasarathy (2006) 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 (2006) showed that querying such a model is PP-hard. Mampaey et al (2012) gave a convex heuristic, allowing more efficient search for the most informative set of patterns. De Bie (2011) 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. 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 (2009). 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 (2007) and evaluate empirical p-values. Tatti and Vreeken (2012) 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 (Dzeroski and Lavrac , editors) with a plethora of approaches ranging from relational association rules (Dehaspe and Toironen, 2000) to inductive logic programming (ILP) (Lavrac and Flach, 2001). The idea of composing redescriptions (Zaki and Ramakrishnan, 2005) and biclusters to form patterns in multi-relational data was first proposed by Jin et al (2008). Cerf et al (2009) introduced DataPeeler algorithm to tackle the challenge of directly discovering closed patterns from n-ary relations in multi-relational data. Later, Cerf et al (2013) 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 (2010) studied randomization techniques for multi-relational databases with the goal to evaluate the statistical significance of database queries. Spyropoulou and De Bie (2011) and Spyropoulou et al (2014) 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 (2013) 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 pairwise 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. Besides that we consider two different data models, another key difference is that we iteratively update our MaxEnt model to include all patterns we mined so far. Mampaey et al (2012) and Kontonasios et al (2013) 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.

7.5 'Finding Plots'

Finally, we give an overview of work on discovering 'plots' in data. Note that 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 (2012) 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., Hossain et al, 2012b; Kumar et al, 2006; Hossain et al, 2012a) 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.

8 Discussion

From the results on synthetic as well as real data, we find that our framework is able to correctly identify highly interesting bicluster chains from unstructured data. Performance is best for the Doc–Entity model, which is partly due to the MaxEnt modelling process adhering closer to this setup, as well as that for the Entity–Entity model it is inherently more complex to find good bi-clusters. Still, the results on real data for both models provide much insight and useful information about the plots hidden in the data.

To score the surprisingness of biclusters our current approach requires that the input data can be transformed into a flat binary table—which we then model by Maximum Entropy. As here we consider text data as our input for us this transformation is straightforward and without loss for the Doc–Entity model. While such transformation into binary data is always possible—we can, e.g., model the adjacency matrix of (un)directed graphs—the more closely the modelling follows the input data, the better. It will be interesting to see whether it is possible to extend our approach to include different MaxEnt models for different relations. With regard to the Entity–Entity model, we identify the need for MaxEnt theory to model integer-valued, or count data as then we can determine the surprisingness of how often entities co-occur. The recent results by Kontonasios et al (2011, 2013) on modelling continuous valued data by MaxEnt seems a natural starting point as the model can incorporate tiles as background knowledge.

While it has very nice theoretic properties, and performs well in practice, calculating our global objective score is computationally expensive. To this end we proposed local approximations, which we found to closely approximate the performance of the global score in practice. To further gain efficiency, one could evaluate candidates in parallel, as well as parallelize inferring the MaxEnt models. However, we should also emphasize here that the focus of this paper is exploratory analysis of multi-relational data, and discovering interesting multi-relational patterns in particular. In this study our main concern was quality; follow-up studies can consider scaling our framework up for application to very large data. With regard to higher quality, we are currently investigating theory and methods for directly discovering surprising biclusters; that is, to

take the current MaxEnt model into account when searching for candidate clusters to add to the chain. Currently, our theory and methods are only directly applicable to discrete binary data, other data types will have to be first transformed to our data models. Here, we used standard NLP tools to transform text into binary relations, without optimizing these towards optimal detectability of the most interesting bicluster chains. It will make for engaging future work to investigate how to optimally transform text for analysis using our framework.

8.1 Doc–Entity vs. Entity–Entity

From the results on the intelligence datasets, we can see that our proposed framework is capable of identifying important hidden plots with associated evidence. We observed that if the dataset contains several key entities that are central to the plot (such as coordinating several activities), the Doc–Entity data model will help us to identify these. The Entity–Entity data model on the other hand reveals more the interactions between different types of entities. Even if some key roles and actions appear only a few times in the data, this approach has the potential to uncover the evidence leading to such key players in the dataset. This is mostly because the Entity–Entity model implicitly puts more on the surprisingness of entities and their coalitions/connections in the dataset, whereas traditional methods, such as measuring support and frequency, cannot really capture. As future work, it may be worthwhile to investigate whether a combination of the two data models is possible, and if this would provide better results.

8.2 Exploration vs. Exploitation

As real datasets are typically rather complex, they may contain many possible 'plots'—hence, it is not guaranteed that all the solution entities will naturally form a single chain without involving other entities. Moreover, for other entities found by our approach that are not in the solution, they cannot be simply determined as important or useless. It really depends on whether these entities are connected to the solution entities. For example, in the bicluster chain shown in Figure 11, the entities *New York Stock Exhange (NYSE)* and *Empire State Vending Services (ESVS)* appear together with the important persons *Mark Davis*, *Hani al Hallak* and *Bagwant Dhaliwal*. This gives the analysts some hints that something suspecious may happen in *NYSE* or *ESVS*. It turns out to be that *Mark Davis* who is an employee at *ESVS* and serves vending machines at *NYSE* plans to get bombs from *Hani al Hallak* and *Bagwant Dhaliwal*, and installs them in the vending machine of *NYSE*. This demonstrates that if such relevant entities (*NYSE*, *ESVS*, etc.) appear together with the important entities in the dataset, they may provide additional information to the intelligence analysts. But, if appearing individually, they may not draw

our attention at all. Thus, we could not simply treat such entities as important or irrelevant entities, which makes the entity precision not be an appropriate evaluation criterion in our scenario. We identify this as a natural aspect of data exploration: our method simply identifies what is surprising in the data with regard to the provided background knowledge, and as long as a discovered chain identifies significant structure—whether about the plot of interest, or explaining another aspect of the data—we regard it as a good result.

9 Conclusion and Future Work

Our approach to discover multi-relational patterns is a significant step in formalizing a previously unarticulated knowledge discovery problem. We have primarily showcased results in intelligence analysis, however, the theory and methods we presented is 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 can be transformed into one of our two data models. That is, data for which 'entities' and/or 'documents' can be identified. We have presented new data modeling primitives, algorithms for extracting patterns, and experimental results on scalability as well as effectiveness of inference.

Some of the directions for future work include (i) obviating the need to mine all biclusters prior to composition, (ii) improving accuracy of estimation when data density becomes larger, (iii) integrating the two data models introduced here, (iv) incorporating weights on relationships, to account for differing veracities and trustworthiness of evidence. Ultimately, the key to a good tool for data analysts is to foster human-in-the-loop knowledge discovery which is one of the key advantages of the methods proposed here.

Acknowledgements Jilles Vreeken is supported by the Cluster of Excellence "Multimodal Computing and Interaction" within the Excellence Initiative of the German Federal Government.

Bibliography

Califano A, Stolovitzky G, Tu Y (2000) Analysis of gene expression microarrays for phenotype classification. In: Proc. Int. Conf. Intell. Syst. Mol. Biol, pp 75–85

Cerf L, Besson J, Robardet C, Boulicaut JF (2009) Closed patterns meet n-ary relations. ACM Transactions on Knowledge Discovery from Data 3(1):3:1–3:36

Cerf L, Besson J, Nguyen KNT, Boulicaut JF (2013) Closed and noise-tolerant patterns in n-ary relations. Data Mining and Knowledge Discovery 26(3):574–619

Cheng Y, Church GM (2000) Biclustering of expression data. In: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology, AAAI Press, pp 93–103

Cover T, Thomas J (2006) Elements of Information Theory. Wiley

Csiszar I (1975) I-Divergence geometry of probability distributions and minimization problems. The Annals of Probability 3(1):pp. 146–158

Darroch JN, Ratcliff D (1972) Generalized iterative scaling for log-linear models. The Annals of Mathematical Statistics 43(5):pp. 1470–1480

Davis WLI, Schwarz P, Terzi E (2009) Finding representative association rules from large rule collections. In: Proceedings of the 9th SIAM International Conference on Data Mining (SDM), Sparks, NV, SIAM, pp 521–532

De Bie T (2011) Maximum entropy models and subjective interestingness: an application to tiles in binary databases. Data Mining and Knowledge Discovery 23(3):407–446

Dehaspe L, Toironen H (2000) Discovery of relational association rules. In: Dĕzeroski S (ed) Relational Data Mining, Springer-Verlag New York, Inc., pp 189–208

Dzeroski S, Lavrac (editors) N (2001) Relational Data Mining. Springer, Berlin

Geerts F, Goethals B, Mielikainen T (2004) Tiling databases. In: Proceedings of Discovery Science, Springer, pp 278–289

Gionis A, Mannila H, Mielikäinen T, Tsaparas P (2007) Assessing data mining results via swap randomization. ACM Transactions on Knowledge Discovery from Data 1(3):167–176

Hanhijärvi S, Ojala M, Vuokko N, Puolamäki K, Tatti N, Mannila H (2009) Tell me something I don't know: randomization strategies for iterative data mining. In: Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Paris, France, ACM, pp 379–388

Hossain M, Gresock J, Edmonds Y, Helm R, Potts M, Ramakrishnan N (2012a) Connecting the dots between PubMed abstracts. PLoS ONE 7(1)

Hossain MS, Butler P, Boedihardjo AP, Ramakrishnan N (2012b) Storytelling in entity networks to support intelligence analysts. In: Proceedings of the 18th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Beijing, China, ACM, pp 1375–1383

Hughes FJ (2005) Discovery, Proof, Choice: The Art and Science of the Process of Intelligence Analysis, Case Study 6, "All Fall Down", unpublished report

Jaynes ET (1957) Information theory and statistical mechanics. Physical Review Series II 106(4):620–630

Jin Y, Murali TM, Ramakrishnan N (2008) Compositional mining of multirelational biological datasets. ACM Transactions on Knowledge Discovery from Data 2(1):2:1–2:35

Kiernan J, Terzi E (2008) Constructing comprehensive summaries of large event sequences. In: Proceedings of the 14th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Las Vegas, NV, pp 417–425

Kontonasios KN, Vreeken J, De Bie T (2011) Maximum entropy modelling for assessing results on real-valued data. In: Proceedings of the 11th IEEE International Conference on Data Mining (ICDM), Vancouver, Canada, IEEE, pp 350–359

Kontonasios KN, Vreeken J, De Bie T (2013) Maximum entropy models for iteratively identifying subjectively interesting structure in real-valued data. In: Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), Prague, Czech Republic, Springer, pp 256–271

Kumar D, Ramakrishnan N, Helm RF, Potts M (2006) Algorithms for storytelling. In: Proceedings of the 12th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Philadelphia, PA, pp 604–610

Lavrac N, Flach P (2001) An Extended Transformation Approach to Inductive Logic Programming. ACM Transactions on Computational Logic Vol. 2(4):pages 458–494

Madeira SC, Oliveira AL (2004) Biclustering algorithms for biological data analysis: A survey. IEEE/ACM Trans Comput Biol Bioinformatics 1(1):24–45

Mampaey M, Vreeken J, Tatti N (2012) Summarizing data succinctly with the most informative itemsets. ACM Transactions on Knowledge Discovery from Data 6:1–44

Ojala M, Garriga GC, Gionis A, Mannila H (2010) Evaluating query result significance in databases via randomizations. In: Proceedings of the 10th SIAM International Conference on Data Mining (SDM), Columbus, OH, pp 906–917

Rasch G (1960) Probabilistic Models for Some Intelligence and Attainnment Tests. Danmarks paedagogiske Institut

Rissanen J (1978) Modeling by shortest data description. Automatica 14(1):465–471

Schwarz G (1978) Estimating the dimension of a model. The Annals of Statistics 6(2):461–464

Segal E, Taskar B, Gasch A, Friedman N, Koller D (2001) Rich probabilistic models for gene expression. Bioinformatics 17(suppl 1):S243–S252

Shahaf D, Guestrin C (2010) Connecting the dots between news articles. In: Proceedings of the 16th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Washington, DC, ACM, pp 623–632

Shahaf D, Guestrin C (2012) Connecting two (or less) dots: Discovering structure in news articles. ACM Transactions on Knowledge Discovery from Data 5(4):24:1–24:31

Sheng Q, Moreau Y, De Moor B (2003) Biclustering microarray data by gibbs sampling. Bioinformatics 19(suppl 2):ii196–ii205

Spyropoulou E, De Bie T (2011) Interesting multi-relational patterns. In: Proceedings of the 11th IEEE International Conference on Data Mining (ICDM), Vancouver, Canada, pp 675–684

Spyropoulou E, De Bie T, Boley M (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, pp 217–232

Spyropoulou E, De Bie T, Boley M (2014) Interesting pattern mining in multi-relational data. Data Min Knowl Discov 28(3):808–849

Tatti N (2006) Computational complexity of queries based on itemsets. Information Processing Letters 98(5):183–187, DOI http://dx.doi.org/10.1016/j.ipl.2006.02.003

Tatti N, Vreeken J (2012) Comparing apples and oranges - measuring differences between exploratory data mining results. Data Mining and Knowledge Discovery 25(2):173–207

Tibshirani R, Hastie T, Eisen M, Ross D, Botstein D, Brown P (1999) Clustering methods for the analysis of dna microarray data. Tech. rep., Stanford University

Uno T, Kiyomi M, Arimura H (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, ACM, New York, NY, USA, OSDM '05, pp 77–86

Wang C, Parthasarathy S (2006) Summarizing itemset patterns using probabilistic models. In: Proceedings of the 12th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Philadelphia, PA, pp 730–735

Zaki M, Hsiao CJ (2005) Efficient algorithms for mining closed itemsets and their lattice structure. IEEE Transactions on Knowledge and Data Engineering 17(4):462–478

Zaki MJ, Ramakrishnan N (2005) Reasoning about sets using redescription mining. In: Proceedings of the 11th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Chicago, IL, ACM, pp 364–373