arXiv:1608.03889v1 [cs.DB] 12 Aug 2016

# Interactive and Iterative Discovery of Entity Network Subgraphs

Hao Wu, Maoyuan Sun, Jilles Vreeken, Nikolaj Tatti, Chris North, Naren Ramakrishnan

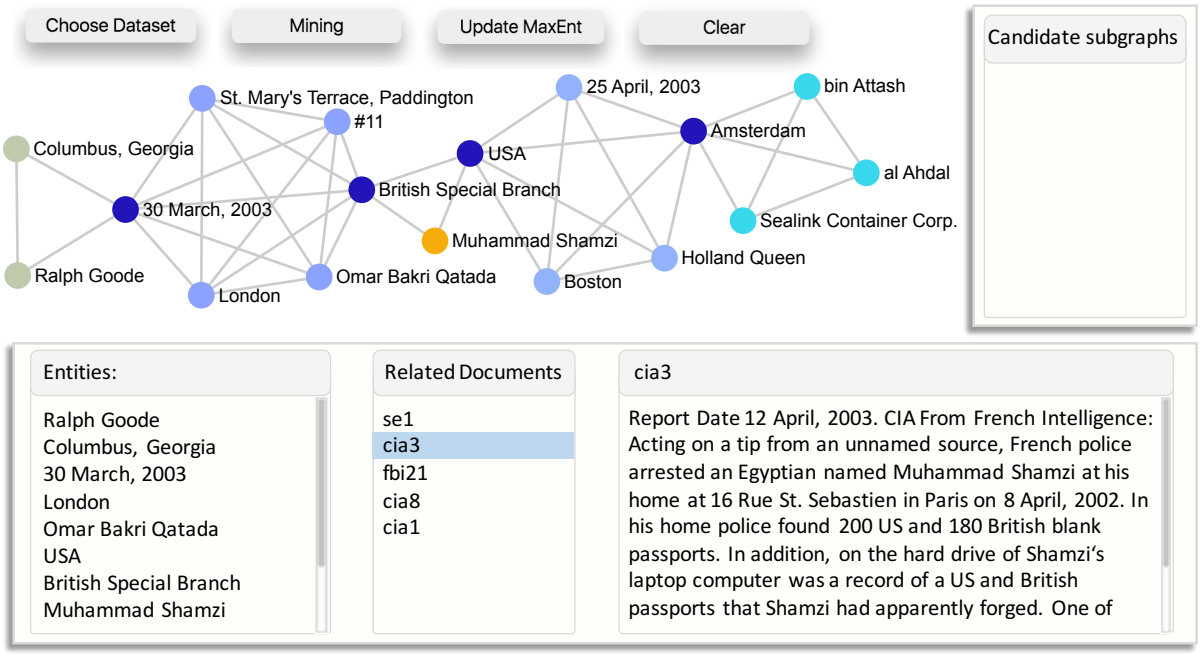



Fig. 1. An overview of the user interface for the proposed interactive and iterative visualization framework to discover entity network subgraphs. A connected subgraph pattern identified from an intelligence analysis dataset is displayed in this example. The involved entities and corresponding intelligence documents are listed below the displayed pattern.

**Abstract**— Graph mining to extract interesting components has been studied in various guises, e.g., communities, dense subgraphs, cliques. However, most existing works are based on notions of frequency and connectivity and do not capture subjective interestingness from a user's viewpoint. Furthermore, existing approaches to mine graphs are not interactive and cannot incorporate user feedbacks in any natural manner. In this paper, we address these gaps by proposing a graph maximum entropy model to discover surprising connected subgraph patterns from entity graphs. This model is embedded in an interactive visualization framework to enable human-in-the-loop, model-guided data exploration. Using case studies on real datasets, we demonstrate how interactions between users and the maximum entropy model lead to faster and explainable conclusions.

## 1 Introduction

Knowledge discovery from graphs is a crutial task that arises from many reasearch and application domains, e.g. social network analysis, biological knowledge discovery, and storytelling from unstructured text documents. As the fast development of Internet technology and social media, social networks like Facebook and Twitter have drawn much attention in both industry and academic community. Identifying interesting group structures, e.g. communities, in such social networks can help sociologists to understand people's social behaviors in the virtual world and compare with that of the real world. From an industrial perspective, subgraph patterns in such social network can help companies deliver their advertisements more precisely and recommand their products and services to potential clients over the Internet. In biology domain, questions like "how do these pathways interact and influence with each other in bological pathway networks?" are typical challenges faced by biologists in their research. Similarly, in storytelling from text datasets, analysts are always interested in interactions between various types of entities, e.g. persons and locations, in entity networks, which could lead the analysts to discover the plots or even conspiracies hidden behind phenomenons.

To address these challenges, a plenty of graph mining algorihms which extract local patterns, such as communities [15], dense subgraphs [31] and cliques, e.g. [14], from networks have been proposed and studied in the recent decade. However, most of such existing works focus on investigating the notions of density or connectivivty to indentify the subgraph patterns and do not capture the subjective interestingness from a user's perspective. Moreover, except for few works, e.g. Apolo [4], most of the existing graph mining approaches are purely algorithmic without involving any algorithm-user interactions, which ignores the user's important feedbacks.

To overcome such drawbacks, in this paper, we fill these gaps by proposing a graph Maximum Entropy (MaxEnt) model together with a subjective interestingness measure to discover interesting connected subgraph patterns from entity graphs. By designing a greedy heuristic algorithm that works together with the proposed graph MaxEnt model, we achieve the same goal of automatically discovering subgraph patterns from graphs compared to the traditional graph mining algorithms. In additional, by embedding the graph MaxEnt model into an interactive visualization framework, we enable the iterative, human-in-the-loop, model-guided data exploration. The key point we would like to emphasize here is that the ultimate objective of knowledge discovery is not to extract an unique answer from the dataset but rather to guide domain experts into deeper consideration and understanding of key process elements.

Our contributions in this paper are:

1. We derive and present the formalization of the graph MaxEnt model, and define a criterion that measures the interestingness of local subgraph patterns based on the MaxEnt model.
2. We design a greedy heuristic algorithm to automatically discover the interesting connected subgraph patterns from entity graphs.
3. By integrating the graph MaxEnt model with a visualization framework, we enable the model-guided, interactive and iterative data exploration and knowledge discovery over graphs.
4. Using the results and a use case study on real world intelligence datasets, we demonstrate how our proposed graph MaxEnt model and visualization framework are adopted to analyze real world datasets. In particular, we show how our approach supports human-in-the-loop knowledge discovery, and leads the analysts to uncover the hidden plots of the datasets.

## 2 PRELIMINARIES

In this section, we will introduce some preliminary concepts that will be useful and helpful to understand our proposed algorithm in the rest of this paper.

### 2.1 Graph Notations

A graph is usually defined as an ordered tuple $G = (V, E)$, where $V$ represents a set of vertices in the graph, and $E$ denotes a set of edges that connect vertices in $V$. Usually, a tuple of two vertices that an edge connects is used to represent this edge. A graph can be further classified as directed graph or undirected graph depending on whether the direction of edges matters. If we let $v \in V$ denote a single vertex in the graph, in the directed graph, an edge $(v_i, v_j)$ from $v_i$ to $v_j$ is different from the edge $(v_j, v_i)$ from $v_j$ to $v_i$. However, in the undirected graph, they are the same. In this paper, we will focus on the directed graph when we describe the proposed model since the undirected graph can be seen as a special case of the directed graph, e.g. if an edge exists between vertices $v_i \in V$ and $v_j \in V$, both edges $(v_i, v_j)$ and $(v_j, v_i)$ belong to the edge set $E$. Thus, in the rest of the paper, all the graphs mentioned will refer to the directed graph unless specified.

A subgraph of $G$ is a graph $G' = (V', E')$ such that $V' \subseteq V$ and $E' \subseteq E$. We use $G' \subseteq G$ to represent the subgraph relationship. Among all the possible subgraphs of $G$, a clique $C = (V_c, E_c)$ is a special type of subgraph such that $\forall v_i, v_j \in V_c$ and $v_i \neq v_j$, the edge $(v_i, v_j) \in E_c$. In other words, a clique is a subgraph that is complete. A maximal clique is a clique that cannot be extended by adding more vertices into it.

In terms of the connected subgraph, we refer to a tuple of ordered subgraphs $CS = (G_1, G_2, \ldots, G_n)$ such that the adjacent subgraphs $G_i$ and $G_{i+1}$ share at least one common vertex $v$. In this case, we say the common vertex $v$ connect the subgraph $G_i$ and $G_{i+1}$.

### 2.2 MaxEnt Models

The Maximum Entropy (MaxEnt) principle [24] has drawn much attention in the pattern mining community recently, especially in the area of discovering subjectively interesting patterns. The concept of entropy is originated from the information theory [39]. In the context of data mining, entropy is adopted to measure how certain a model is about the data. Lower entropy indicates the model is quite certain about the data it models. It would be perfect if we find a low entropy model where the model summarizes the majority information conveyed by the data we are modeling. However, the given prior information about the data is usually quite limited in practice. Inferring a low entropy model may require us to make additional assumptions about the data, which is unreasonable due to the lack of support in the prior knowledge of the data. In addition, making such unreasonable assumptions would not guarantee that the resulting model is able to capture the actual characteristics of the data. Thus, the only reasonable choice would be avoiding such unreasonable assumptions and only relying on the given prior information about the data although it would increase the entropy of the model and make the model more uncertain about the data. This is exactly what the Maximum Entropy principle addresses. Generally speaking, MaxEnt principle identifies the best probability distribution, which maximizes the entropy, over the dataset at hand given the prior knowledge about the data. The result MaxEnt probability distribution uses the prior information optimally and best summarizes the dataset, and is unbiased otherwise.

To be more specific, suppose we have a dataset $\boldsymbol{D}$ and a set of functions $\mathcal{F} = \{f_i \mid f_i(\boldsymbol{D}) = S_i\}$ that compute several statistics about the dataset $\boldsymbol{D}$. Such statistics will serve as the prior information or background knowledge about the dataset. Using this prior information as constraints, it defines a set of probability distributions $\mathcal{P}$ that are consistent with the given dataset statistics, e.g. $\mathcal{P} = \{p \mid \mathbb{E}_p[f_i(\boldsymbol{D})] = S_i, \forall f_i \in \mathcal{F}\}$ where $\mathbb{E}_p[\cdot]$ denotes the expectation under the probability distribution $p$. Among all these possible probability distributions, MaxEnt principle identifies the distribution $p^*$ which maximizes the entropy,

$$p^* = \operatorname*{argmax}_{p \in \mathcal{P}} H(p) \ .$$

Here, $H(p)$ represents the entropy of probability distribution $p$.

### 2.3 Subjective Interestingness

In the data mining and knowledge discovery, the aim is to uncover the highly informative information with respect to the prior knowledge or what we have already known about the data — we are not quite interested in what we already know or what we can trivially infer from such knowledge.

To this end, we introduce the concept of subjective interestingness or surprisingness. Suppose we have a probability distribution $p$ which models our current beliefs about the data. When we evaluate the data mining results, we can use $p$ to determine the likelihood of a result under our current knowledge about the data. If the likelihood is high, this indicates that we probably already know about this result, or we can easily infer such result. Thus, reporting it would provide little novel information about the data. On the contrary, if the likelihood is very low, the result could be very interesting or surprising, which means it conveys a lot of new information compared to what we have already known. In Section 4, we will formal define a quantitative criterion to measure the subjective interestingness. In the rest of this paper, we will assume that the two words interesting and surprising refer to the same concept in our context and use them interchangeably.

## 3 MAXENT MODEL ON GRAPH

In this section, we will define the MaxEnt model over graphs. Although there exist several other generative models for graphs, e.g. the classic stochastic block models [1] which are well studied and widely used to recover community structures in graphs, we choose to adopt the MaxEnt framework here due to its natural fit for the interactive and iterative knowledge discovery scenario in this paper. Before we formally state the MaxEnt model, we first describe some basic statistics about graphs, which will serve as the prior information about the graph. Then, we will introduce the MaxEnt model over graphs by applying the Principle of Maximum Entropy, and finally, we will describe how we can estimate the graph MaxEnt model by maximizing the likelihood.

### 3.1 Notations for Graph Prior Knowledge

In our scenario, we choose to use in-degrees and out-degrees of vertices and subgraphs to characterize the prior knowledge of a given graph $G = (V, E)$. The in-degrees and out-degrees of vertices are types of graph statistics that describe the given graph from a global perspective. For the purpose of convenience, we normalized the in-degree and out-degree of a vertex into the range of $[0, 1]$ by dividing the total number of vertices in the graph. In the rest of this paper, we will use the terms in-degree and out-degree to refer to the normalized in-degree and out-degree for each vertex. On the other hand, subgraphs identify local information about graphs, which could be useful as prior knowledge about the local structures of graphs. Although

there are many statistics available for subgraphs, we choose the densities of subgraphs to characterize the local structures of the entire graph, which is defined as:

$$f(G') = \frac{|E'|}{|V'|^2}, \text{ where } G' = (V', E'), \text{ and } G' \subseteq G,$$

where $|V'|$ and $|E'|$ represent the number of vertices and edges in the subgraph $G'$, respectively. Here, notice that we are considering a more general scenario that the edge from a vertex to itself is allowed.

Let $d_{in}(v)$ and $d_{out}(v)$ to represent the in-degree and out-degree of a vertex $v$, and $f(G')$ denote the density of the subgraph $G'$. Suppose $\mathcal{G}$ is the space that contains all the possible graphs that have $|V|$ vertices. Let $p$ be the probability distribution defined over the graph space $\mathcal{G}$, then the expectation of in-degree and out-degree of a given vertex $v$ and the expectation of the density of a given subgraph $G'$ would be:

$$\mathbb{E}_p\left[d_{in}(v)\right] = \sum_{G \in \mathcal{G}} p(G) d_{in}(v)$$

$$\mathbb{E}_p\left[d_{out}(v)\right] = \sum_{G \in \mathcal{G}} p(G) d_{out}(v)$$

$$\mathbb{E}_p\left[f(G')\right] = \sum_{G \in \mathcal{G}} p(G) f(G')$$

### 3.2 MaxEnt Model with Prior Information

In this section, we will derive a global statistical model for graphs based on the given graph prior knowledge. The graph prior information is provided in the form of vertex degrees and the densities of various subgraphs as we discussed in Section 3.1. For a given graph $G = (V, E)$, suppose we are given a set of vertex degree constraints $\mathcal{D} = \{d_{in}(v_i) = D_i^{in}, d_{out}(v_i) = D_i^{out} \mid v_i \in V\}$ and a set of subgraph density constraints $\mathcal{F} = \{f(G') = F_{G'} \mid G' \subseteq G\}$. Notice that the subgraph constraints $\mathcal{F}$ may not necessarily contain every possible subgraph of $G$, which is also infeasible in practice. We would like to infer a probability distribution $p$ over the space of all possible graphs $\mathcal{G}$ that is consistent with information given by the constraints $\mathcal{D}$ and $\mathcal{F}$. In other words, we want to determine how likely is a graph $G \in \mathcal{G}$ given these vertex degree and subgraph density constraints $\mathcal{D}$ and $\mathcal{F}$.

In order to derive a good statistical model, we adopt a principled and statistically well-founded approach. We employ the MaxEnt principle introduction in Section 2.2. To formally define the MaxEnt distribution, we first need to specify the space $\mathcal{P}$ that contains all the graph probability distribution candidates which conform the given prior information. Given the constraints $\mathcal{D}$ and $\mathcal{F}$ as prior knowledge, the graph probability distribution space can be defined as:

$$\mathcal{P} = \{p \mid \mathbb{E}_p[d_{in}(v_i)] = D_i^{in}, \mathbb{E}_p[d_{out}(v_i)] = D_i^{out}, \\ \mathbb{E}_p[f(G')] = F_{G'}, \forall d_{in}(v_i), d_{out}(v_i) \in \mathcal{D}, \forall f(G') \in \mathcal{F}\}.$$

Among all these candidate distributions, we choose the distribution $p^*$ which maximizes the entropy $H(p)$. To infer the MaxEnt distribution, we rely on a classical theorem in [7] which states that a distribution $p^*$ is the MaxEnt distribution if and only if it can be written as an exponential form. In our scenario, the MaxEnt distribution would be:

$$p^*(G) \propto \exp\left(\sum_{d_{in}(v_i) \in \mathcal{D}} \lambda_i^{in} d_{in}(v_i) + \sum_{d_{out}(v_i) \in \mathcal{D}} \lambda_i^{out} d_{out}(v_i) + \sum_{f(G') \in \mathcal{F}} \lambda_{G'} f(G')\right). \quad (1)$$

Here, $\lambda_i^{in}$, $\lambda_i^{out}$ and $\lambda_{G'}$ are the model parameters.

By rearranging the terms within the summations, we could further factorize the MaxEnt distribution $p^*$ into the product of a number of Bernoulli distributions:

$$p^*(G) = \prod_{v_i, v_j \in V} p^*(\mathbb{I}[(v_i, v_j) \in E]),$$

**Algorithm 1:** Iterative Scaling algorithm for estimating MaxEnt distribution over graphs

**input** : Graph $G = (V, E)$, vertex degree constraints $\mathcal{D} = \{d_{in}(v_i) = D_i^{in}, d_{out}(v_i) = D_i^{out} \mid v_i \in V\}$, and subgraph constraints $\mathcal{F} = \{f(G') = F_{G'} \mid G' \subseteq G\}$.
**output**: MaxEnt distribution $p^* \leftarrow p$.

$p \leftarrow$ a uniform distribution where $p(\mathbb{I}[(v_i, v_j) \in E] = 1) = \frac{1}{2}, \forall v_i, v_j \in V$;
**while** *not converged* **do**
  **for** $d.(v_i) \in \mathcal{D}$ **do**
    $h \leftarrow \mathbb{E}_p[d.(v_i)]$, $\tilde{h} \leftarrow D_i^{\cdot}$;
    $x \leftarrow \frac{\tilde{h}(1-h)}{h(1-\tilde{h})}$;
    $p(\mathbb{I}[(v_i, v_j) \in E] = 1) \leftarrow \frac{x \cdot p(\mathbb{I}[(v_i, v_j) \in E] = 1)}{1 - (1-x) \cdot p(\mathbb{I}[(v_i, v_j) \in E] = 1)}$, for all $v_j \in V$;
  **end**
  **for** $f(G') \in \mathcal{F}$ **do**
    $h \leftarrow \mathbb{E}_p[f(G')]$, $\tilde{h} \leftarrow F_{G'}$;
    $x \leftarrow \frac{\tilde{h}(1-h)}{h(1-\tilde{h})}$;
    $p(\mathbb{I}[(v_i, v_j) \in E] = 1) \leftarrow \frac{x \cdot p(\mathbb{I}[(v_i, v_j) \in E] = 1)}{1 - (1-x) \cdot p(\mathbb{I}[(v_i, v_j) \in E] = 1)}$, for all $v_i, v_j \in V', G' = (V', E')$;
  **end**
**end**
**return** $p$;

where

$$p^*(\mathbb{I}[(v_i, v_j) \in E] = 1) = \frac{\exp\left(\sum_{\lambda_i \in \Lambda} \lambda_i\right)}{\exp\left(\sum_{\lambda_i \in \Lambda} \lambda_i\right) + 1}, \text{ or } 0, \ 1.$$

Here, $\mathbb{I}[(v_i, v_j) \in E]$ is an indicator function which equals to 1 if the edge $(v_i, v_j)$ belongs to the edge set $E$ of the graph $G = (V, E)$, otherwise 0. $\Lambda = \{\lambda_{G'} | \forall G' = (V', E'), f(G') \in \mathcal{F}, v_i, v_j \in V'\} \cup \{\lambda_i^{out}, \lambda_j^{in}\}$ is a set of model parameters $\lambda_{G'}$ where the corresponding subgraphs $G'$ that are used to provide the prior information of the graph $G$ contain the specific vertices $v_i$ and $v_j$, plus the model parameters $\lambda_i^{out}$ and $\lambda_j^{in}$ for the out-degree and in-degree constraints of the vertices $v_i$ and $v_j$, respectively.

### 3.3 MaxEnt Distribution Estimation

In order to estimate the parameters of the MaxEnt distribution mentioned in the previous section, e.g. $\lambda_{G'}$, $\lambda_i^{in}$ and $\lambda_i^{out}$, we follow a standard approach and adopt the well-known Iterative Scaling (IS) algorithm [8] to infer the MaxEnt model over graphs. Algorithm 1 describes the detail of this IS algorithm where $d.(v_i)$ denotes either $d_{in}(v_i)$ or $d_{out}(v_i)$. Similarly, $D_i^{\cdot}$ denotes either $D_i^{in}$ or $D_i^{out}$. Generally speaking, for each vertex degree constraint $d.(v_i) \in \mathcal{D}$ or subgraph density constraint $f(G') \in \mathcal{F}$, the IS algorithm updates the probability distribution $p$ such that the expectation of the vertex degree or subgraph density under distribution $p$ will be consistent with the given value in the corresponding constraint. Obviously, during such a single update, we may change the expected vertex degree or subgraph density corresponding to other constraints. Thus, several iterations are needed until the probability distribution $p$ converges. The proof of the convergence for the IS algorithm is beyond the scope of this paper. Readers who are interested in this topic, please refer to the Theorem 3.2 in [7]. In practice, it typically takes on the order of seconds for the IS algorithm to converge.

## 4 Interactive Discovery of Connected Subgraphs

In this section, we illustrate our visualization framework that discovers connected subgraphs from an entity graph with an interactive manner.

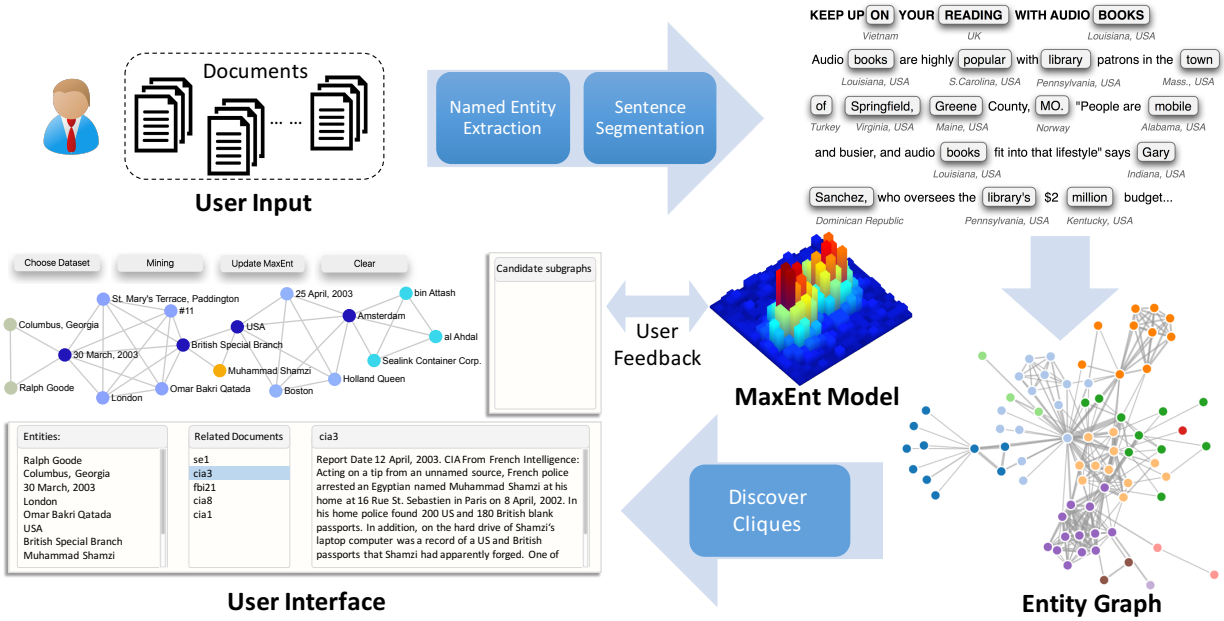



Fig. 2. The architecture of the proposed connected subgraph discovery framework. The user input is a text corpus with a collection of documents. After named entity extraction and sentence segmentation, an entity graph is created based on the entity co-occurrence on the sentence level. Then the graph MaxEnt model is inferred over the entity graph, and the subgraphs (cliques in our scenario) are identified from the entity graph. By iteratively interacting with the MaxEnt model through the user interface, interesting connected subgraph patterns could be discovered from the entity graph.

Figure 2 shows the architecture of the proposed interactive visualization framework. Although our proposed framework can be used to analyze graphs originated from various types of raw data, e.g. biology data, social networks, we focus on the entity graph constructed from text corpus in this paper. In this scenario, the user input is a collection of text documents from which the entity graph is constructed (Section 4.1). With the entity graph, we infer a background MaxEnt model (Section 4.2) and define an interestingness measure (Section 4.3), which will be used to automatically discover connected subgraphs (Section 4.4) or guide the user exploration process (Section 4.5).

### 4.1 Creating Entity Graph

Briefly speaking, we construct the entity graph based on the entity co-occurrences on the sentence level in the text corpus. To be more specific, given a text corpus $TC = \{doc_1, doc_2, \ldots, doc_n\}$, we perform the sentence segmentation on each document $doc_i \in TC$ to split each document into a set of sentences, e.g. $doc_i = \{sent_{i,1}, sent_{i,2}, \ldots, sent_{i,m}\}$, and also extract named entities from each document $doc_i$.

With the extracted named entities and segmented sentences from the text corpus, we can build the undirected entity graph $G = (V, E)$ with the following approach. The vertex set $V$ of the graph is just the set of all the extracted named entities. For the edge set $E$, if two named entities $\epsilon_i$ and $\epsilon_j$ appear together in some sentence $sent_{k,l}$ in the document $doc_k$, we add an undirected edge $(\epsilon_i, \epsilon_j)$ into $E$. Although the entity graph created here is undirected, as we mentioned in Section 2.1, an undirected graph can be treated as a special case of a directed graph. Thus, our graph MaxEnt formalization can be easily extended to the scenario of undirected graphs.

### 4.2 Background MaxEnt Model

Next, given the entity graph, we discuss how to specify the background MaxEnt model which incorporates the basic prior knowledge about the given graph. In order to discover non-trial connected subgraphs, we need some basic background information about the entity graph and infer a background MaxEnt model so that we can evaluate the surprisingness of patterns (connected subgraphs here) as we discussed in Section 2.3. In our scenario, we choose to use the degree of each vertex in the graph as the prior knowledge to infer the background MaxEnt model. Formally, for the entity graph $G = (V, E)$, the prior information that used as constraints to infer the background MaxEnt model is $\mathcal{D} = \{d(v_i) = D_i \mid v_i \in V\}$. Recall the form of the MaxEnt distribution described in Equation (1), the background MaxEnt model $p_{back}$ in this scenario would be:

$$p_{back}(G) \propto \exp\left(\sum_{d(v_i)\in\mathcal{D}} \lambda_i d(v_i)\right),$$

and it can be inferred with the Iterative Scaling algorithm described in Algorithm 1.

### 4.3 Interestingness Measure

In order to determine the interestingness or surprisingness of any subgraph pattern $G'$ discovered from the graph $G = (V, E)$, we propose an interestingness measure that characterizes how much new information the subgraph $G'$ conveys with respect the background MaxEnt model $p_{back}(G)$. We will do this by inferring two MaxEnt models, and then compute the divergence between these two models.

With the background MaxEnt model as described in Section 4.2, we need another MaxEnt model which also incorporates the new information given by the subgraph $G'$ under consideration, e.g. the MaxEnt model inferred with the prior information $\mathcal{D} = \{d(v_i) = D_i \mid v_i \in V\}$ and the given subgraph density information $\{f(G') = F_{G'}\}$. We call such MaxEnt model $p_{G'}$. Then, the interestingness measure we would like to propose is defined as:

$$s(G') = KL(p_{G'} || p_{back}). \quad (2)$$

Here, $KL$ denotes the Kullback-Leibler (KL) divergence [5] which is well studied, easy to compute, and fits our modeling requirements. In theory, other divergence measures could also be considered. With the KL divergence, larger $s(G')$ indicates that more new information is brought into the MaxEnt model $p_{G'}$ by the subgraph $G'$, thus $G'$ would be more interesting compared to other subgraph patterns.

### 4.4 Automatic Connected Subgraph Discovery

In this section, we describe a greedy heuristic strategy to automatically discover interesting connected subgraph patterns from the entity graph

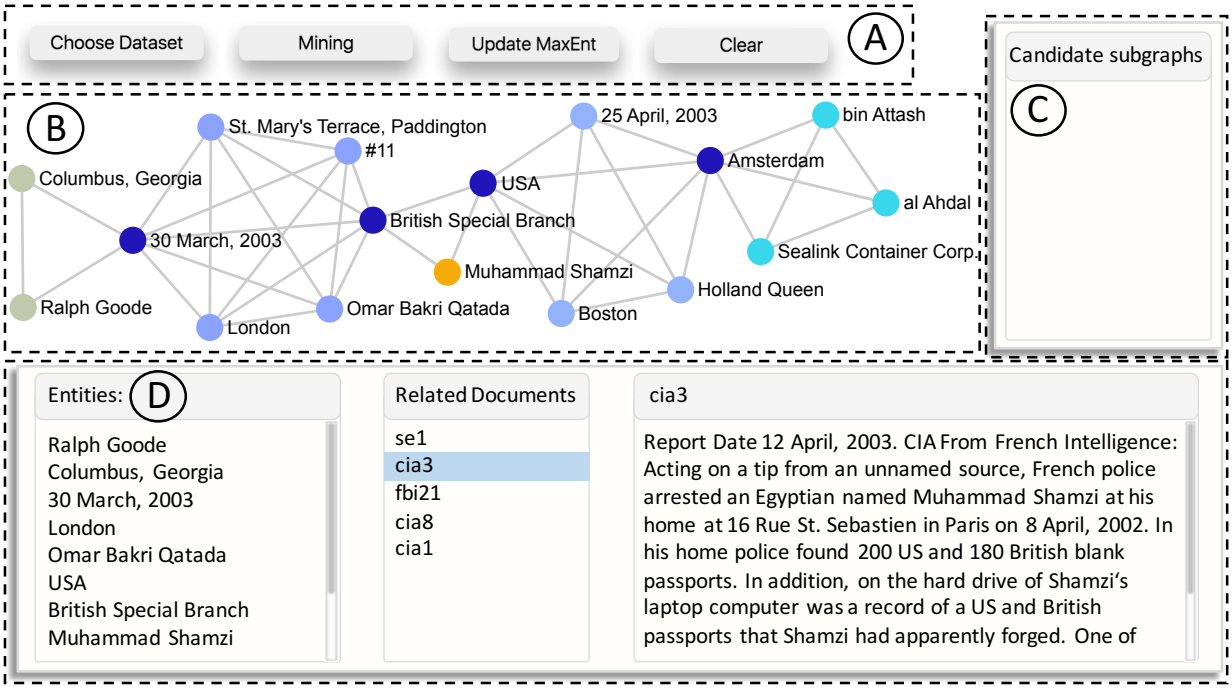



Fig. 3. The layout of the visualization interface. Region (A) contains the functional buttons where the user can load datasets, execute the automatic connected subgraph discovery, update the MaxEnt model, and clear the displayed results. Region (B) displays the current connected subgraph pattern under investigation. Region (C) lists the candidate subgraphs that can extend the displayed connected subgraph pattern in the user centric exploration process. Region (D) shows the corresponding entities and documents involved in the displayed connected subgraph pattern.

**Algorithm 2:** Greedy Heuristic for Connected Subgraph Discovery

**input** : background MaxEnt model $p_{back}$;
entity graph $G = (V, E)$;
$K$, the desired number of connected subgraphs.
**output**: the set of connected subgraph $\mathcal{CS}$.

1 $\mathcal{CS} \leftarrow \emptyset$;
2 $\mathcal{G}' \leftarrow$ `discoverSubgraphs(`$G$`)`;
3 **while** $(|\mathcal{CS}| < K)$ **do**
4 $\quad CS \leftarrow \arg\max_{G' \in \mathcal{G}'} s(G')$;
5 $\quad \mathcal{G}'_c \leftarrow$ `candidateSubgraph(`$\mathcal{G}'$, $CS$`)`;
6 $\quad$ **while** $|\mathcal{G}'_c| \neq 0$ **do**
7 $\quad\quad G'_n \leftarrow \arg\max_{G' \in \mathcal{G}'_c} s(G')$;
8 $\quad\quad CS \leftarrow$ `extend(`$CS$, $G'_n$`)`;
9 $\quad\quad \mathcal{G}'_c \leftarrow$ `candidateSubgraph(`$\mathcal{G}'$, $CS$`)`;
10 $\quad$ **end**
11 $\quad p_{back} \leftarrow$ `UpdateMaxEntModel(`$p_{back}$, $CS$`)`;
12 $\quad \mathcal{CS} \leftarrow \mathcal{CS} \cup \{CS\}$;
13 **end**
14 **return** $\mathcal{CS}$;

$G$ with the MaxEnt model and the interestingness measure defined above. Ideally, to discover a set of interesting connected subgraphs, we could exhaustively explore the entire search space, find all the possible connected subgraphs, evaluate their interestingness with the criterion defined in Section 4.3, and choose the top $K$ candidates. However, the entire search space for the connected subgraphs could be quite large when the subgraph patterns discovered from the graph $G$ is huge. In this case, the intuitive exhaustive exploration approach would be very inefficient or even infeasible in practice. Moreover, the search space does not exhibit a particular structure which we could leverage to perform an efficient search. Hence, we turn to heuristics.

We adopt a simple iterative greedy search strategy to automatically discover interesting connected subgraph patterns. Algorithm 2 illustrates our proposed heuristic greedy search approach. Given an entity graph $G = (V, E)$, we first discover a set of subgraph patterns from $G$ (Line 2). Various subgraph patterns have been studied in the realm of graph mining, e.g. communities, dense subgraphs, cliques. In our text corpus analysis scenario, we choose to use maximal cliques [14] as our subgraph patterns. To discover a connected subgraph pattern, we start from the most interesting clique with respect to the interestingness measure defined in Equation (2) (Line 4). We believe that more interesting cliques are more likely to form an interesting connected subgraph pattern, and thus help to reveal useful information in the entity graph and in the corresponding text corpus. Then we extend the current connected subgraph pattern $CS$ by greedily choosing the most interesting clique from all the cliques that have at least one common vertex with the first or last clique in the current pattern $CS$. This process continues until we cannot find any other cliques to add into the current connect subgraph pattern $CS$ (Line 6—10). After we find a connected subgraph pattern, we update the background MaxEnt model with the new information in the discovered connected subgraph, e.g. each clique $G'$ in this connected subgraph together with its density $f(G')$, in order to avoid discovering redundant patterns in the future iterations (Line 11). The algorithm keeps discovering such surprising connected subgraphs until the desired number of patterns are found or no more connected subgraphs can be formulated.

### 4.5 User Centric Exploration

Although we have proposed automatic approach to discover the interesting connected subgraph patterns from the entity graph as described in the last section, we should not overlook the important role that user feedbacks play in exploratory data mining tasks. In such scenarios, even if we have criteria to evaluate the data mining results, we may still need a domain expert to verify the results. If we enable an interactive scenario during the data mining process, the data mining model can help the domain expert filter and refine the large amount of patterns. While, on the other hand, the domain expert could also verify the data mining results, update the data mining model, and lead the model to the correct parts of the data to explore. That is the real power of interactive, human-in-the-loop data mining approach.

Having realized the advantages of such human-in-the-loop data

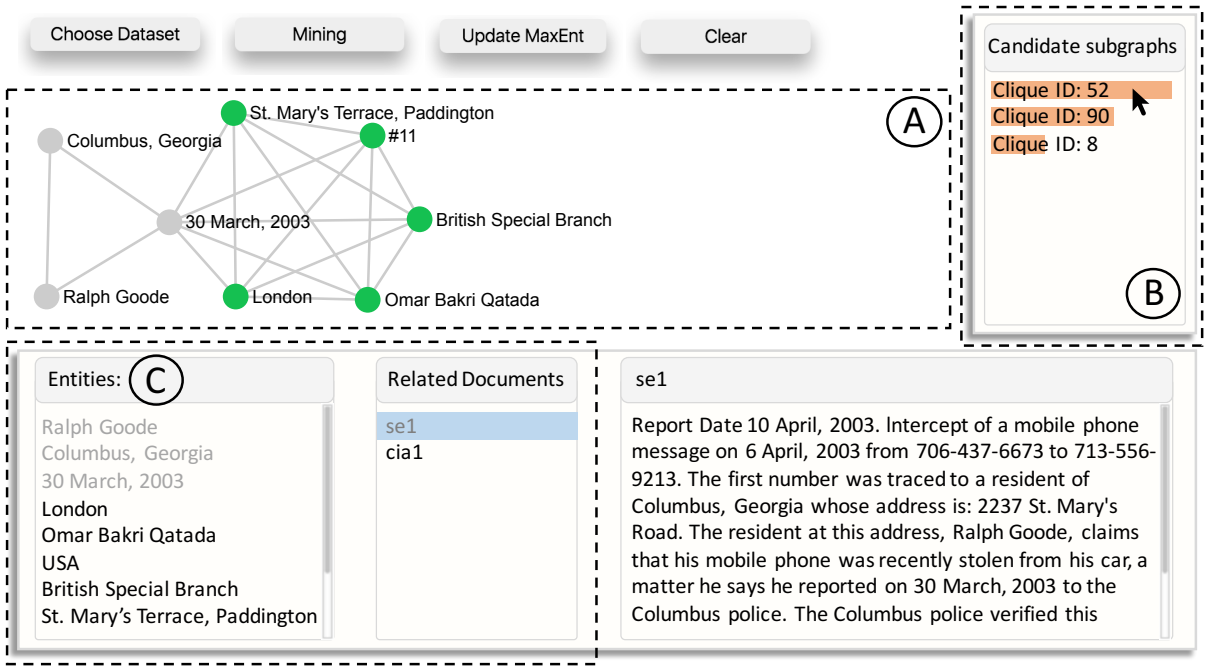



Fig. 4. A snapshot of the user centric exploration process. In the first step, the user chose the clique displayed in green color in Region (A). In the current step, three candidate subgraphs are listed in Region (B) with the most interesting one on the top based on the evaluation with the graph MaxEnt model. When the user selects one candidate subgraph from the list, it is displayed in Region (A) with gray color, and shows how the chosen subgraph is connected to the current pattern. At the same time, related entities and documents are also listed in Region (C) in gray color.

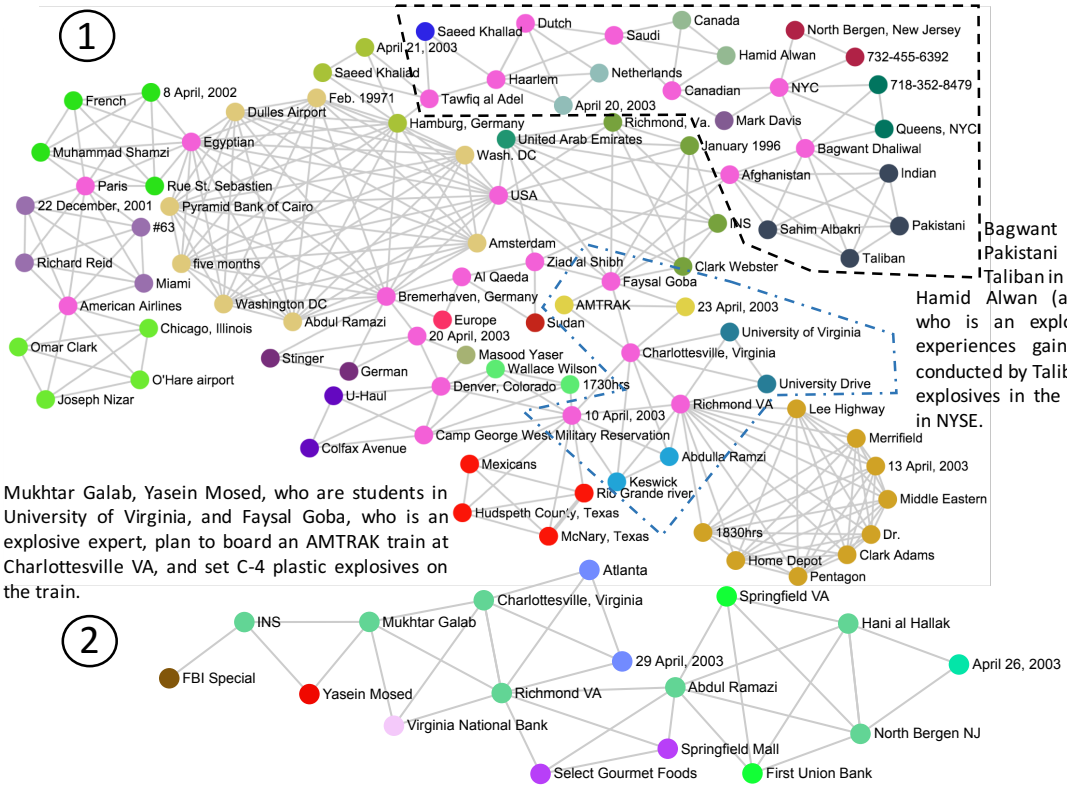
mining approach, in this section, we proposed an interactive, user centric approach with a visualization interface to discover the connect subgraph patterns from the entity graph with the assistance of the MaxEnt model. Figure 3 shows the visualization interface that the analyst uses to communicate with the MaxEnt model. Region (A) contains functionalities through which the user can choose the dataset to explore (*Choose Dataset* button), automatically discover the connected subgraph patterns using the greedy heuristic discussed in Section 4.4 (*Mining* button), update the MaxEnt model with the discovered connected subgraph (*Update MaxEnt* button), and clear the results currently displayed (*Clear* button). Region (B) displays the current discovered connected subgraph pattern with color encodings. The vertices within the same clique are displayed with the same color, and all the vertices that connect adjacent cliques are also displayed using the same color, e.g. the vertices *30 March, 2003*, *British Special Branch*, *USA*, and *Amsterdam* in the displayed connected subgraph of Figure 3. Region (C) shows a list of candidate subgraph patterns that can be used to extend the current connected subgraph pattern. The different candidate subgraph patterns in the list are identified by their unique IDs. In the example shown in Figure 3, the displayed connected subgraph is fully extended, thus this candidate subgraph list is empty. Finally, region (D) displays the named entities involved in the current connected subgraph pattern as well as the corresponding text documents in the corpus.

The user centric data exploration process works in the following way. By clicking the button *Choose Dataset*, the user can select the dataset he wants to explore. By clicking the *Mining* button, an automatic discovery of the connected subgraph patterns will be performed using the greedy heuristic search strategy as discussed in Section 4.4. In the user centric exploration, the user is able to choose which subgraph he would like to use to extend the current connect subgraph pattern. Figure 4 shows a snapshot of this user centric exploration process. Region (A) displays the current incomplete connected subgraph pattern, and Region (B) lists all the candidate subgraphs that can be used to extend this incomplete connected subgraph pattern. Notice that the candidate subgraphs listed here are sorted based on their interestingness measures (Equation (2)) evaluated by the embedded MaxEnt model. The length of the orange bar indicates the interestingness value of the corresponding candidate subgraph [50]. The most interesting candidate subgraph is listed on the top. When the user moves the mouse over a specific candidate subgraph in the list, it will be displayed the in Region (A) in gray color showing how the chosen subgraph extends the current incomplete connected subgraph pattern. The entities and text documents corresponding to the chosen subgraph are also displayed with gray color in the entity list and related document list in Region (C). However, the subgraph in such status is not actually added into the current incomplete connected subgraph pattern. To add the chosen subgraph, the user needs to click the specific candidate subgraph listed in Region (B). By performing such click operation, the chosen candidate subgraph will finally be used to extend the current incomplete connected subgraph pattern, and be displayed in solid color as shown in Figure 3. The corresponding candidate subgraph list, entity list and related document list will also be updated.

Whenever the user thinks the connected subgraph pattern he is exploring is informative enough, or the current connected subgraph pattern is fully extended (no other candidate subgraph can be added), the user can click the *Update MaxEnt* button to update the embedded graph MaxEnt model with the discovered connected subgraph pattern. Such update operation is performed in the same way, e.g. using each subgraph in the connected subgraph pattern and its corresponding density, as we described in Section 4.4 for automatic connected subgraph discovery with the iterative greedy heuristic. By updating the embedded graph MaxEnt model with the discovered connected subgraph pattern, the new information is incorporated into the MaxEnt model so that the duplicate patterns and redundant information will not be identified again in the future iterations of the user centric exploration process over the same dataset. By clicking the *Clear* button, the user can clear the results displayed for the current iteration of the user centric exploration, and start a new iteration of the analysis for the same dataset. However, if the update MaxEnt operation is not performed before this clear operation, the new information contained in the last discovered connected subgraph pattern will not be incorporated into the embedded graph MaxEnt model, and thus be disregarded.

## 5 Experimental Results

In this section, we describe the experimental results over some real world text datasets, primarily text corpora from intelligence analysis domain. Although as stated before, the proposed framework is widely




Bagwant Dhaliwal, who is a Pakistani and served with Taliban in Afghanistan, and Hamid Alwan (a.k.a Mark Davis), who is an explosive expert with experiences gained in operations conducted by Taliban, plan to put C-4 explosives in the vending machines in NYSE.

Mukhtar Galab, Yasein Mosed, who are students in University of Virginia, and Faysal Goba, who is an explosive expert, plan to board an AMTRAK train at Charlottesville VA, and set C-4 plastic explosives on the train.

Fig. 5. Top connected subgraphs discovered by the greedy heuristic search strategy from *Crescent* dataset. These two patterns help the intelligence analysts uncover two potential terrorist attacks from the dataset: (1) *Bagwant Dhaliwal and Mark Davis plan to set explosives in the vending machines in NYSE*; (2) *Mukhtar Galab et al. are going to blow up an AMTRAK train with C-4 plastic explosives*.

applicable to many other types of data where an entity graph can be created. We focus our experimental investigations on answering the following questions:

1. How to preprocess the text corpus so that the entity graph can be constructed? (Section 5.1)
2. How does the iterative greedy heuristic search strategy perform with respect to discovering interesting connected subgraphs when comparing to the true hidden plots of the text datasets? (Section 5.2)
3. How the visualization framework with embedded graph MaxEnt model could help and lead the user to identify the plots of the text corpus in the user centric exploration scenario? (Section 5.3)

### 5.1 Dataset Processing

Given a text corpus, in order to construct the entity graph, we preprocess the text documents in the corpus with natural language processing software from Basis Technology [3]. Sentence segmentation and named entity extraction are performed on each document in the corpus to split each document into a set of sentences and extract named entities from each document. A co-reference operation is also performed over the extracted named entities so that we can merge the named entities that refer to the same object into a single unique entity. Then, we construct the entity graph based on the named entity co-occurrences in sentences as we described previously in Section 4.1 with igraph library [6].

### 5.2 Results on Intelligence Datasets

In this section, we show the results of adopting the proposed iterative greedy heuristic search strategy described in Section 4.4 to a real world intelligence analysis dataset *Crescent* [22]. By providing a collection of intelligence documents, the task for the *Crescent* dataset is to try to discover any imminent threats or possible terrorist attacks by analyzing the intelligence documents. Through searching the entity graph with the greedy heuristic strategy, a large connected subgraph pattern which is the most interesting one with respect to the background MaxEnt model defined in Section 4.2 is discovered (pattern ① in Figure 5). By reading through the corresponding intelligence documents, we notice that the top right part of this connected subgraph pattern (surrounded by the black dash line) helps us identify several pieces of useful evidence:

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

By connecting such evidence together, we are probably going to draw the conclusion that an potential terrorist attack to the *New York Stock Exchange* (*NYSE*) will happen soon.

This connected subgraph pattern help us uncover one of the plots hidden in this *Crescent* dataset. By updating the background MaxEnt model with this pattern, we incorporate the information conveyed by this pattern into the background MaxEnt model. After the update, the greedy heuristic search algorithm identify another interesting connected subgraph pattern as shown by the pattern ② in Figure 5. By checking corresponding intelligence documents in the dataset, we discover the following important pieces of evidence:

- *Mukhtar Galab and Yasein Mosed who have been enrolled at*

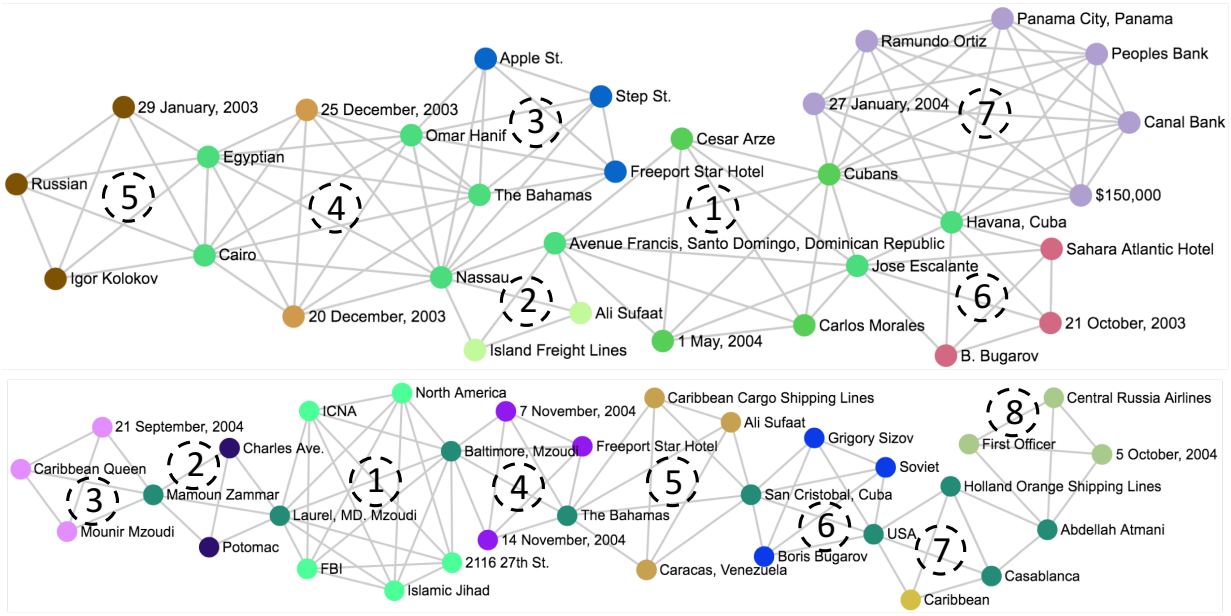



Fig. 6. Two connected subgraph patterns discovered by the user centric data exploration process from the *Atlantic Storm* dataset. The numbers in dashed circles indicate the order that the connected subgraph pattern is extended following the interactions between the analyst and the graph MaxEnt model. These two patterns lead the analyst to reveal the hidden plot that *Boris Bugarov et al. are trying to smuggle biological agents via Caribbean area to USA by using the Holland Orange Shipping Lines*.

*University of Virginia was discovered holding expired student visas , and they have not attended any classes for the past two semesters.*

- *Mukhtar Galab, Yasein Mosed and Faysal Goba reserved three one-way first class tickets for AMTRAK Train #19 from Charlottesville, VA to Atlanta, GA on 29 April, 2003. In addition, Faysal Goba received explosive training in Sudan in 1994 with Al Qaeda.*
- *Abdul Ramazi sent money of $13,000 and $8,500 to Mukhtar Galab and Hani al Hallak of North Bergen, NJ respectively. Muhammed bin Harazi served with Taliban from 1987 to 1993, and entered USA in March, 1993 with an alias name Abdul Ramazi.*

Remember that one piece of evidence revealed by the previous connected subgraph is *Hani al Hallak manages a carpet store in North Bergen, NJ, and a fire happend at his carpet shop where C-4 explosives were discovered in the basement*. Combining with this evidence, we could connect these dots together, and make the hypothesis that Abdul Ramazi purchases C-4 explosives through Hani al Hallak and send the explosives to Mukhtar Galab, Yasein and Faysal Goba who will probably set these explosives on the AMTRAK Train #19 from Charlottesville, VA to Atlanta, GA on 29 April, 2003.

Here, we should emphasize that by incorporating the new information contained in the first discovered connected subgraph pattern (pattern ① in Figure 5) into the background MaxEnt model, we successfully identify another connected subgraph pattern which leads to a second potential terrorist attack hidden in the dataset. This indicates that the newly discovered pattern contains almost no redundant information and is surprising with respect what we already know. Such results demonstrate that our proposed graph MaxEnt model and greedy heuristic search strategy combined together are able to discover interesting and surprising connected subgraph patterns from entity graphs, which additionally leads the analysts to identify the plots that may hide in the dataset. On the other hand, by iteratively identifying interesting connected subgraph patterns and updating the graph MaxEnt model with the discovered patterns, we can incorporate the knowledge we have just learned into the MaxEnt model, and guarantee to discover no redundant information in the future iterations of knowledge discovery. That's the most fascinating part of iterative data mining with MaxEnt models in the scenario of data exploration and knowledge discovery, identifying only interesting results with respect to what we have already learned and at the same time reducing the redundancy as much as possible.

### 5.3 Use Case Study

In this section, we demonstrate the user centric data exploration with our proposed graph MaxEnt model guided visualization framework by showing a use case study on another real world intelligence dataset *Atlantic Storm*. The goal of the *Atlantic Storm* dataset is to try to identify any potential illegal international weapon transportation from a given collection of intelligence documents. Figure 6 shows two connected subgraph patterns discovered by the analyst through the user centric data exploration process.

The analyst starts the exploration process by loading the *Atlatic Storm* dataset into the visualization framework. Then, from the entity graph constructed from the *Atlantic Storm* dataset, the system discovers all the maximal cliques from the entity graph, and assigns an unique ID to each of the maximal clique. By looking through a few maximal cliques ranked on the top by the graph MaxEnt model and the corresponding intelligence documents, the analyst chooses a maximal clique with six vertices (the clique marked with number 1 in dashed circle in the top connected subgraph pattern in Figure 6) to start his exploration of the *Atlantic Storm* dataset since it contains three person entities and one location entity. Usually in intelligence analysis, these two types of entities play an important role in uncovering the hidden plots. After choosing the starting maximal clique, the system updates the candidate maximal cliques that can be used to extend the current connected subgraph pattern, and display them in the user visualization interface for the analyst to select. This time, the analyst chooses the maximal clique with four vertices (the clique marked with number 2 in dashed circle) to extend the current connected subgraph pattern. By repeating these steps, the analyst successfully extends the current pattern to contain seven maximal cliques discovered from the entity graph. Although this connected subgraph pattern can still be further extended, the analyst thinks the remaining candidate maximal cliques are not informative enough to be added into the current pattern. Thus, he decides to end the current iteration of exploration and update the graph MaxEnt model with this connected subgraph pattern (the top connected subgraph pattern shown in Figure 6). However, with this single connected subgraph pattern, the analyst thinks the evidence provided by this pattern is not enough to draw any conclusion, thus he decides to start a new iteration of exploration.

By repeating the exploration process described in the last paragraph, the analyst identifies another connected subgraph pattern as shown at the bottom of Figure 6, which contains eight maximal cliques (the numbers indicate the order by which the pattern is extended). Notice that the connected subgraph pattern at the bottom contains several entities that also appear in the pattern on the top, e.g. *Ali Sufaat*, *The Bahamas*, and *Boris Bugarov* (refer to the same person with *B. Bugarov*), which indicates that these two connected subgraph patterns may describe the same hidden plot of the dataset. By reviewing the intelligence documents related to these two connected subgraph patterns shown in Figure 6 and connecting the pieces of evidence together, the analyst identifies the following possible illegal biological agent transportation:

> *Boris Bugarov and Jose Escalante, with the help of Abdellah Atmani who works for Holland Orange Shipping Lines, coordinate with each other to recruit Al Qaeda field agents to transport biological agents to Caribbean area via Holland Orange Shipping Lines. Jose Escalante, Cesar Arze and Carlos Morales are involved in transferring biological agents from Bahamas to USA.*

Through this use case study, such results demonstrate that with the embedded graph MaxEnt model in the visualization framework, good candidate subgraphs are provided to the analyst in the user centric data exploration process so that interesting connected subgraph patterns can be identified, which further serves as informative hints to lead to the hidden plots in the intelligence datasets.

## 6 Related Work

**MaxEnt models** have drawn much attention recently in the pattern mining community, especially on the topic of mining representative/succinct/surprising patterns, e.g. [25], as well as explicit summarization [9, 34]. Wang and Parthasarathy [47] summarized a collection of frequent patterns by using a row-based MaxEnt model, heuristically mining and adding the most important itemsets in a level-wise fashion. Tatti [43] showed that querying such a model is PP-hard. Mampaey et al. [33] gave a convex, MaxEnt model based heuristic, allowing more efficient search for the most informative set of patterns. De Bie [10] systematically formalized how to model a binary matrix by the MaxEnt principle using row and column margins as background knowledge. Tatti and Vreeken [44] compared the informativeness of data mining results given by different approaches over the same data by applying the binary MaxEnt model. Spyropoulou et al. [41, 42], Spyropoulou and De Bie [40] formally defined Maximal Complete Connected Subset (MCCS) patterns, and proposed to use the K-partite graph and the MaxEnt model to discover surprising MCCS patterns from multi-relational data with $n$-ary relationships. Kontonasios et al. [26, 27] introduced a real-valued MaxEnt model and proposed a subjective interestingness measure called *Information Ratio* to iteratively discover the interesting structures in real-valued data.

**Iterative data mining** was first introduced by Hanhijärvi et al. [17]. The basic idea is to iteratively identify the results from the data, which are most significant given our accumulated knowledge about the data. To assess significance, they built upon the swap-randomization approach of Gionis et al. [16] and evaluated empirical p-values. Kontonasios et al. [27] and Mampaey et al. [34] demonstrated separately that ranking data mining results with a static MaxEnt model leads to redundancies among the high-ranked patterns, and the iterative data mining methodology provides a principled approach to keep the data mining model updated and thus avoid such type of redundancy.

**Storytelling** or **finding plots** from text corpus is another exploratory data mining task that has been extensively studied in recent a few years. By finding a chain of intermediate articles that are maximally coherent given either a start or end-point article, Shahaf and Guestrin [38] studied the problem of summarizing a large collection of news articles by identifying a chain of main events. Storytelling algorithms [19, 28, 18] provide algorithmic frameworks to automatically connect pieces of evidence which may scatter into various different text documents and reveal the stories hidden in the dataset. Wu et al. [48] proposed a MaxEnt based framework to identify the plots by detecting non-obvious coalitions of entities from multi-relational datasets and further support iterative, human-in-the-loop, knowledge discovery. Ning et al. [36] adopted a document similarity based storytelling algorithm to discover story chains from news articles, and studied the relationships between the identified story chains and the tweeters that belong to the same topics.

**Group structure visualization in graphs** has been a well studied research topic in the visualization community. A lot of techniques have been proposed to visualize the group structures in the node-link diagram representation of graphs. Colors are often used to explicitly visualize the group membership of vertices. Vertices that belong to only a single group are simply colored with a unique color to represent the membership [13]. For the scenario that vertices can belong to multiple groups, one approach is to represent vertices by pie charts with sections filled by corresponding colors that denote the groups the vertices belong to [23, 32, 35]. Vehlow et al. [45] proposed to use different size of the pie chart sections to encode the fuzzy membership degrees, while for crisp overlapping communities, Xu et al. [49] used glyphs to encode group overlap by integrating various visual channels, e.g. intensity of color, hue, size, and shape. Some other approaches optimize the color assignment to maximize the color differences between the adjacent neighbor groups [20, 30]. The group structures could also be visualized using contours within the graph, e.g. vertices within the same contour belong to the same group. The contour shape could be rectangles [37], circles [29], convex hulls [2], and arbitrary two-dimensional curves or splines [2, 21, 11, 12]. However, group structure visualization in graphs is not our primary focus here in this paper. Readers who are interested in this topic would find a comprehensive survey of related work in [46].

## 7 Conclusion

In this paper, we formally introduce the graph MaxEnt model, which is a significant step to bring the well-found Maximum Entropy principle into the graph data. With the proposed graph MaxEnt model, we study the problem of discovering surprising connected subgraph patterns from entity graphs, which could help to uncover interesting facts hidden inside the graphs. By designing a MaxEnt model embedded visualization framework, we illustrate how the model guided, human-in-the-loop iterative data mining process can help the exploratory data mining tasks. Although we primarily focus on demonstrating our proposed approach by showing the results over text datasets from intelligence analysis, the theories and methods we present here are also applicable to graphs originated from other types of data in general, e.g. biology and social network data. Possible directions for future work may include improving the efficiency of MaxEnt model estimation and the design of visualization interface, e.g. adopting better color encoding and layout algorithm to display the connected subgraph patterns, to better support the interactive visual analytics of large graphs.

## Bibliography


[1] E. Abbe and C. Sandon. Community detection in general stochastic block models: fundamental limits and efficient recovery algorithms, 2015. arXiv:1503.00609.

[2] B. Bach, E. Pietriga, and J. D. Fekete. Graphdiaries: Animated transitions and temporal navigation for dynamic networks. *IEEE Transactions on Visualization and Computer Graphics*, 20(5): 740–754, May 2014.

[3] Basis Technology, Cambridge, MA. Rosette Big Text Analytics, 2016. http://www.basistech.com/text-analytics/.

[4] D. H. Chau, A. Kittur, J. I. Hong, and C. Faloutsos. Apolo: Making sense of large network data by combining rich user interaction and machine learning. In *CHI '11*, pages 167–176, 2011.

[5] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley, 2006.

[6] G. Csardi and T. Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006. http://igraph.org.

[7] I. Csiszár. I-divergence geometry of probability distributions and minimization problems. *Annals of Probability*, 3(1):146–158, 1975.

[8] J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43(5): pp. 1470–1480, 1972.

[9] W. L. I. Davis, P. Schwarz, and E. Terzi. Finding representative association rules from large rule collections. In *SDM'09*, pages 521–532. SIAM, 2009.

[10] T. De Bie. Maximum entropy models and subjective interestingness: an application to tiles in binary databases. *Data Min. Knowl. Discov.*, 23(3):407–446, 2011.

[11] K. Dinkla, M. J. van Kreveld, B. Speckmann, and M. A. Westenberg. Kelp diagrams: Point set membership visualization. *Computer Graphics Forum*, 31(3pt1):875–884, 2012.

[12] K. Dinkla, M. El-Kebir, C.-I. Bucur, M. Siderius, M. J. Smit, M. A. Westenberg, and G. W. Klau. examine: Exploring annotated modules in networks. *BMC Bioinformatics*, 15(1):1–14, 2014.

[13] C. Dunne and B. Shneiderman. Motif simplification: Improving network visualization readability with fan, connector, and clique glyphs. In *CHI '13*, pages 3247–3256, 2013.

[14] D. Eppstein, M. Löffler, and D. Strash. Listing all maximal cliques in sparse graphs in near-optimal time. In O. Cheong, K.-Y. Chwa, and K. Park, editors, *Algorithms and Computation: 21st International Symposium, ISAAC 2010, Proceedings, Part I*, pages 403–414. Springer, 2010.

[15] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010.

[16] A. Gionis, H. Mannila, T. Mielikäinen, and P. Tsaparas. Assessing data mining results via swap randomization. *TKDD*, 1(3): 167–176, 2007.

[17] S. Hanhijärvi, M. Ojala, N. Vuokko, K. Puolamäki, N. Tatti, and H. Mannila. Tell me something I don't know: randomization strategies for iterative data mining. In *proceeding of KDD '09*, pages 379–388. ACM, 2009.

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

[19] M. S. Hossain, P. Butler, A. P. Boedihardjo, and N. Ramakrishnan. Storytelling in entity networks to support intelligence analysts. In *KDD'12*, pages 1375–1383, 2012.

[20] Y. Hu, E. R. Gansner, and S. Kobourov. Visualizing graphs and clusters as maps. *IEEE Computer Graphics and Applications*, 30 (6):54–66, Nov 2010.

[21] Y. Hu, S. G. Kobourov, and S. Veeramoni. Embedding, clustering and coloring for dynamic maps. *Journal of Graph Algorithms and Applications*, 18(1):77–109, 2014.

[22] F. J. Hughes. Discovery, Proof, Choice: The Art and Science of the Process of Intelligence Analysis, Case Study 6, "All Fall Down". Unpublished report, 2005.

[23] T. Itoh, C. Muelder, K. L. Ma, and J. Sese. A hybrid space-filling and force-directed layout method for visualizing multiple-category graphs. In *PacificVis '09*, pages 121–128, April 2009.

[24] E. T. Jaynes. Information theory and statistical mechanics. *PhysRevII*, 106(4):620–630, 1957.

[25] J. Kiernan and E. Terzi. Constructing comprehensive summaries of large event sequences. In *KDD'08*, pages 417–425, 2008.

[26] K.-N. Kontonasios, J. Vreeken, and T. De Bie. Maximum entropy modelling for assessing results on real-valued data. In *ICDM'11*, pages 350–359, 2011.

[27] K.-N. Kontonasios, J. Vreeken, and T. De Bie. Maximum entropy models for iteratively identifying subjectively interesting structure in real-valued data. In *proceeding of ECMLPKDD '13*, pages 256–271. Springer, 2013.

[28] D. Kumar, N. Ramakrishnan, R. F. Helm, and M. Potts. Algorithms for storytelling. In *KDD'06*, pages 604–610, 2006.

[29] G. Kumar and M. Garland. Visual exploration of complex time-varying graphs. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):805–812, Sept 2006.

[30] A. Lambert, F. Queyroi, and R. Bourqui. Visualizing patterns in node-link diagrams. In *International Conference on Information Visualisation*, pages 48–53, July 2012.

[31] V. E. Lee, N. Ruan, R. Jin, and C. Aggarwal. A survey of algorithms for dense subgraph discovery. pages 303–336. Springer US, 2010.

[32] S. Liu, X. Wang, J. Chen, J. Zhu, and B. Guo. Topicpanorama: A full picture of relevant topics. In *VAST '14*, pages 183–192, Oct 2014.

[33] M. Mampaey, N. Tatti, and J. Vreeken. Tell me what I need to know: Succinctly summarizing data with itemsets. In *proceeding of KDD '11*, pages 573–581, 2011.

[34] M. Mampaey, J. Vreeken, and N. Tatti. Summarizing data succinctly with the most informative itemsets. *Trans. Knowl. Discov. Data*, 6:1–44, 2012.

[35] R. Nakazawa, T. Itoh, J. Sese, and A. Terada. Integrated visualization of gene network and ontology applying a hierarchical graph visualization technique. In *International Conference on Information Visualisation*, pages 81–86, July 2012.

[36] Y. Ning, S. Muthiah, R. Tandon, and N. Ramakrishnan. Uncovering news-twitter reciprocity via interaction patterns. In *ASONAM '15*, pages 1–8, 2015.

[37] N. H. Riche and T. Dwyer. Untangling euler diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 16(6): 1090–1099, Nov 2010.

[38] D. Shahaf and C. Guestrin. Connecting two (or less) dots: Discovering structure in news articles. *Trans. Knowl. Discov. Data*, 5(4):24:1–24:31, Feb 2012.

[39] C. E. Shannon. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(1):3–55, Jan. 2001.

[40] E. Spyropoulou and T. De Bie. Interesting multi-relational patterns. In *ICDM'11*, pages 675–684, 2011.

[41] E. Spyropoulou, T. De Bie, and M. Boley. Mining interesting patterns in multi-relational data with n-ary relationships. In *Discovery Science*, volume 8140 of *Lecture Notes in Computer Science*, pages 217–232. 2013.

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

[43] N. Tatti. Computational complexity of queries based on itemsets. *Information Processing Letters*, 98(5):183–187, 2006.

[44] N. Tatti and J. Vreeken. Comparing apples and oranges - measuring differences between exploratory data mining results. *Data Min. Knowl. Discov.*, 25(2):173–207, 2012.

[45] C. Vehlow, T. Reinhardt, and D. Weiskopf. Visualizing fuzzy overlapping communities in networks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2486–2495, Dec 2013.

[46] C. Vehlow, F. Beck, and D. Weiskopf. The State of the Art in Visualizing Group Structures in Graphs. In *Eurographics Conference on Visualization (EuroVis) - STARs*, 2015.

[47] C. Wang and S. Parthasarathy. Summarizing itemset patterns using probabilistic models. In *KDD'06*, pages 730–735, 2006.

[48] H. Wu, J. Vreeken, N. Tatti, and N. Ramakrishnan. Uncovering the plot: Detecting surprising coalitions of entities in multi-relational schemas. *Data Min. Knowl. Discov.*, 28(5-6):1398–1428, Sept. 2014.

[49] P. Xu, F. Du, N. Cao, C. Shi, H. Zhou, and H. Qu. Visual analysis of set relations in a graph. In *EuroVis '13*, pages 61–70, 2013.

[50] M. A. Yalcin, N. Elmqvist, and B. B. Bederson. Aggreset: Rich and scalable set exploration using visualizations of element aggregations. *IEEE Trans. Vis. Comput. Graph.*, 22:688–697, 2016.