Cumhuriyet Üniversitesi Fen Fakültesi
Fen Bilimleri Dergisi (CFD), Cilt:36, No: 4 Özel Sayı (2015)
ISSN: 1300-1949

Cumhuriyet University Faculty of Science
Science Journal (CSJ), Vol. 36, No: 4 Special Issue (2015)
ISSN: 1300-1949

An organized method for criminal networks web mining from unstructured text documents

Armita ABARGHOEI[1,*], Ebrahim MAHDIPOUR[2], Majid ASKARZADEH[1], Mohammad Javad GOLKAR[3]

[1]*Department of computer, Qeshm international Branch, Islamic Azad University, qeshm, Iran.*

[2]*Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.*

[3]*Young Researchers and Elite Club, Shiraz Branch, Islamic Azad University, Shiraz, Iran.*

Received: 22.03.2015; Accepted: 29.05.2015

Abstract. Digital data collected for analysis at the interrogation, often contain valuable information about social networks of the suspect. Most collected records, such as emails, chats and text documents are in form of non-organized text data. A user must manually extract useful information from the text, and gather key parts in an organized database for further investigations of the criminal networks using analysis tools. It is Obvious that this process of data mining is boring along with a possibility of errors. In addition, the quality of the analysis depends on the experience and skills of the human user. This paper presents an organized method for automatic discovery of criminal networks using a set of text documents obtained from suspect devices through web mining techniques. Thus useful information about the suspected criminal network will be extracted. Our proposed method discovers direct and indirect relationships between members of criminal groups.

Keywords: Digital data, web mining, criminal networks, text documents

INTRODUCTION

The rapid growth of the web over the past decade made it the world's largest available source of information. Web has unique features; extracting information from it is along with a lot and of course fun challenges. Law enforcement agencies always need to obtain useful and hidden information data relating to crimes. A major challenge for them is the analysis of large volumes of information about the crime, which involves a large number of criminals' information in form of tables, in which Data Mining is the perfect solution for this problem. Data mining defines as a process of discovery, extraction and analysis of significant structures including large amount of information.

In many criminal cases, computer devices owned by the suspect, such as desktops, notebooks, and smart phones, are target objects for forensic seizure. These devices may not only contain important evidences relevant to the case under investigation, but they may also have important information about the social networks of the suspect, by which other criminals may be identified. In the United States, the FBI Regional Computer Forensics Laboratory (RCFL) conducted over 6000 examinations on behalf of 689 law enforcement agencies across the United States in one year (RCFL, 2009). The amount of data they examined in 2009 has reached 2334 Tera Bytes (TB), which is a double of the size processed in 2007. To accommodate the increasing demand, better resources are needed to help investigators process forensically collected data.

Chen et al. (2004) demonstrated a successful application of data mining techniques to extract criminal relations from a large volume of police department's incident summaries. They use the

* Corresponding author. Armita ABARGHOEI

Special Issue: Technological Advances of Engineering Sciences

co-occurrence frequency to determine the weight of relationships between pairs of criminals .A criminal network follows a social network paradigm.. Thus, the approach used to analyze social networks can also be used for criminal networks. Many studies presented of various approaches to build a social network through text documents. Vel et al. (2001), propose a framework to extract social networks from text document that are available on the web. Wang et al. (2007) proposed a method to rank companies based on the social networks extracted from web pages. These approaches rely mainly on web mining techniques to search for the actors in the social networks from web documents. Another direction of social network studies targets some specific type of text documents such as e-mails. Hauck et al. (2002) proposed a probabilistic approach that not only identifies communities in email messages but also extracts the relationship information using semantics to label the relationships. However, the method is applicable to only e-mails and the actors in the network are limited to the authors and recipients of the e-mails.

Researchers in the field of knowledge discovery have proposed methods to analyze relationships between terms in text documents in a forensic context. Diligenti et al (2000) introduced a concept association graph-based approach to search for the best evidence trail across a set of documents that connects two given topics. Jain (1999) employed the open discovery approach to search for keywords provided by the user and return documents containing other different but related topics. They further apply clustering techniques to rank the results and present the user with clusters of new information that are conceptually related to their initial query terms. Their open discovery approach searches for novel links between concepts from the web with the goal of improving the results of web queries. In contrast, this paper focuses on extracting information for investigation from text files.

The problem of criminal community discovery

The problem of criminal community discovery is to identify the hidden communities from a collection of text documents obtained from one (or multiple) suspect's file systems. In this paper, a text document is generally defined to be a logical unit of textual data, such as an e-mail message, a chat session, a webpage, a blog session, and a text file. Let D be a set of input text documents. Let U be the set of distinct personal names identified in D. Each document doc $\in$ D is represented as a set of names such that doc$\subseteq$ U. Let C$\subseteq$ U be a set of personal names called a community. A document doc contains a community C if C$\subseteq$ doc A community having k personal names is a k community. The support of a community C is the number of documents in D containing C. For example, {Alan, Kim} in Table 1 is a 2-community with support = 1. A community C is a prominent community in a set of documents D if the support of C is greater than or equal to a user-specified minimum support threshold. Suppose the threshold is set to 2. Then, {Alan, Kim} is not a prominent community in Table 1, but {Jenny, John, Mike} is a prominent 3-community with support = 2.

Table 1: Document set (D)

Document	Names in doc_i
doc_1	{Alan, John, Kim}
doc_2	{Jenny, John, Mike}
doc_3	{Alan, Jenny, John, Mike}
doc_4	{Jenny, Mike}

Definition 1 (Prominent community): Let D be a set of text documents. Let support(C) be the number of documents in D that contain C, where C$\subseteq$U A community C is a prominent community in D if Support (C) $\geq$ min sup, where the minimum support min sup is a user-specified positive integer threshold.

Definition 2 (Problem of criminal community discovery): Let D be a set of text documents. Let min sup be a user-specified minimum support threshold. The problem of criminal community

discovery is to identify all prominent communities from D with respect to min sup, and to extract useful information from the documents of every prominent community for crime investigation.

The problem of indirect relationship hypothesis generation

A person is indirectly related to a prominent community if there exists a sequence of intermediate terms that links a person and a prominent community through a chain of documents, in which the starting document and the ending documents contain the prominent community and the personal name, respectively. The problem of indirect relationship hypothesis generation is to identify all indirect relationships. Specifically, an indirect relationship consists of a sequence of intermediate terms between a prominent community and a personal name identified in the given document set. The generated indirect relationships may reveal some hidden links that the investigator might not be aware of. Yet, they are only hypotheses; the investigator has to further verify the truthfulness and usefulness of these relationships .

Definition 3 (Indirect relationship :(Let D be a set of documents. Let U be a set of distinct names identified in D. Let C⊆ U be a prominent community and p ∈ (U - C) be a person name that is not in C. Let D(.) ⊆ D denote the set of documents containing the enclosed argument where the enclosed argument can be a community, a personal name, or a text term. Let D(C) and D(p) be the sets of documents in D that contain C and p, respectively. An indirect relationship of depth d between C and p is defined by a sequence of terms [t1,…, td] such that

1. $D(C) \cap D(p) = \emptyset$
2. $(t_1 \in D(C)) \wedge (t_d \in D(p))$
3. $(t_r \in D(t_{r-1})) \wedge (t_r \in D(t_{r+1}))$ for $1 < r < d$
4. $D(t_{r-1}) \cap D(t_{r+1}) = \emptyset$ for $1 < r < d$ ■

Condition (1) requires that a prominent community C and a personal name p do not co-occur in any document. Condition (2) states that the first term t1 must occur in at least one document containing C and the last term td must occur in at least one document containing p. Condition (3) requires that the intermediate terms tr must co-occur with the previous term tr-1 in at least one document, and tr must co-occur with the next term tr+1in at least one document. This requirement defines the chain of documents linking C and p. Condition (4) requires that the previous term tr-1 and the next term tr+1 do not co-occur in any document. The problem of indirect relationship hypothesis generation is formally defined as follows:

Definition 4 (Problem of indirect relationship hypothesis generation): Let D be a set of text documents. Let U be the set of distinct personal names identified in D. Let G be the set of prominent communities discovered in D according to Definition 3-2. The problem of indirect relationship hypothesis generation is to identify all indirect relationships of maximum depth max_depth between any prominent community C∈ G and any personal name p∈ U in D, where max_depth is a user-specified positive integer threshold.

The proposed method

Fig. 1 depicts an overview of our proposed Criminal Community Mining System (CCMS). The first step is to read the investigated text documents and extract the personal names from them. The name extraction task is followed by a normalization process to eliminate duplicate names that refer to the same person. The next step is to discover the prominent criminal communities from the extracted names. Then, we extract the profile information that is valuable to investigators, such as contact information and summary topics, from the documents that contribute to the prominent communities. Next, we search for indirect relationships between the criminals across the document set. Finally, we provide a visual representation of the prominent communities, their related information, and the indirect relationships found in the document set. Below, we elaborate the steps of prominent community discovery, community information extraction, and indirect relationship generation.

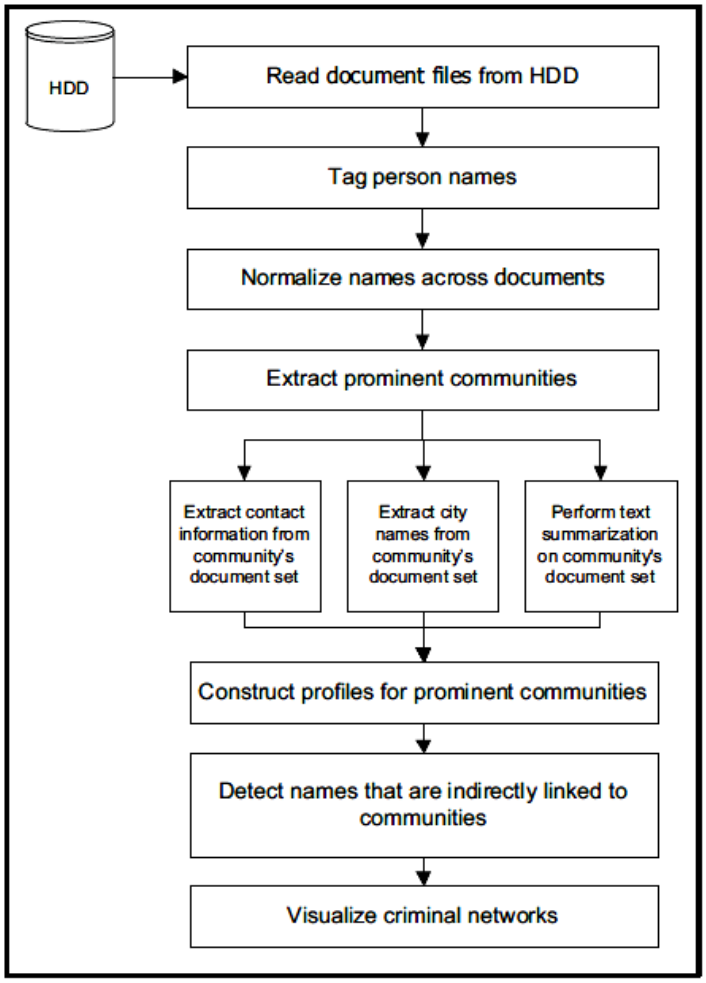

Figure 1: Criminal Community Mining System

Identifying prominent communities

The first step is to identify the personal names from the input document files. There are many Named Entity Recognition (NER) tools and methods available in the market to extract personal names. In our system, we adopt the Stanford Named Entity Tagger (Finkel et al., 2005), which is a promising tool for identifying English names. For each document doc ∈ D, we apply the NER tagger to obtain a set of personal names in doc. Variant names of the same person are merged into one name. For instance, John, J. Smith, and John Smith are transformed into a common form John Smith. Our method also allows the user to incorporate his/her domain knowledge to merge the names. For instance, some people use nicknames in a chat log and their real name is mentioned in the same session. Other NER tools can be employed if the document files contain non- English names; however, NER is not the focus of this paper.

The next step is to identify the prominent criminal communities. When two or more persons interact frequently or their names appear together frequently, this indicates a strong direct linkage. Analyzing the strength of linkages is a key step for effective crime investigation. Them strength of a linkage can be measured by comparing the frequency of the interaction between the individuals to a fixed threshold. A linkage is strong if the number of interactions passes a given threshold; otherwise, the linkage is weak or there is no linkage. A community is considered to be a prominent community if its support is equal to or greater than a given threshold.

A naive approach to identify all prominent communities is to enumerate all possible communities and identify the prominent ones by counting the support of each community in D. Yet, in case the number of identified personal names |U| is large, it is infeasible to enumerate all possible communities because there are 2^|U| possible combinations. To efficiently extract all prominent communities from the set of identified individuals, we use inductive algorithm which is originally designed to extract frequent patterns from transaction data.

Recall that U denotes the universe of all personal names in D, and each document doc ∈ D is represented as a set of names such that doc ⊆ U. Our proposed algorithm, Prominent Community Discovery (PCD), is a level-wise iterative search algorithm that uses the prominent k-communities to explore the prominent (k + 1)-communities. The generation of prominent (k + 1)-communities from prominent k communities is based on the following PCD property.

Property 1 (PCD property). All nonempty subsets of a prominent community must also be prominent because support (C′) ≥ support(C) if C′⊆ C. By definition, a community C′ is not prominent if support (C′) < min_sup. The above property implies that adding a personal name p to a non-prominent community C′ will never make it prominent. Thus, if a k-community C′ is not prominent, then there is no need to generate (k + 1)- community C′ U {p} because C′U{p} must not be prominent. The strength of the linkages among the members in a prominent community C is indicated by support(C). The presented algorithm can identify all prominent communities by efficiently pruning the communities that cannot be prominent based on the PCD property.

```
Input: A set of text documents D.
Input: User-specified minimum support min sup.
Output: Prominent communities G = {G1; : : : ;Gk}.
Output: support(Cj) and D(Cj) for every Cj □ G.
1: G1 = all prominent 1-communities in D;
2: for (k = 2; Gk-1 /= Ø; k++) do
3:   Candidatesk = Gk-1 on Gk-1;
4:   for all community C ∈ Candidatesk do
5:     if ∃C′ ⊂ C such that C′ =∈ Gk-1 then
6:       Candidatesk = Candidatesk - C;
7:     end if
8:   end for
9:   support(Cj) = 0 and D(Cj) = Ø for every Cj ∈ Candidatesk;
10:  for all document doci ∈ D do
11:    for all Cj ∈ Candidatesk do
12:      if Cj ⊆ doci then
13:        support(Cj) = support(Cj) + 1;
14:        D(Cj) ← doci;
15:      end if
16:    end for
17:  end for
18:  Gk = {Cj ∈ Candidatesk | support(Cj) ≥ min sup};
19: end for
20: return G = {G1; : : : ;Gk} with support(Cj) and D(Cj);
```

Algorithm 1 summarizes our Prominent Community Discovery algorithm. The algorithm finds the prominent communities from the prominent (k - 1) communities based on the PCD property. The first step is to find the set of prominent 1-communities, denoted by G1. This is achieved by scanning the document set once and counting the support count for each 1-community Cj. G1 contains all prominent 1-communities Cj with support (Cj) ≥ min sup. The set of prominent 1-communities is then used to identify the set of candidate 2-communities, denoted by Candidates2. Then the algorithm scans the database once to count the support of each candidate Cj in Candidates2. All candidates Cj that satisfy support (Cj) ≥ min sup are prominent 2-communities, denoted by G2. The algorithm repeats the process of generating Gk from Gk-1 and stops if Candidate is empty.

Personal names in a community are sorted by lexicographical order. Two prominent (k-1) communities can be joined together to form a candidate k-community only if their first (k-2) personal names are identical and their last (k-1) personal names are different. This operation is based on the PCD property: A community cannot be prominent if any of its subsets is not prominent. Thus, the only potential prominent communities of size k are those that are formulated by joining prominent (k-1) communities. Lines 4–8 describe the procedure of removing candidates that contain at least one nonpermanent (k-1) community.

Lines 9–17 describe the procedure of scanning the database and obtaining the support count of each community Cj in Candidates. Each candidate community Cj is looked up in each document doci in the document set. Once a match is found, the value of support (Cj) is incremented by 1 and the document doci is added to the set D (Cj). If support (Cj) is greater than or equal to the user specified minimum threshold min_sup, then Cj is added to Gk, the set of prominent k-communities with k members. The algorithm terminates when no more candidates can be generated or when none of the candidate communities pass the min_sup threshold. The algorithm returns all prominent communities G = {G1, ., Gk} with support counts and sets of associated documents for each prominent community.

Example 1 (Prominent communities discovery). Consider Table 1 with min_sup = 2. First, we scan the table to find G1 = {Alan, Jenny, John, Mike}. Next, we perform G1) G1 to generate Candidates2 = {{Alan, Jenny}, {Alan, John}, {Alan, Mike}, {Jenny, John}, {Jenny, Mike}, {John, Mike}}. Then we scan the table once to obtain the support of every community in Candidates2 with support ≥ 2, and identify G2 = {{Alan, John}, {Jenny, John}, {Jenny, Mike}, {John, Mike}}. Similarly, we perform G2 on G2 to generate Candidates3 = {{Jenny, John, Mike}} and scan the table once to identify the prominent 3-community G3 = {{Jenny, John, Mike}}.

Extracting information of prominent communities

The next phase is to retrieve useful information for crime investigation, such as contact information, from the discovered prominent communities. In the context of this paper, a group of people are considered to be in the same prominent community if their names appear together frequently in a minimum number of text documents.

Thus, the topics of the set of documents containing their names are the "reasons" bringing them together. By analyzing the content of the text documents containing the names of the community members, a crime investigator may obtain valuable clues that are useful for further investigation, especially during the early stages of the investigation. For instance, if a set of community member names are all contained in the same chat sessions, then summarizing the topics of the discussion can help the investigator infer the type of relationship the community members share. To facilitate the crime investigation process, we extract the following types of information from the set of documents D (Cj) for each prominent community Cj:

1. Key topics
2. Names of other people who are not in Cj
3. Locations and addresses
4. Phone numbers
5. E-mail addresses
6. Website URLs.

In some real-life cyber-criminal cases, there could be thousands of identified individuals and hundreds of prominent communities. Even with a data mining software, an investigator may still find it difficult to cope with such a large volume of information. The summarized key topics from D (Cj) can provide the investigator with an overview of each community and the related topics. The extracted key topics can be a link label when the communities are visualized on the screen. Some people names may appear only a few times in D(Cj) but may not be

frequent enough to be included as a member in Cj. Identifying these infrequent people names may lead to some new clues for investigation. Locations, addresses, and contact information, such as phone numbers and e-mail addresses, are valuable information for crime investigation because they may reveal other potential channels of communications among members of the criminal community.

To extract the key topics, we employ an Open Text Summarizer (OTS). To extract the city names, we search the documents for the cities in the GeoWorldMap database. To extract other addresses, phone number, and e-mail addresses, we use regular expressions.

Other useful information may be extracted to further describe the relationship between the members of an identified prominent community, such as the duration of the relationship which is a key piece of information regarding the activity of members of a community. It is especially useful to provide the investigator with a sense of a time line for the relationships that the communities share. To specify the duration of the relationship between a criminal communities identified in a set of text documents, we make use of the metadata of the documents. The metadata of a file is the data linked to this file by the hosting system upon creation of the document. We can define the duration of a relationship as all or some of the values of: (1) the starting date of the relationship, (2) the ending date, (3) and the amount of time the relationship lasted. We can identify the starting date of the relationship between members of a prominent community Cj, by the oldest of the dates attached to the documents in D (Cj). The end date of the relationship is the most recent of the dates associated with the documents. The duration of the relationship is then calculated as the difference between the start and end dates.

Discovering indirect relationships

In this section, we present a method to discover the evidential trails between a prominent community identified in a dataset and other people in the document set who are not in the community. An evidential trail represents a relationship between the prominent community and other people through a common topic rather than co-occurrence. This trail is extracted as a chain of intermediate terms that link a community to a person. Thus, for a given prominent community Cj and a personal name p, the indirect relationship discovery method identifies a chain of intermediate terms t from the dataset that links Cj with p. The length of the chain is limited by a user-specified threshold, denoted by max_depth.

Profiles

Any term t in a document set D can be profiled by extracting interesting information about it from the textual content of documents in D. For example, if the document set is obtained from newswire documents, the profile of a topic such as Microsoft can be: Corporation, Windows, Bill Gates, and Office. In the same sense, the profile for a prominent community C existing in a hard drive can be city, phone, and email. This information can be retrieved from the documents in which the prominent community occurs. However, this information should not be chosen randomly because of the importance of the profile information in the hypothesis discovery process. If the profile information is too general, the discovered relationship is unlikely to be significant and the investigator may be overwhelmed by a large number of false hypotheses. Thus, data must only be added to the profile if it satisfies some pre-specified constraints or conditions that are set to ensure the usefulness of this data.

The structure of a profile is based on semantic types. This structure ensures that only a specific type of information is added to the profile. In the criminal network analysis context, we select semantic types that are significant to investigations. In particular, the following semantic types are selected: (1) summary topics of the documents representing the prominent community's interactions, (2) other names of people mentioned in the documents with the prominent communities, (3) cities and locations, (4) email addresses, (5) phone numbers, and (6) website URLs. These semantic types are also used to identify the relationship between the members of

prominent communities. For the profiles of the prominent communities, we use the same information that is retrieved for the prominent communities, as explained earlier, for several reasons. First, it is less costly in the information extraction process. Second, these semantic types are extracted as forensically valuable information about the set of related individuals and consequently any other relationships that are found using this information are likely to be valuable as well. Within each semantic type in the profile of a prominent community, each term has a weight associated with it. In order to minimize the computations, we define the profiles for the prominent communities of maximal size, and combine all the profiles of the sub-communities.

Definition 5 (Profile): A profile for a prominent community C, denoted by P(C), is defined by a collection of vectors Vx1, Vx2,..., Vxn , where n denotes the number of semantic types considered. Each vector Vxi, of length lxi, where lxi is the number of terms of semantic type xi, is given by:

$$V_{x_i} = \begin{bmatrix} t_1, f_{x_i}(t_1) \\ t_2, f_{x_i}(t_2) \\ \vdots \\ t_{l_{x_i}}, f_{x_i}(t_{l_{x_i}}) \end{bmatrix}$$

Where fxi(tj) is the weight of term tj and is given by

$$f_{x_i}(t_j) = \frac{f'_{x_i}(t_j)}{\max_j f'_{x_i}(t_j)}$$

And

$$f'_{x_i}(t_j) = n_{x_i,t_j} \times log(|D|/n_{t_j})$$

Where |D| is the total number of documents, ntj is the frequency of occurrence for term tj in the document set D, and nxi;tj is the frequency of term tj of semantic type xi in D(C).

Indirect relationship generation algorithm

Given a prominent community C with profile P(C), we propose the indirect relationships generation algorithm to extract indirect relationships between the prominent community C and other individuals in the document set. This algorithm is a hybrid version of both the open and closed discovery algorithms. The closed discovery requires two terms, A and C, and generates hypothetical relationships between A and C through intermediate terms B. On the other hand, open discovery requires the entry of only one initial term; the B and C are provided by the algorithm. We propose a model that is open. However, the other end of the relationship must be the name of an individual from the document set.

Algorithm 2 shows the steps of the indirect relationship generation algorithm. The method is applied for each prominent community Cj ∈ G. The algorithm requires the profile of the prominent community P(C) as input, where at least one of its term vectors Vx is of the semantic type persons. Both the number of intermediate terms N and the depth of the indirect relationship max_depth are set by the user. If the depth is 1, for example, then the indirect relationship between community and person c is through one connecting term, e.g., a→b→c. However, if the depth is 2, then the relationship is of the form a →b → e →c.

Input: Profile $P(C) = \{V_{x_1}, \ldots, V_{x_n}\}$ for community C, number of intermediate terms N, maximum depth threshold max_depth
Output: Set of personal names indirectly related to C through intermediate terms

$P = P(C)$;
$dep = max\_depth$;
while $dep > 0$ **do**
  Let B_{x_i} denote the N top ranking terms in V_{x_i} corresponding to P;
  Construct profile $P(B_x[y])$, $x \in X$, $y = 1, \ldots, N$;
  Combine profiles $P(B_x[y])$ into one profile $P(O)$, where the weight of a term in $P(O)$ is the sum of its weights in profiles $P(B_x[y])$, $y = 1, \ldots, N$;
  for each $t_j \in V_x$ of $P(O)$, $x \in X$ **do**
    Conduct a query (t_j AND C) in D;
    if result $\neq \emptyset$ **then**
      remove t_j from $P(O)$;
    end if
  end for
  $dep = dep - 1$;
  $P = P(O)$;
end while
return terms O_x ranked by weight, for semantic type x=*persons*;

Algorithm 2: Indirect Relationship Generation Algorithm

The algorithm proceeds with the truncation of the term vectors, Vx, comprising the profile P(C), by selecting the top N ranking terms in each Vx for all values of x ∈ X. The new truncated vectors are called Bx for each semantic type x accordingly. Next, for each xi ∈ X, we search the document set for each term Bx[y], y = 1,…, N in order to build its profile. The constructed profiles are P(Bx[1]), P(Bx[2]), ., P(Bx[N]). Now, a combined profile is computed where the combined weight of a term is the sum of its weights in each of P(Bx[1]), P(Bx[2]), ., P(Bx[N]) in which it occurs. This combined profile is called P(O) and is comprised of vectors Vx for each semantic type x ∈ X. For each term in tj in the profile P(O), if a search for (C AND tj) returns a nonempty set, the term tj is removed from P(O). If the depth is set to a value greater than 1, the algorithm iterates again using the value of the profile P(O) produced from the previous iteration as the input profile for the next iteration. Finally, the method returns the terms in P(O) for the semantic type persons ranked by combined weight and terminates.

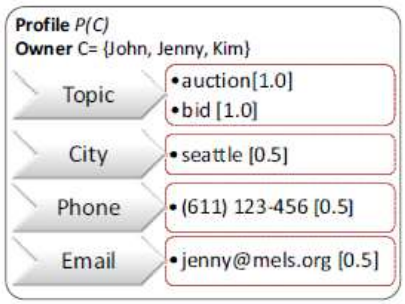

Figure 2: Example of a profile

Example 2 (Indirect relationship hypothesis generation).

To illustrate the steps of the algorithm, consider the community C with profile P(C) in Fig. 2 with depth = 1. First, start with the profile P(C) for the community C = {John, Jenny, Kim} and construct profiles for each term in P(C). The profiles for the terms auction and seattle are denoted by P(auction) and P(seattle), respectively, with values as shown in Fig. 3. Next, the

profiles are combined into one profile P(O) as shown in Fig. 3. For each name tj in the persons vector, search for documents containing both tj and C. In this example, the first lookup searches for documents containing all four names: John, Jenny, Kim, and Sam. If no document contains all of them, then it implies that Sam has no co-occurrence relationship with the prominent community. Thus, Sam is indirectly linked to C through the term auction and Bob is linked to C through the term settle. Fig. 3 shows the final results of the discovery method when applied to this example.

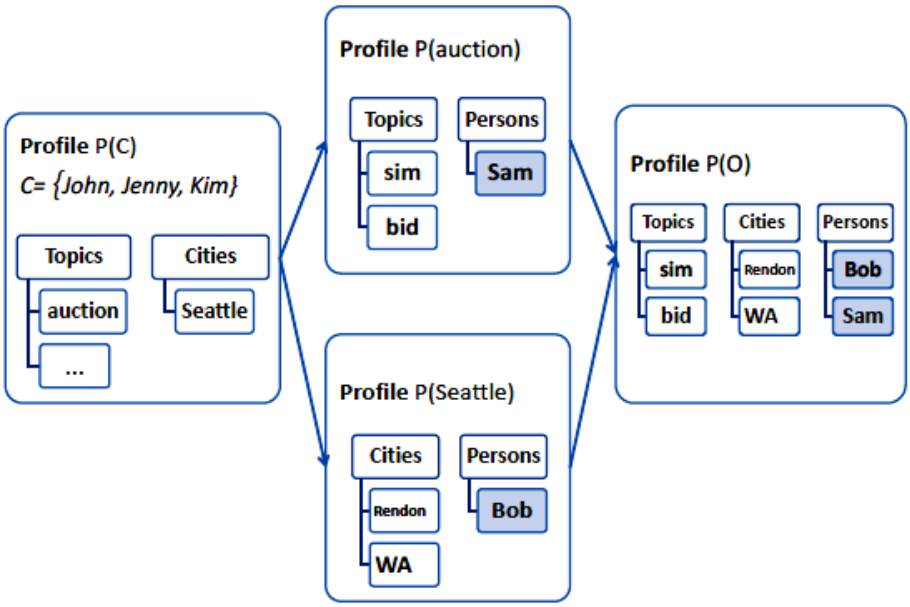

Figure 3: Indirect relationship hypothesis generation

Case study on real-life cybercrime

The objective of this case study is to demonstrate the effectiveness of our proposed notions and methods in a real-life cybercrime investigation. The dataset was provided by a Canadian law enforcement unit. In particular, we performed experiments on an MSN chat log from a hard disk that was confiscated from a suspect in a computer hacking case. The chat log, with a size of approximately 500 MB, contains the chat messages and file attachments from 220 distinct chat accounts. The case had already been solved by the investigator of the law enforcement unit. To judge the effectiveness of our proposed method, we compared the investigation result using our implemented prototype with the result manually obtained by the investigator. We were informed that the nature of the crime was related to computer hacking. No other information was provided to us regarding the chat log to be analyzed. This scenario is similar to the early stage of an investigation when an investigator has limited prior knowledge about the suspect(s). Due to confidentiality and privacy concerns, some of the information had to be masked and all the identities, e-mail accounts, and server names had to be replaced by pseudonyms in the following discussion .

Fig. 4 depicts the prominent communities identified from the MSN chat log with min_sup = 5. Each node in the figure represents a distinct chat account. The distances among the nodes in a community C represent the closeness of its members, which are computed from the inverse of support(C). Specifically, there are 24 distinct chat accounts, forming 32 prominent 2-communities, 4 prominent 3- communities, and 1 prominent 4-community. The interactions of the other 196 accounts are not frequent enough to form a prominent community. The two central nodes, denoted by S1 and S2, represent two chat log accounts owned by the same

suspect, the owner of the confiscated computer; therefore, the other 22 users communicate with at least one of S1 and S.

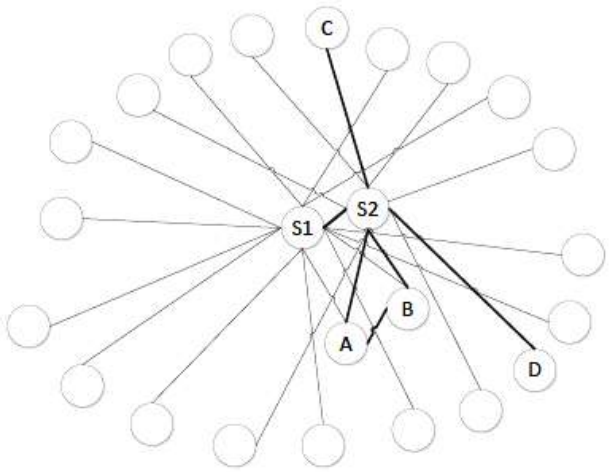

Figure 4: Prominent communities in the case study

Our proposed framework extracts some information, namely key topics, person names, locations, addresses, e-mails, and URLs, from the documents (the chat messages) of each prominent community. By performing a simple search on the key topics of each community, we identified a suspicious user, denoted by C, who discussed about "botnet" with S2. This led us to further look into the details of the extracted community information of this particular prominent 2- community. We found that S2 and C had exchanged several e-mail addresses in a form similar to this anonymous form "aaa999@123.456.789.101.dsl.isp.ca". The subdomain and domain of the e-mail address indicate that it is a temporary IP address assigned by a DHCP server. Yet, it is unusual to have an e-mail server running on a dynamically changing IP address. Therefore, we alleged that the servers were not real e-mail servers, but some bots controlled by the suspect. Furthermore, A and C exchanged several suspicious URLs, in a form similar to "http:// <user>. <free hosting company>.com/save.exe", which point to some binary executable files. We concluded that these executables were probably used for spreading the malware to victims. The indirect relationships discovery process also illustrates that C is indirectly related to the prominent 4- community {S1, S2, A, B} through the shared URLs .

Among the 37 prominent communities, we identified 6 prominent 2-communities and 1 prominent 3-community that share suspicious information similar to the aforementioned e-mail addresses and URLs. These suspicious communications are indicated by the dark lines in Fig. 4. We confirmed the correctness of these identified criminal communities and activities with two cybercrime investigators in the lawenforcement unit who solved the real case by manually reading all the chat messages. Thus, the precision of our method in this analysis is 100%. Yet, our proposed method missed 1 suspicious community that was identified by the investigators. As a result, the recall of our method in this analysis is 7/8 - 88%. Our method failed to identify such community due to its infrequent communication. There are two ways to further improve the recall. The first obvious solution is to lower the min_sup threshold at the expense of larger number of prominent communities. The second solution is to identify the suspicious information, e.g., e-mail addresses and URLs, from the prominent communities, and then search for such information in the rest of the infrequent communities.

Performance analysis

The performance analysis is performed on two real-life datasets. The first dataset is the Enron e-mail corpus (Mark and Perrault). We used Enron to analyze the effectiveness of the prominent communities discovery algorithm. Although Enron is a de facto benchmark dataset used in the

field of e-mail forensics, the expected input of our proposed method should be a large collection of files obtained from a file system, not only emails. Therefore, to further evaluate the performance, especially the scalability, of our proposed method; we used the hard drive of the first author's personal computer as the second dataset. Throughout the rest of the section, we refer to this dataset as Filesystem.We present the analysis results of Enron and Filesystem in Sections 6.1 and 6.2, respectively.

The Enron dataset

The Enron dataset contains the e-mails of 158 employees in Enron Corporation, which was an American energy, commodities, and services company before its bankruptcy. In this experiment, we created two versions of data from the Enron dataset, namely EnronSmall and EnronFull.

EnronSmall contains the e-mails from 30 randomly selected employees, resulting in 48,618 e-mails and 5481 distinct person names. Our proposed method required 16 min to complete the entire process, in which 12 min are spent on extracting all prominent communities for min_sup = 8, 4 min are spent on identifying all indirect relationships, and 3 s are spent on displaying the result. Fig. 5 depicts only a subset of prominent communities identified in EnronSmall. Among the 14 prominent communities in the figure, 5 of them are prominent 2- communities and the remaining 9 are prominent 3- communities. When a user clicks on a community, the relevant information, namely other person names, cities, emails, phones numbers, and discussed topics, of the community is shown at the bottom of the screen. We inspected the e-mails manually and compared the resulting contact information with the actual content of the messages. The system correctly identified all e-mails and phone numbers without false positives in this case. Fig. 6 depicts a subset of indirect relationships identified in EnronSmall. In this particular example, when the mouse was hovered on the link between john and {bush, gray, davis, kevin scott}, the email jeff.skilling@enron.com popped up, indicating that john was related to the prominent community {bush, gray, davis, kevin scott} through the email jeff.skilling@enron.com.

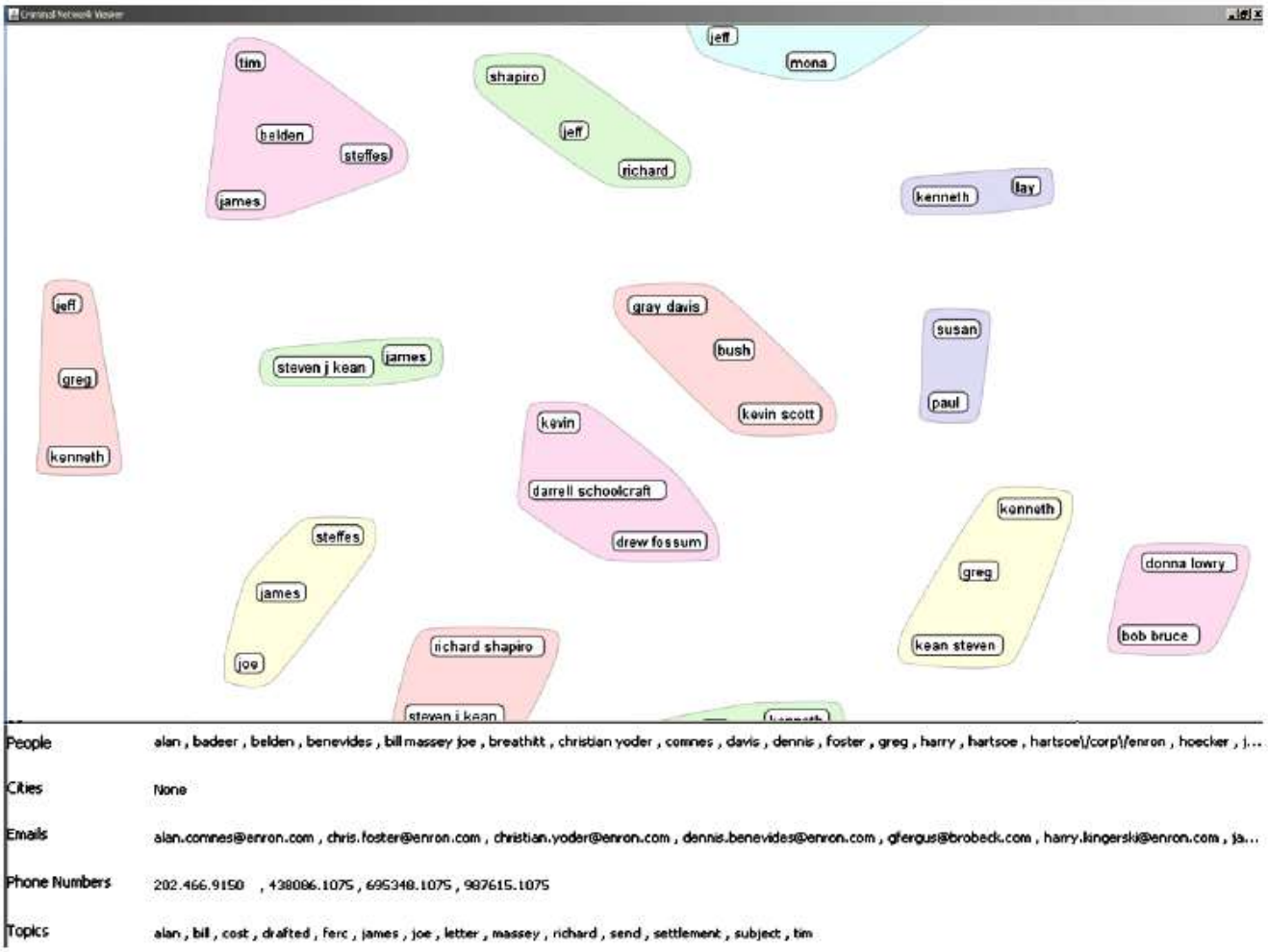

Figure 5: Prominent communities in *EnronSmall*

EnronFull contains the e-mails from all 158 employees with 2.53 GB of 515,767 e-mails and 108,835 distinct person names. Fig. 7 depicts the number of prominent communities with respect to the number of members (k) in a community for min_sup ¼ 250. The number of prominent communities peaks at k = 7. As k increases, a community becomes more difficult to satisfy the minimum support threshold; therefore, the number of prominent communities drops. Our method required 193 min to extract all prominent communities and around 3 s to display the results.

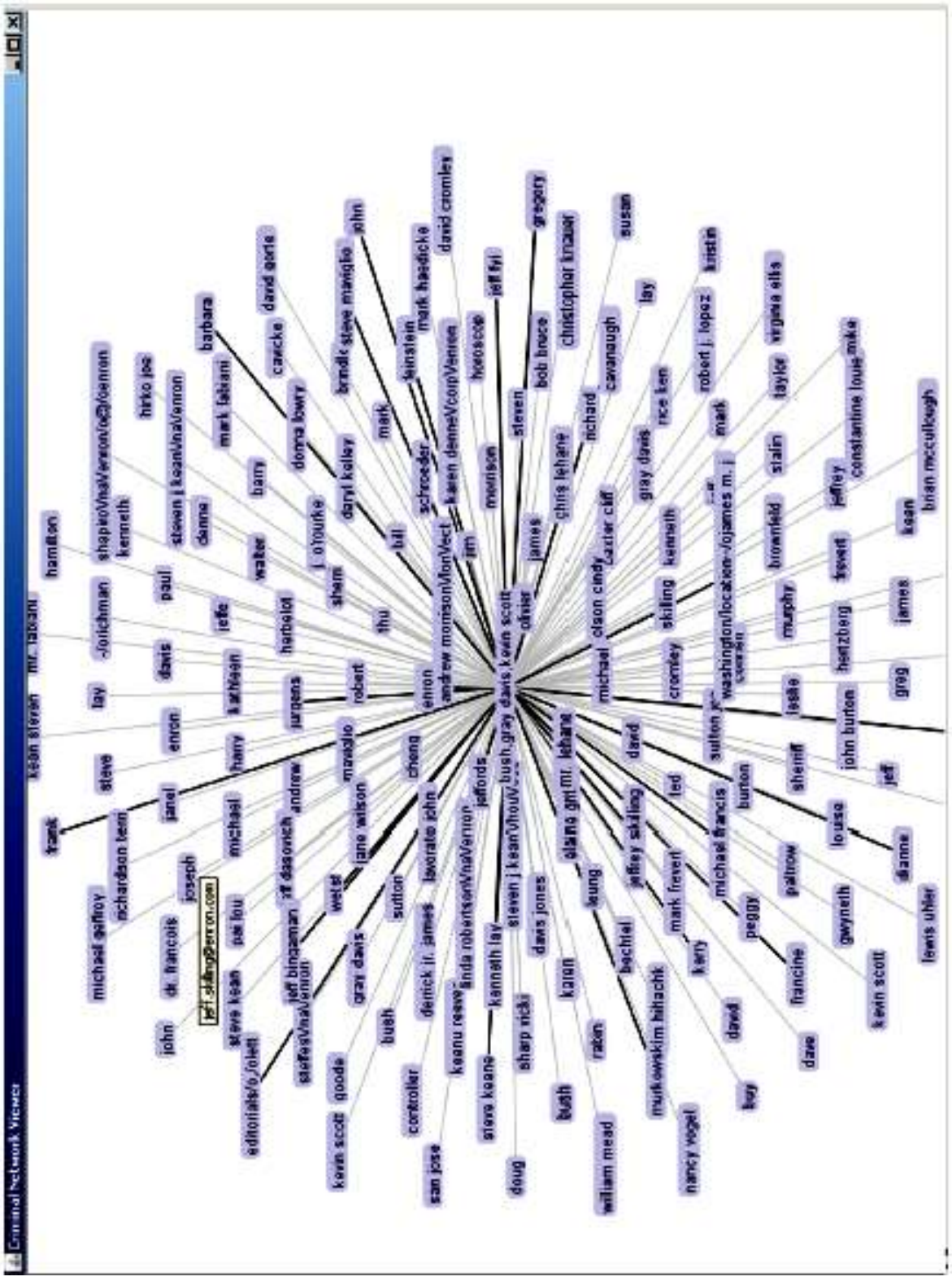

Figure 6: Indirect relationships in *EnronSmall*

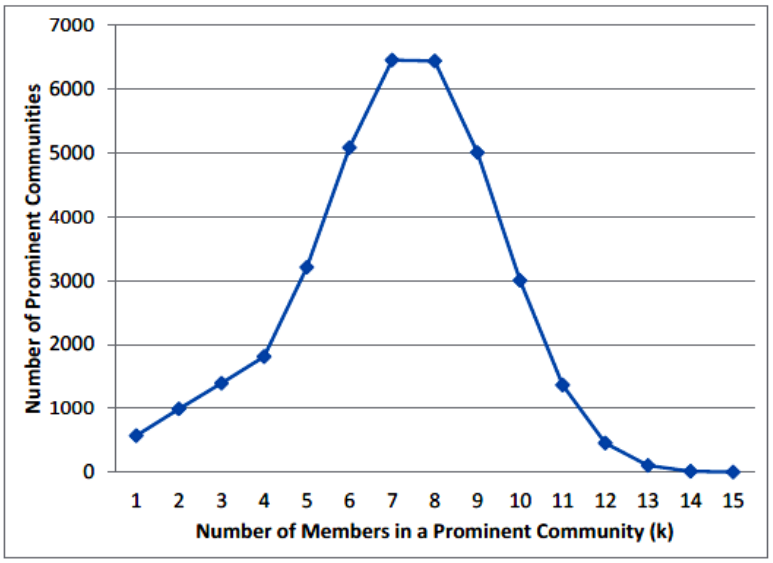

Figure 7: Number of Prominent Communities vs. k in *EnronFull*

The File system dataset

Fig. 8 shows the number of prominent communities for $4 \leq \text{min sup} \leq 12$. As the minimum support threshold increases, the number of prominent communities quickly decreases because the number of documents containing all members in a community decreases very quickly.

File Type	Number of Files	Size in MB	Percentage
All	43562	40000	100
html	14045	420	1.05
pdf	326	200	0.5
txt	434	1.3	0.00325
xml	995	60	0.15
audio	215	1058	2.645
video	105	840	2.1
MS office	253	66	0.165
other	27022	37354.7	93.38

Table 2: Description of *Filesystem* (40GB)

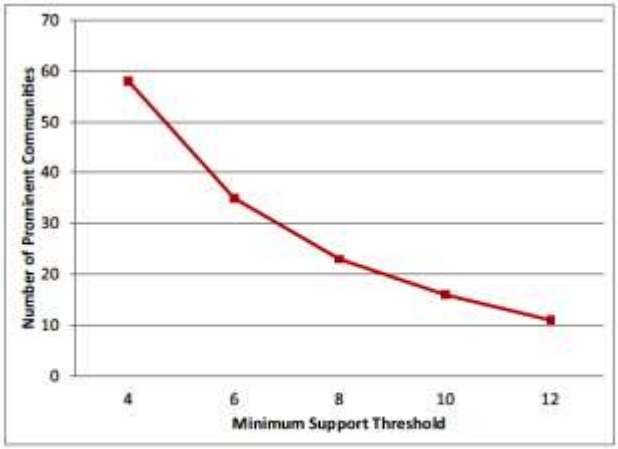

Figure 8: Number of prominent communities vs. minimum support threshold

Next, we evaluate the scalability of our proposed methods by measuring its runtime. The evaluation is conducted on a PC with Intel 3 GHz Core2 Duo with 3 GB of RAM. Fig. 9 shows the runtime of our proposed methods with respect to the size of the document set which varies from 10 GB to 40 GB with min_sup = 8. The program takes 1430 s to complete the entire process for 40 GB of data, excluding the time spent on reading the document files from the hard drive. As shown in the figure, the total runtime is dominated by prominent community discovery procedure. The runtime of the indirect relationship generation and visualization procedures is negligible with respect to the total runtime.

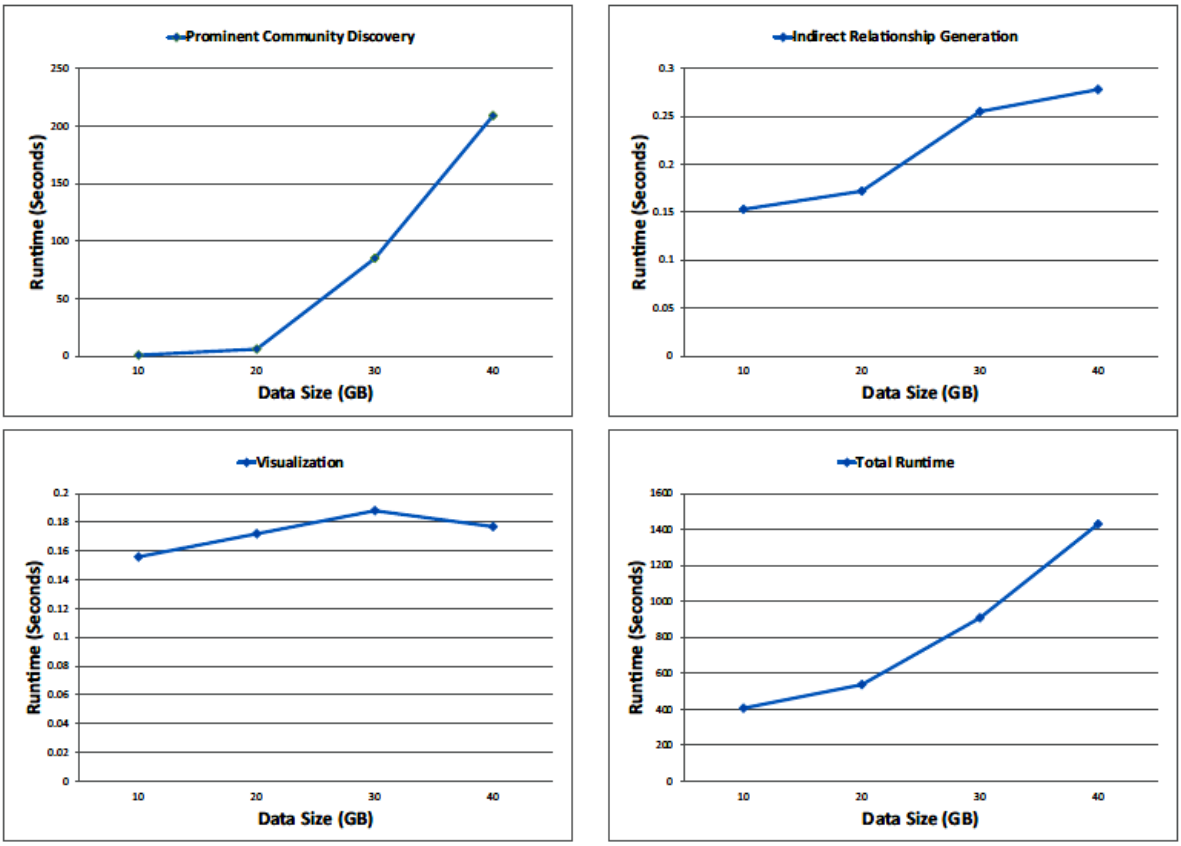

Figure 9: Scalability: Runtime vs. data size

CONCLUSION

We have proposed an approach to discover and analyze criminal networks in a collection of investigated text documents. Previous studies on criminal network analysis mainly focus on analyzing links between criminals in structured police data. As a result of extensive discussions with a digital forensics team of a law enforcement unit in Canada, we have introduced the notion of prominent criminal communities and an efficient data mining method to bridge the gap of extracting criminal networks information and unstructured textual data. Furthermore, our proposed methods can discover both direct and indirect relationships among the members in a criminal community. The developed software tool has been evaluated by an experienced crime investigator and has received positive feedback.

REFERENCES

[1] H. Chen, W. Chung, J. J. Xu, G. Wang, Y. Qin, and M. Chau, "Crime data mining: a general framework and some examples," Computer, vol. 37, pp. 50-56, 2007.

[2] O. de Vel et al., “Mining E-Mail Content for Author Identification Forensics,” SIGMOD Record, vol. 30, no. 4, 2001, pp. 55-64.

[3] G. Wang, H. Chen, and H. Atabakhsh, “Automatically Detecting Deceptive Criminal Identities,” Comm. ACM, Mar. 2007, pp. 70-76.

[4] R.V. Hauck et al., "Using Coplink to Analyze Criminal-Justice Data," Computer, Mar. 2002, pp. 30-37.
[5] M. Diligenti, F. Coetzee, S. Lawrence, C. L. Giles, and M. Gori. Focused Crawling Using Context Graphs. In Proc. of Intl. Conf. on Very Large Databases (VLDB'00), pp. 527–534, 2000.
[6] Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys 31(3), 264–323 (1999.(
[7] Agrawal, R., Imielinski, T., Swami, A.N: Mining association rules between sets of items in large databases. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (1993).