Type: dg.o 2002 final paper Jennifer J. Xu, Hsinchun Chen
Demo: No
Authors: Jennifer Jie Xu, Hsinchun Chen
Address: Department of Management Information Systems
        The University of Arizona
        Tucson, AZ 85721
        Tel: (520) 626-9239
        E-mail: jxu, hchen@bpa.arizona.edu

Using Shortest Path Algorithms to Identify Criminal Associations

Jennifer Jie Xu
Department of Management
Information Systems
The University of Arizona
Tucson, AZ 85721
jxu@bpa.arizona.edu

Hsinchun Chen
Department of Management
Information Systems
The University of Arizona
Tucson, AZ 85721
hchen@bpa.arizona.edu

Abstract

Frequently in criminal investigations, law enforcement agencies face the problem of identifying associations between a group of entities such as individuals and organizations. In this paper we present a link analysis technique to solve such a problem. This approach uses shortest path algorithms to find the strongest associations between two or more given entities. The experimental results have demonstrated that our approach is potentially useful in terms of quality and efficiency. Specifically, we found that the two-tree Priority-First Search algorithm in most cases was the fastest algorithm to find shortest paths and the paths found consisted of meaningful criminal associations around 80 percent of the time.

1. Introduction

Link analysis techniques have been used in a wide variety of areas. These techniques in a large part rely on structural analysis of domain-specific networks to discover valuable knowledge. In law enforcement, intelligence analysts often refer to nodes and links in a criminal network as entities and relationships (Sparrow, 1991). The entities may be criminals, organizations, vehicles, weapons, bank accounts, etc. The relationships between the entities specify how these entities are associated together.

Law enforcement agencies and intelligence analysts frequently face the problems of identifying possible relationships among a specific group of entities in a criminal network. However, such tasks can be fairly time-consuming and labor-intensive without the help of link analysis tools. Although there have been many link analysis software packages available, most of them only provide a visual representation of a criminal network. The computers are "still not doing the analysis" (Sparrow, 1991). To alleviate law enforcement agencies' intelligence analysis burden, we present in this paper a link analysis technique based on graph search algorithms. More specifically, we use the shortest path algorithms to identify the strongest associations between two or more entities in a criminal network. When investigating the potential of our algorithms, we focus on two important issues: quality and efficiency. The quality issue concerns whether the association paths found by the algorithms are meaningful and useful to users, and the efficiency issue concerns how fast an analysis task can be completed.

The remainder of the paper is organized as follows. Section 2 reviews the literatures and related work on link analysis in law enforcement. Section 3 presents our shortest path algorithms. Experiments and results are presented and discussed in section 4. Section 5 concludes the paper and suggests directions for future work.

2. Literature Review

Many link analysis studies in the law enforcement domain focus on methods of constructing criminal networks from database records or textual documents; a few studies have developed techniques for performing criminal link analysis. In this section we first review the techniques of criminal network construction and structural link analysis in law enforcement, and then introduce the shortest path algorithms.

2.1 Link Analysis in Law Enforcement

2.1.1 Network Construction

Several network construction techniques have been developed to build links between structured records in databases. For example, Goldberg and Senator (Goldberg and Senator, 1998) suggested that consolidation and link formation operations be performed on transactional database records to build a criminal network to support link analysis. The links or associations between the consolidated individuals are formed when they have related transactions. This technique has been employed by the U.S. Department of the Treasury to detect money laundering transactions and activities (Goldberg and Wong, 1998). A different network construction technique used by COPLINK Detect (Hauck et al., 2002) is based on the concept space approach developed by Chen and Lynch (1992). A concept space can be treated as a network consisting of domain-specific concepts (nodes) and their weighted co-occurrence relationships (links) (Hauck et al., 2002) or associations. In COPLINK Detect, the concepts are database elements of persons, organizations, vehicles, and locations. In such a network, a link exists between a pair of concepts if the two concepts ever appear together in the same criminal incidents.

Some other network construction techniques can build networks based on information extracted from textual documents. Lee (Lee, 1998) developed a technique to construct criminal networks of entities, events, and relationships from free texts. Relying heavily on Natural Language Processing (NLP) techniques, this approach can extract entities and events from free texts by applying large collections of patterns. Associations among the extracted entities and events are formed using relation-specifying words and phrases.

2.1.2 Link Analysis

A few existing link analysis systems employ different techniques to perform analytical tasks.

A link analysis tool called Watson (Anderson et al., 1994) searches and identifies useful links and associations between entities by querying databases. Given an entity, Watson can automatically forms a database query to search for other related records. Those related records are linked to the given entity and the result is presented in a link chart. An analyst may then examine these records and links to discover useful clues for further investigations.

Link Discovery Tool (Horn, Birdwell, and Leedy, 1997) has been developed to discover groups of individuals and organizations that are strongly related by associations in a criminal network. Specifically, it can use shortest path algorithms to discover association paths between two individuals that on the surface appear to be unrelated. However, it cannot work in the situations where associations need to be found between more than two entities.

The next section reviews the conventional shortest path algorithms. Although these techniques have been studied and employed widely in other domains, they have not received enough attention in law enforcement.

2.2 Serial Shortest Path Algorithms

As a special type of graph search algorithm, the shortest path algorithms search for optimal paths between the nodes by examining the link weights in a graph (e.g. a network).

The Dijkstra algorithm (Dijkstra, 1959) is the classical method for computing shortest paths from a single source node to all the other nodes in a weighted graph. Most other algorithms for solving the shortest path problems are based on Dijkstra algorithm but have improved data structures for implementation (Evans and Minieka, 1992). For example, the Priority-First Search (PFS) algorithm (Cormen, Leiserson, and Rivest, 1991) is faster than the Dijkstra algorithm because it employs a priority queue for implementation.

Unlike the classical Dijkstra algorithm, the two-tree Dijkstra algorithm computes the shortest path from a single source node to a single sink node, rather than to all other nodes in a graph. Previous studies have

demonstrated that the two-tree Dijkstra algorithm can be much faster than Dijkstra algorithm (Helgason, Kennington, and Stewart, 1993).

In summary, there have been a number of studies on link formation and network construction in the law enforcement domain. However, research on structural link analysis has not been sophisticated. Moreover, little research has been done to solve the problem of identifying the strongest criminal associations between two or more entities in a criminal network. In the next section, we will apply the conventional shortest path algorithms with modifications to this problem.

3. Applying Shortest Path Algorithms

As stated earlier, this paper intends to (a) propose our solution to the problem of identifying the strongest criminal associations between two or more entities, and (b) evaluate the potential of our approach in terms of quality and efficiency.

We employ the conventional shortest path algorithms to address the first research question. However, although shortest path algorithms can identify the strongest association between a single pair of source nodes, they cannot achieve the goal if there are more than two source nodes. We therefore propose to repeatedly use a shortest path algorithm to solve the multiple source node problem, since a group of nodes is strongly associated if each pair of the nodes in the group is strongly associated. That is, given k source nodes, we compute the shortest path for every possible pair of the source nodes. Thus, the total number of shortest paths is $k(k-1)/2$. It is possible that some of these paths share common links. If this happens, we combine the common links to avoid redundancy.

3.1 The Modified Dijkstra/PFS Algorithm

The original Dijkstra algorithm finds the shortest paths from a single source node to all the other nodes in a graph. It works by maintaining a shortest path tree T rooted at a source node, say s. T contains nodes whose shortest distance from s is already known. Initially, T contains only s. At each step, we select from the candidate set Q a node with the minimum distance to s and add this node to T. Once T includes all nodes in the graph, the shortest paths from the source node s to all the other nodes have been found.

With minor modifications, Dijkstra algorithm can be used to compute the shortest paths from a single source node to a set of specified nodes in the graph. That is, given a set of nodes $K \subseteq N$, $|K| = k \geq 2$, and a source node $s \in K$, the modified Dijkstra algorithm can compute the shortest paths from s to all $u \in K$, and $u \neq s$, where N is a set containing all the nodes in the graph. The simple modification is made to the algorithm so that it stops as soon as all $u \in K$ are included in the shortest path tree T.

The modified PFS differs from the modified Dijkstra only in the data structure used. PFS uses a priority queue rather than a linear linked list for implementing Q.

The modified Dijkstra or PFS algorithm is used k times to compute all possible shortest paths for k source nodes.

3.2 Two-tree Dijkstra/PFS algorithm

No modification is made to the two-tree Dijkstra algorithm because it can only find the shortest path between two nodes. The two-tree Dijkstra algorithm works by searching from both ends of a shortest path simultaneously (Helgason, Kennington, and Stewart, 1993). A shortest path tree rooted at source node s and a shortest path tree rooted at another source node t grow in alternate steps. The two trees are analogous except that the tree rooted at s expands a node by examining its outgoing links, and the tree rooted at t expands a node by examining its incoming links. A shortest path is found when both trees have a common node, say r, such that the sum of the distance from r to s and the distance from r to t is a minimum. Assuming a priority queue data structure is used for implementation, we call this algorithm two-tree PFS. The two-tree PFS algorithm is used $k(k-1)/2$ times to find the shortest paths for all possible pairs of nodes in K.

4. System Evaluation

In order to investigate the potential of the shortest path algorithms to identify associations between entities in criminal networks, we performed a user evaluation and a series of simulations. The user evaluation was aimed at addressing quality issues, namely, whether the associations identified by the algorithms are meaningful to users. The aim of the simulations, on the other hand, was to address our research question regarding system efficiency, specifically the execution speed of each algorithm.

4.1 COPLINK Concept Space

The criminal network used in our experiment was constructed based on the same concept space approach (Chen and Lynch, 1992) used in COPLINK Detect (Hauck et al., 2002). However, unlike COPLINK Detect which uses structured database elements, our concept space used the noun phrases extracted from textual documents as concepts. The automatic noun-phrasing tool called AZNP (Tolle and Chen, 2000) extracted noun phrases from texts using lexicons and stop word lists. Co-occurrence weights between the concepts (noun phrases) were calculated to generate the associations. We used unstructured textual data because law enforcement agencies often rely on report narratives to obtain information that may not otherwise be available from structured data.

4.2 The Data Set

The textual documents were crime report narratives provided by the Phoenix Police Department. The original data set contained one-year worth's of reports with total size of 1GB. In order to sample a data set having a medium size and including the complete one-year's worth of reports, we selected kidnapping reports as our test data. The size of this collection was 4.5MB. This collection contained 271 individual reports, from which 95,328 noun phrases were extracted. After irrelevant terms were filtered out based on a 3400-item stop word list, 280 concepts remained. The resulting concept space contains these 280 concepts connected by 25,862 co-occurrence associations. On average, a concept has 92.4 associations.

4.3 Results and Discussions

4.3.1 User Evaluation: Quality Issue

In this evaluation, we wanted to find out whether the association paths identified by the system were meaningful to users. An association path is meaningful if it can tell the user the relationship between entities.

The authors acted as the subjects of this user evaluation. The results from a test bed with 30 randomly generated cases were evaluated. The number of source concepts in these cases ranged from 2 to 5. Which algorithm was used is not important here because the three shortest path algorithms always produced the same results. The subjects examined the paths by reading the original police reports and determined whether the association paths were meaningful.

A precision type measure was developed to test the quality of association paths. For a specific number of source nodes, k, this measure is defined as follows:

$$p_k = \frac{number\ of\ meaningful\ paths\ selected\ by\ subjects}{total\ number\ of\ paths\ identified\ by\ the\ system}$$

$$= \frac{x}{l \times k(k-1)/2} \times 100\%$$

where p_k is the percentage of meaningful paths for a specific k; x is the number of meaningful paths identified by subjects; k is the number of source concepts (nodes); l is the number of cases having k source concepts (nodes) in the test bed. Table 1 presents the evaluation results. As the results indicate, the paths identified by the system are meaningful around 80 percent of the time.

Number of souce nodes (k)	Number of cases (l)	Number of paths per case ($k(k-1)/2$)	Number of meaningful paths (x)	Pctg of meaningful paths (p)
2	8	1	6	75.00%
3	9	3	22	81.48%
4	8	6	38	79.17%
5	5	10	42	84.00%

Table 1: Quality of the association paths.

4.3.2 Simulations: Efficiency Issue

Efficiency may be an important issue when a crime analysis is a time-critical task. Higher speed may help law enforcement agencies solve crimes more effectively.

In order to investigate the efficiency of the three algorithms (Dijkstra, PFS, and two-tree PFS), we performed a series of simulations with k ranging from 2 to 5. Given a specific k, 100 sets of k source concepts were randomly generated. The execution time was recorded for each algorithm. Table 2 summarizes the results obtained from these simulations. The results show that the two-tree PFS algorithm is the fastest for computing the shortest paths, confirming the results in (Helgason, Kennington, and Stewart, 1993).

Algorithm	$k = 2$	$k = 3$	$k = 4$	$k = 5$
Dijkstra	1,170.4 (627.1)	3,388.2 (1162.7)	7,082.6 (1,517.3)	12,263.8 (1,517.3)
PFS	1,000.8 (543.1)	2,886.8 (967.8)	6,004.2 (1,259.7)	10,669.6 (2,092.8)
Two-tree PFS	351.3 (186.8)	946.2 (283.5)	1,940.0 (373.2)	3,447.9 (654.2)

Table 2: Mean execution time (in milliseconds) for the three shortest path algorithms.
(Numbers in parentheses are standard deviations.)

5. Conclusions and Future Work

In this paper, we present a link analysis technique based on shortest path algorithms (Dijkstra, PFS, and two-tree PFS) to identify the strongest associations between two or more entities in a criminal network. The experimental results have demonstrated the potential of these three shortest path algorithms. The three algorithms always produced identical results but the two-tree PFS algorithm in most cases was the fastest. Moreover, the association paths identified were meaningful for 80 percent of the time.

Future research can proceed to incorporate domain-specific heuristic functions to help the system extract only meaningful associations from texts. On the other hand, visualization tools may be developed to help users visualize the association paths retrieved.

References

Anderson, T., Arbetter, L., Benawides, A., and Longmore-Etheridge A. (1994). Security works. *Security Management*, 38(17), 17-20.

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269-271.

Evans, J. and Minieka, E. (1992). *Optimization Algorithms for Networks and Graphs*, 2nd ed., Marcel Dekker, New York.

Horn, R. D, Birdwell, J. D., and Leedy, L. W. (1997). Link discovery tool. *Proceedings of the Counterdrug Technology Assessment Center* and *Counterdrug Technology Assessment Center's ONDCP/CTAC International Symposium*, Chicago, IL. August 18-22, 1997.

Helgason, R. V., Kennington, J. L., and Stewart, B. D. (1993). The one-to-one shortest-path problem: Am empirical analysis with the two-tree Dijkstra algorithm. *Computational Optimization and Applications*, 1, 47-75.

Chen H., and Lynch, K. J. (1992). Automatic construction of networks of concepts characterizing document databases. *IEEE Transactions on Systems, Man and Cybernetics*, 22(5), 885-902.

Cormen, T. H, Leiserson, C., E., and Rivest, R. L. (1991). *Introduction to Algorithms*. Cambridge, MA: The M. I. T. Press.

Goldberg, H. G., and Senator, T. E. (1998). Restructuring databases for knowledge discovery by consolidation and link formation. In *Proceedings of 1998 AAAI Fall Symposium on Artificial Intelligence and Link Analysis* (Menlo Park CA, 1998). AAAI Press.

Goldberg, H. G. and Wong, R. W. H. (1998). Restructuring transactional data for link analysis in the FinCEN AI system. In *Proceedings of 1998 AAAI Fall Symposium on Artificial Intelligence and Link Analysis* (Menlo Park CA, 1998). AAAI Press.

Hauck, R. V., Atabakhsh, H., Ongvasith, P., Gupta, H., and Chen, H. (2002). COPLINK concept space: An application for criminal intelligence analysis. *IEEE Computer Digital Government Special Issue*, 35(3), 30-37.

Lee, R. (1998). Automatic information extraction from documents: A tool for intelligence and law enforcement analysts. In *Proceedings of 1998 AAAI Fall Symposium on Artificial Intelligence and Link Analysis* (Menlo Park CA, 1998). AAAI Press.

Sparrow, M. (1991). The application of network analysis to criminal intelligence: An assessment of the prospects. *Social Networks,* 13, 251-274.

Tolle, K. M., and Chen, H. (2000). Comparing noun phrasing techniques for use with medical digital library tools. *Journal of the American Society for Information Science*, 51(4), 352-370.