Identifying Criminals’ Interactive Behavior and Social Relations Through Data Mining on Call Detail Records

YOURONG FAN, TAO YANG, GUOQING JIANG, LEI ZHU and RUXIANG PENG

ABSTRACT

A new method of identifying criminals’ interactive behavior and social relations based on graph computation and clustering model is proposed in this paper. Firstly, a graph database is created in Neo4j to generate a knowledge graph on the basis of calling and crime datasets, and graph traversal and algorithms are implemented to identify criminals’ interactive patterns and practical clues for detecting suspects. Secondly, nine typical features are extracted from calling records. Thirdly, a Gaussian mixture model based on the features is built using Python, and the distributed computing of the model is achieved in Spark. The experiments prove the graph traversal and algorithms can reveal how criminals contact with each other, and the Gaussian mixture model can identify 5 kinds of calling patterns. Furthermore, this study finds that two criminals usually keep a special interactive pattern, in which duration of keeping in contact and holding time are long, the amount of calls is large and the calls happen in the wee hours mostly.

KEYWORDS
Criminal calls, Interactive Behavior, Graph Computation, Clustering, and Data Mining.

INTRODUCTION

In the age of big data, a variety of different types of data have been collected from our daily life, and everyone can be described by data mining. The dataset of call detail records (CDR) is a typical example, because its amount is huge and it reveals people’s calling patterns, relationships and social attributes. Two primary methods are used to analyze CDR, i.e., social network analysis (SNA) and clustering model. Lots of researches have been done. Zhao et al (2008) proposed several features in clustering of call records, such as monthly cost, total duration [1]. W. Xu et al (2008) analyzed subscribers’ calling patterns to detect anomalies [2]. Ro et al (2001) found relationships defined as friends and acquaintances using a Gaussian mixture model [3]. Cheng et al (2010) proposed a technique of discovering communities which can be applied to a large scale of data [4]. Zhou et al (2010) extracted social attributes and eight activity modes using hierarchical conditional random fields [5]. Hu et al (2012) built a community detection algorithm based on campus social network [6]. Zhang et al (2010) proposed an outlier detection algorithm in CDR based on skeleton points [7].

Yourong Fan, Tao Yang, Guoqing Jiang, Lei Zhu, Ruxiang Peng, Third Research Institute of Ministry of Public Security, Shanghai, China.
Bianchi et al (2016) proposed two implementations of the data mining procedure, LD-ABCD and PROCLUS, to identify patterns and regularities [8]. Nattapon et al (2016) employed SNA to analyze the influencers in a social circle [9]. An effective approach of pattern recognition and machine learning procedures has been successfully employed in many fields [10]. We can find the study on identification of criminals’ interactive pattern using by calling records is few in previous researches.

This paper proposes a practical solution not only identifying criminals’ interactive behavior but also offering inner clues for detecting suspects. And the solution can be applied to large scale of data set and different kinds of application situations.

**KNOWLEDGE GRAPH ANALYSIS**

CDR consists of phone numbers and calling records. A social network, in which phones are nodes and calling records are edges, is generated after CDR is imported into Neo4j which is chosen as the graph database. As well, the data of criminal case is imported into Neo4j, in which the case numbers are nodes and crime records are edges that connect related phones and cases. The combination of the two datasets generates a knowledge graph, from which valuable information can be gained by data mining.

**Importing data into Neo4j**

The source data includes two datasets, CDR and crime information. The CDR we use is randomly selected from a telecommunication operator and the calling time is during January and December in 2015. The crime information is confidential files in our research institute and indicates whether a phone number is related to crime cases. The sample data in this paper is anonymous considering confidentiality and privacy.

Firstly, the source data is cleaned and parsed to the required format that can be imported into Neo4j. Then four data files is gained, case nodes, phone nodes, calling edges, crime edges. The samples are shown in table I to table IV respectively. After loading into Neo4j, the knowledge graph contains 201.6 million nodes and 3.6 billion edges.

**TABLE 1. EXAMPLE OF CASE NODE.**

<table>
<thead>
<tr>
<th>caseId:ID</th>
<th>case number</th>
<th>:LABEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>B310******5</td>
<td>B310******5</td>
<td>case</td>
</tr>
</tbody>
</table>

**TABLE 2. EXAMPLE OF PHONE NODE.**

<table>
<thead>
<tr>
<th>phonedId:ID</th>
<th>phoneNumber</th>
<th>:LABEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>13663750067</td>
<td>13663750067</td>
<td>phone</td>
</tr>
</tbody>
</table>

**TABLE 3. EXAMPLE OF CRIME EDGE.**

<table>
<thead>
<tr>
<th>:START_ID</th>
<th>:END_ID</th>
<th>TAKE_TIME</th>
<th>:TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1502***</td>
<td>B310******5</td>
<td>2014/12/4 14:34</td>
<td>crime</td>
</tr>
</tbody>
</table>

**TABLE 4. EXAMPLE OF CONTACT EDGE.**

<table>
<thead>
<tr>
<th>:START_ID</th>
<th>:END_ID</th>
<th>Times</th>
<th>:TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1865***</td>
<td>1860***</td>
<td>84</td>
<td>contacts</td>
</tr>
</tbody>
</table>
Graph traversal and algorithms

Criminals’ Interactive mode can be detected during complex, deep traversals across the graph, and the traversals can implemented through graph query and graph algorithms. The APOC library which consists of various graph algorithms, including Shortest Path algorithm, PageRank, Centrality, Community Detection, has been deployed into Neo4j, and can be called through Cypher language directly.

CYPHER QUERY

Neo4j’s open graph query language, Cypher, is a declarative, SQL-inspired language for describing patterns in graphs visually using an ascii-art syntax. Cypher is used in this study to match particular patterns of nodes and relationships in graph. For example, a node with a particular phone number and related edges with particular values can be matched and returned by using Cypher.

SHORTEST PATH ALGORITHM

Shortest Path algorithm calculates path between a pair of nodes such that the number of edges the path passes through is minimized. A common algorithm used is Dijkstra. By running the algorithm, the shortest path between two particular phones can be gained to discover their contacting chains. Furthermore, all shortest paths between two phones can be gained to discover their potential relationships.

BETWEENNESS CENTRALITY ALGORITHM

Betweenness Centrality algorithm is a measure of centrality in a graph based on all shortest paths. The betweenness centrality of each vertex is the number of these shortest paths that pass through the vertex, so it indicates a node’s influence in a graph. Using this algorithm, the pivotal person in a social circle can be detected. The pivotal person can be the key clue in solving a case. And the formula is shown as follow.

\[
C_B (p_i) = \sum_{j=1}^{N} \sum_{k=1}^{j-1} \frac{g_{jk}(p_j)}{g_{jk}}
\]

In the formula, \( g_{jk} \) is the total number of shortest paths connecting \( p_j \) and \( p_k \), and \( g_{jk}(p_i) \) is the number of the shortest paths that pass through \( p_i \).

PAGERANK ALGORITHM

PageRank algorithm is Google’s popular search algorithm. The underlying assumption is that more important websites are likely to receive more links from other websites. The algorithm is used to calculate nodes’ importance in a graph, and detect core people in a social circle. A big event could occur when two important suspects interact. The formula of the PageRank is proposed as follow.

\[
PR(A) = (1-d) + d \left( \frac{PR(T1)}{C(T1)} + \ldots + \frac{PR(Tn)}{C(Tn)} \right)
\]
In the above formula, node A has pages T1 to Tn which point to it. PR (A) is the PageRank of node A. The parameter d is a damping factor which can be set between 0 and 1, and it’s set to 0.85 in this paper. Also C (A) is defined as the number of links going out of node A.

Application in real situations

USING CYPHER TO FIND RELATED INFORMATION

In real criminal cases, analysts need to locate a suspect in a social network and observe related calling records and crime files to detect useful clues. Many important clues for cracking a criminal case could be found during the process. For example, we need to find a criminal whose phone number starts with “136018” and has a crime record occurred in 2016. The corresponding Cypher query is as below:

```
MATCH (Person:Phone)-[:crimes]->(Crime:Case)
WHERE Person.number STARTS WITH "136018"
And Crime.time STARTS WITH "2016"
RETURN Person.number AS suspect, Crime.ID as CrimeID;
```

The query above returns the criminal’s phone and related case number in the visualization web of Neo4j. And, in the web, we can click “expand” on a node to drill down the graph, so as to find people with whom the criminal has contacted. Figure 1 displays the query result. Node “a” is the criminal, and node “b” is related criminal case. Besides, by expanding node “a” and node “b”, all nodes related are shown. Through the graph traversal, the criminal’s contacts and the person involved in a same criminal case can be found to offer the police valuable clues to investigate.

USING SHORTEST PATH TO FIND CONTACT CHAINS

When two criminals don’t have direct contact record, Shortest Path algorithm can find the shortest contact chain between them. In a real situation, two criminals in a same case don’t have any relation seemingly, however, their contact chains are gained through “allShortestPaths” algorithm. Thereby, the criminals’ contacting mode and the whole criminal gang can be detected. The algorithm can be called as below:
MATCH Paths = allShortestPaths{ (criminal1: Person { Phone:"1531****"}) -[:contacts*..]-( criminal2: Person { Phone:"1305***"}) }
RETURN Paths;

Computing result of the algorithm is shown in fig. 2. Five suspicious middle nodes between the two criminals are detected as new suspects.

ITY TO FIND PIVOTAL NODES

By computing betweenness centrality in a subset of the graph, as the code below, node “a” and node “b” with highest betweenness are located.

MATCH {suspect1:phone {name:"15317***"}}
CALL apoc.path.subgraphAll(criminal1, {maxLevel:5}) YIELD nodes
WITH collect(nodes) AS nodes
CALL apoc.algo.betweenness(["contacts"], nodes, 'BOTH') YIELD nodes, score
RETURN nodes, score
ORDER BY score DESC

As fig. 3 shows, node “a” and node “b” serve as a bridge from criminal 1 to criminal 2, and any path between them must go through this bridge. This makes the two nodes pivotal, so they are key clues for the police to arrest the criminal gang.

UTING PAGERANK TO FIND CORE NODES

When a criminal’s phone has been confirmed, to find his confederates, it is important to find the core people in his social circle. PageRank is called as below.

Figure 2. Example of Shortest Path algorithm.

Figure 3. Example of Betweenness Centrality algorithm.
MATCH (n:phone) WHERE n.name = "15317***"
CALL apoc.path.subgraphAll(n, {maxLevel:2}) YIELD nodes
CALL apoc.algo.pageRank(nodes) YIELD node, score
RETURN node.name AS name, score
ORDER BY score DESC
LIMIT 10;

The computing result is shown in table V. On the basis of the assumption, the node is more important when it receives more links from other nodes. So, in a criminal case, people with high PageRank score could be the core of the social circle.

Furthermore, the result is returned in the visualization web, as figure 4 shows, the top three of PageRank scores are node “a”, node “b” and node “c”, and the three nodes are at core locations in this graph. This helps the police find key person to investigate, so as to detect criminal gangs.

<table>
<thead>
<tr>
<th>Phone</th>
<th>PageRank Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;1381***&quot;</td>
<td>3.6644</td>
</tr>
<tr>
<td>&quot;1379***&quot;</td>
<td>3.39745</td>
</tr>
<tr>
<td>&quot;1312***&quot;</td>
<td>3.11938</td>
</tr>
<tr>
<td>&quot;1500***&quot;</td>
<td>2.50453</td>
</tr>
<tr>
<td>&quot;1363***&quot;</td>
<td>1.30164</td>
</tr>
</tbody>
</table>

Figure 4. Example of PageRank algorithm.
CLUSTERING ANALYSIS

Different social relationships leading to distinctions of calling behavior patterns, such as, most of callings in a steady working relationship occur in working hours and the span of the relationship lasts for a long time, and most of callings in a family relationship occur in spare time and the calling frequency is relatively high. So this study extracts calling features to build a clustering model which can detect different kinds of relationship. Combined with crime information, the calling behavior of criminals can be mined to discover potential suspects.

Data preprocessing

By joining CDR and crime information, the phones are labeled with “criminal” or “not criminal”, as fig. 5 shows. The data after preprocessing is described in table VI. As the calling records are selected from a telecommunication operator, “main_phone” represents the phone number of a subscriber, and “contact_phone” represents the subscriber’s contacts’ phone number. The summary of the data is shown in table VII.

![Figure 5. Data preprocess.](image)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>main phone</td>
<td>The phone number of a subscriber</td>
<td>151****4011</td>
</tr>
<tr>
<td>contact phone</td>
<td>The phone number of a contact</td>
<td>156****7709</td>
</tr>
<tr>
<td>begin time</td>
<td>The start time of the call</td>
<td>2015-12-01T11:54:36.000Z</td>
</tr>
<tr>
<td>use time</td>
<td>The duration of the call</td>
<td>57(second)</td>
</tr>
<tr>
<td>case_id1</td>
<td>the case number, when the subscriber is criminal, otherwise, “NULL”</td>
<td>B310******5</td>
</tr>
<tr>
<td>case_id2</td>
<td>the case number, when the contact is criminal, otherwise, “NULL”</td>
<td>NULL</td>
</tr>
</tbody>
</table>

TABLE 6. DATA EXAMPLE AFTER PREPROCESS.

<table>
<thead>
<tr>
<th>Field</th>
<th>Amount of Call Detail Records</th>
<th>Amount of distinct main phone</th>
<th>Amount of distinct contact phone</th>
<th>Amount of distinct main phone which is criminal</th>
<th>Amount of distinct couple of main phone and contact phone which both are criminal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3.6 billion</td>
<td>1.6 million</td>
<td>200 million</td>
<td>8 thousand</td>
<td>5 hundred</td>
</tr>
</tbody>
</table>

665
Extracting features

A subscriber has different calling behavior with different contacts, because of their particular relationships. Five typical examples which represents five types of calling patterns are shown in fig. 6.

As shown in fig. 6, different colors represent different interactive types in 12 days, and each day is divided into 48 segments with 30 minutes each. And, red represents two criminals’ calls in which they only contact between 0 o'clock and 6 o'clock in the morning with high frequency, green represents two colleagues who call during work with high frequency, pink represents a temporary working relationship in which calling occurs just once during working time, blue represents two friends contact in working hours or resting time with relatively low frequency, and yellow represents a couple contact each other after work very frequently.

From the example in figure 6, we can find the two criminals’ interactive mode is different from others, so we attempt to verify whether criminals contact with their criminal partners mostly in the wee hours.

In this paper, the timeline in one day is split into three major periods, the wee hours (from 0 a.m. to 6 a.m.), working time (from 7 a.m. to 6 p.m.), and resting time (from 7 p.m. to 11 p.m.). The ratios of call durations of the three time periods in three situations are computed, and the result is shown in fig. 7. From the figure, we can find when two contacts are not criminal or just one of them is criminal, the calling patterns are similar and the ratio of call duration in spare time is the highest, but when both the contacts are criminal, the ratio of call duration in the wee hours is the highest. Furthermore, we find that two criminals in one case have rather high calling frequency during the wee hours, as fig. 8 shows below.
Figure 7. The ratios of call duration in three situations.

Figure 8. Example of two contacts in one criminal case.

<table>
<thead>
<tr>
<th>Feature</th>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>diff</td>
<td>features0</td>
<td>The duration of two people keep in contact</td>
</tr>
<tr>
<td>cnt</td>
<td>features1</td>
<td>Total number of calls of two people</td>
</tr>
<tr>
<td>length</td>
<td>features2</td>
<td>Total holding time of two people’s calls</td>
</tr>
<tr>
<td>whour_cnt</td>
<td>features3</td>
<td>Total number of calls of two people during working time</td>
</tr>
<tr>
<td>whour_length</td>
<td>features4</td>
<td>Total holding time of two people’s calls during working time</td>
</tr>
<tr>
<td>rhour_cnt</td>
<td>features5</td>
<td>Total number of calls of two people during resting time</td>
</tr>
<tr>
<td>rhour_length</td>
<td>features6</td>
<td>Total holding time of two people’s calls during resting time</td>
</tr>
<tr>
<td>am_cnt</td>
<td>features7</td>
<td>Total number of calls of two people in the wee hours</td>
</tr>
<tr>
<td>am_length</td>
<td>features8</td>
<td>Total holding time of two people’s calls in the wee hours</td>
</tr>
</tbody>
</table>

On the basis of the analysis above, nine features are extracted from the calling records to identify different types of calling pattern. Descriptions of the nine features are shown in table VIII.
The clustering method of Gaussian mixture model (GMM) is used in this paper, and the model is based on the nine features in table VIII. In order to improve computational efficiency, the model is implemented in a distributed platform based on hadoop2.7.3, spark2.0.1. The main code is shown below.

```
# import data
data = spark.sql('select * from tablename')
# generate vectors of features
vector = data.rdd.map(lambda line:(line[0],line[1],Vectors.dense(...)))
dataFrame = spark.createDataFrame(vector,['main_phone','contact_phone','features'])
# normalize the data
normalizer = Normalizer(inputCol="features", outputCol="normFeatures", p=1.0)
l1NormData = normalizer.transform(dataFrame)
# running GMM model
gmm = GaussianMixture(featuresCol="normFeatures")
model = gmm.fit(l1NormData)
# generate the clustering result
transformed = model.transform(l1NormData).select("main_phone", "contact_phone", "features","prediction")
```

The clustering result is visualized in nine dimensions, as fig.9 shows, by using RadViz which is a way of visualizing multi-variate data in pandas, so as to view and analyze characteristics of each cluster on the basis of the nine features in table VIII. After trying different parameters in the GMM model, the optimal clustering result is achieved and each cluster in fig.9 have clear characteristics.

Figure 9. Clustering visualization.
The data amounts in every cluster and the corresponding percentages are calculated, grouping by three situations. As table IX shows, the percentages’ distributions of situation 1 and situation 2 are similar, but situation 3 is noticeably different as its percentage of cluster 0 is the highest. So we can conclude that calls of situation 3 have different characteristics, and this conclusion is consistent with the findings in fig. 7.

In order to illustrate each cluster’s unique characteristics, the average values of 5 typical features in every cluster have been calculated and normalized, and are shown in radar maps. As fig. 10 to fig. 14 show below.

TABLE 9. COMPONENT OF EACH CLUSTER.

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Situation 1 (neither of main_phone and contact_phone is criminal)</th>
<th>Situation 2 (only main_phone is criminal)</th>
<th>Situation 3 (both main_phone and contact_phone are criminal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster 0</td>
<td>5.43%</td>
<td>9.49%</td>
<td>37.63%</td>
</tr>
<tr>
<td>Cluster 1</td>
<td>9.61%</td>
<td>13.90%</td>
<td>19.40%</td>
</tr>
<tr>
<td>Cluster 2</td>
<td>28.31%</td>
<td>26.49%</td>
<td>15.26%</td>
</tr>
<tr>
<td>Cluster 3</td>
<td>39.63%</td>
<td>36.03%</td>
<td>18.96%</td>
</tr>
<tr>
<td>Cluster 4</td>
<td>17.02%</td>
<td>14.08%</td>
<td>8.75%</td>
</tr>
</tbody>
</table>

Figure 10. Characteristics of cluster 0.

Figure 11. Characteristics of cluster 1.

Figure 12. Characteristics of cluster 2.
Based on fig.9 to fig.14, main characteristics of each cluster can be summarized and described as below.

Cluster 0: The contacts keep in contact for a long time, with max number of calls and long holding time, and the calls occur mostly in the wee hours.

Cluster 1: The contacts keep in contact for a relatively short time, with less call and short holding time, and the calls occur mainly in the wee hours.

Cluster 2: The contacts keep in touch for a median time, with less call and relatively short holding time, and the calls occur mostly in resting time.

Cluster 3: The contacts keep in contact for a relatively long time, with relatively large number of calls and relatively long holding time, and the calls occur mainly in working time.

Cluster 4: The contacts keep in contact for the shortest time, with the fewest call and the shortest holding time, and the calls occur mostly in resting time.

Different characteristics of the clusters reveal different relationships, so we can assume cluster 0 and cluster 1 reveal a kind of special relationship, for example, criminal gangs. And cluster 2 reveals a close relationship like close friends or families, cluster 3 reveals a working relationship, and cluster 4 reveals a short calling relationship like telesales or nodding friends. Table IX proves that most calls of criminals are in cluster 0, but the relationships in other clusters can’t be testified because of the lack of relationship data.

According the clustering analysis above, we can make a conclusion that when both “main phone” and “contact phone” are criminal, they usually keep in contact for a long time with a big amount of calls and a long holding time, and call each other in the wee hours mostly.

CONCLUSION AND FUTURE WORK

To analyze the criminals’ interactive behavior and potential relationships between criminals and their contacts, a knowledge graph consists of social network and crime information is built in this paper. Cypher query and graph algorithms are implemented in several real criminal situations. And the method is proved effective and efficient to identify criminals’ interactive patterns and detect suspects, so as to offer technical support for cracking a criminal case.

Furthermore, the GMM model based on nine typical features is built, and the experiment based on real data which contains billions of calling records proves the
model can find criminals’ possible relationships like criminal confederates, friends or colleagues. And the model finds two criminals’ main calling characteristics: keeping in contact for a long time, with max number of calls and a long holding time, and contact each other mostly in the wee hours.

In the future, more accurate algorithms and models are to be studied to find criminals’ social relations to offer key clues to the police.

ACKNOWLEDGEMENTS

This research is supported by the Program of Technology and Equipment Research on Criminal Information of Narcotics-control (grant No. 2016YFC0800909), the Program of Research on Internet Security Policies and Laws (grant No. C17253), and the Key Lab of Information Network Security at the Third Research Institute of Ministry of Public Security.

REFERENCES