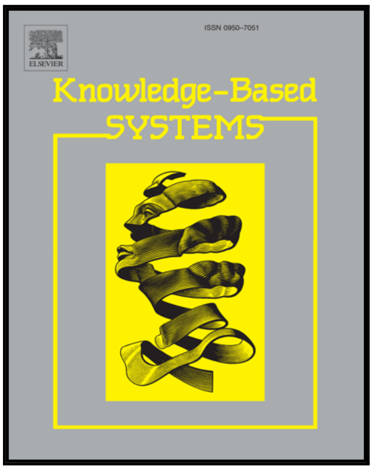


A decision support system for detecting serial crimes

Hong Chi , Zhihong Lin , Huidong Jin , Baoguang Xu ,
Mingliang Qi

PII: S0950-7051(17)30083-7
DOI: 10.1016/j.knosys.2017.02.017
Reference: KNOSYS 3831

To appear in: *Knowledge-Based Systems*

Received date: 9 June 2016
Revised date: 9 February 2017
Accepted date: 11 February 2017

Please cite this article as: Hong Chi , Zhihong Lin , Huidong Jin , Baoguang Xu , Mingliang Qi , A decision support system for detecting serial crimes, *Knowledge-Based Systems* (2017), doi: 10.1016/j.knosys.2017.02.017

A decision support system for detecting serial crimes

Hong Chi[a], Zhihong Lin[a,*], Huidong Jin[b], Baoguang Xu[a], Mingliang Qi[a]

[a] *Institute of Policy and Management, Chinese Academy of Sciences, Beijing 100190, China*

[b] *Software & Computational Systems, Data61, CSIRO, GPO Box 664, Canberra ACT 2601, Australia*

Abstract

Serial crimes pose a great threat to public security. Linking crimes committed by the same offender can assist the detecting of serial crimes and is of great importance in maintaining public security. Currently, most crime analysts still link serial crimes empirically especially in China and desire quantitative tools to help them. This paper presents a decision support system for crime linkage based on various, including behavioral, features of criminal cases. Its underlying technique is pairwise classification based on similarity, which is interpretable and easy to tune. We design feature similarity algorithms to calculate the pairwise similarities and build up a classifier to determine whether a case pair should belong to a series. A comprehensive case study of a real-world robbery dataset demonstrates its promising performance even with the default setting. This system has been deployed in a public security bureau of China and running for more than one year with positive feedback from users. The use of this system would provide individual officers with strong support in crimes investigation then allow law enforcement agency to save resources, since the system not only can link serial crimes automatically based on a classification model learned from historical crime data, but also has flexibility in training data update and domain experts interaction, including adjusting the key components like similarity matrices and decision thresholds to reach a good tradeoff between caseload and number of true linked pairs.

Keywords:

Serial crimes; Crime linkage; Pairwise case linkage; Robbery; Hierarchical similarity;

[*] Corresponding author. Tel.: +86 13581726994. The work was partially done when he was visiting CSIRO, Australia.

E-mail addresses: chihong@casipm.ac.cn (H. Chi), linzhihong1991@gmail.com (Z. Lin), warren.jin@csiro.au (W. Jin), xbg@casipm.ac.cn (B. Xu), mlqi@casipm.ac.cn (M. Qi)

1. Introduction

Serial crimes can be defined as multiple criminal incidents committed by the same offender or group of offenders (e.g., a gang) [11, 48]. According to some studies, a large proportion of crimes are committed by a minority of offenders [8, 18, 43]. For example, in the USA, researchers found that 5% of offenders are involved in 30% of crimes [43]. Consequently, police officers are usually required to link serial crimes or detect series of crimes in their daily work for case investigation. Linking serial crimes is of great significance to law enforcement for several reasons. Firstly, the aggregation of information from different crime scenes increases the amount of available evidence. For example, the somatotype of the offender seen in one case and the accent of the offender heard in another case could be utilized together to profile an offender if these two cases are identified as cases in the same series. Secondly, the joint investigation of multiple crimes enables a more efficient use of law enforcement resources [50].

Among all types of crimes, robbery is frequently occurred. Although the social hazardous nature of a single robbery case is lower than a murder case, robbery also poses a serious threat to the social public security due to its relatively high frequency. Hence, the efficiency of robbery case linkage and investigation is significant in fighting against the crime of robbery, maintaining the social security order and building a harmonious society.

After investigating the work of crime analysts, we find that they mainly link serial crimes manually especially in China. It is tedious and time-consuming for crime analysts to manually search incidents in the database, read through incidents' reports and compare incidents to find out suspicious serial cases. Considering that caseload is huge, police officers and other resources are relatively scarce, an intelligent tool is desired to improve the work efficiency of police officers.

According to the experience of crime analysts, they usually divide the basis of crime linkage into two classes: direct basis and indirect basis. Direct basis includes the identification of forensic evidence like the DNA, fingerprint etc. of offenders in different cases, the identification of the physical traits of offenders in different cases and the identification of the remnants in different crime scenes. Indirect basis includes the identification or similarity of temporal and spatial features, the identification or similarity of the target, the similarity of modus operandi (MO) features, the similarity of offenders' posture and facial features, etc. [19, 30]. Although the direct basis, such as forensic evidence could be used to determine different cases that belong to a series exactly, it rarely appears in the robbery cases. So, in practice, it is unable to link serial robbery cases using direct basis.

In the absence of direct basis, the problem of crime linkage could be defined as how to use behavior information as main alternative data source to link serial crimes. In literature, many researchers have illustrated that it is feasible to detect serial crimes on the basis of behavior information. Bennell and Canter argued that the behavioral information such as MO, temporal and spatial features can be used to distinguish linked case pairs from unlinked pairs [4]. Tokin et al. concluded that both the distance and the time proximity between crimes have a significant difference between linked and unlinked crimes [43]. Woodhams et al. showed that linked pairs in two types of crimes, robbery and rape, were significantly more similar in MO behavior [51, 52].

This article presents a decision support system (DSS) for linking serial crimes automatically based on various information, including the behavioral information (see Fig. 1, Fig. 2, and Fig. 3), motivated by the need to support the crime investigation decision of law enforcement. It has two key functions. The first one is data collecting and processing. The second one is case pair's similarity calculation and suspicious linked case pair prediction. This paper focuses on describing its core model and algorithms with a case study demonstration of robbery in mind. A pairwise similarity-based approach framework is proposed to link serial crimes on the basis of temporal, spatial and MO features of robbery cases. The framework consists of features' similarity measures, classification model and an associated coefficient learning algorithm. A real-world robbery dataset is used to evaluate the effectiveness and efficiency of our approach.

The contribution of this study is twofold.

First, a DSS for data collecting, processing and serial crimes linking is presented. The system absorbs the knowledge of domain experts reasonably. For example, the crime features structures and some similarity tables are designed after discussing with experts, and crime analysts could change some similarity coefficients and thresholds during the process of using this tool according to their actual demand.

Secondly, as for the underlying techniques of this system, we propose our approach framework by making improvement on some existing techniques. For example, several similarity algorithms are proposed accordingly. Especially the hierarchical similarity algorithm is superior to the taxonomic similarity algorithm in terms of both effectiveness and efficiency. Besides, a similarity network structure is designed to formulate the classification model in which the two-phase parameters could be learned simultaneously. The underlying techniques have a promising performance in terms of precision and recall as well as interpretation and computational efficiency by using the default model and settings. It enables crime analysts to achieve a high precision of crimes linkage while only comparing and analyzing a small proportion of case pairs.

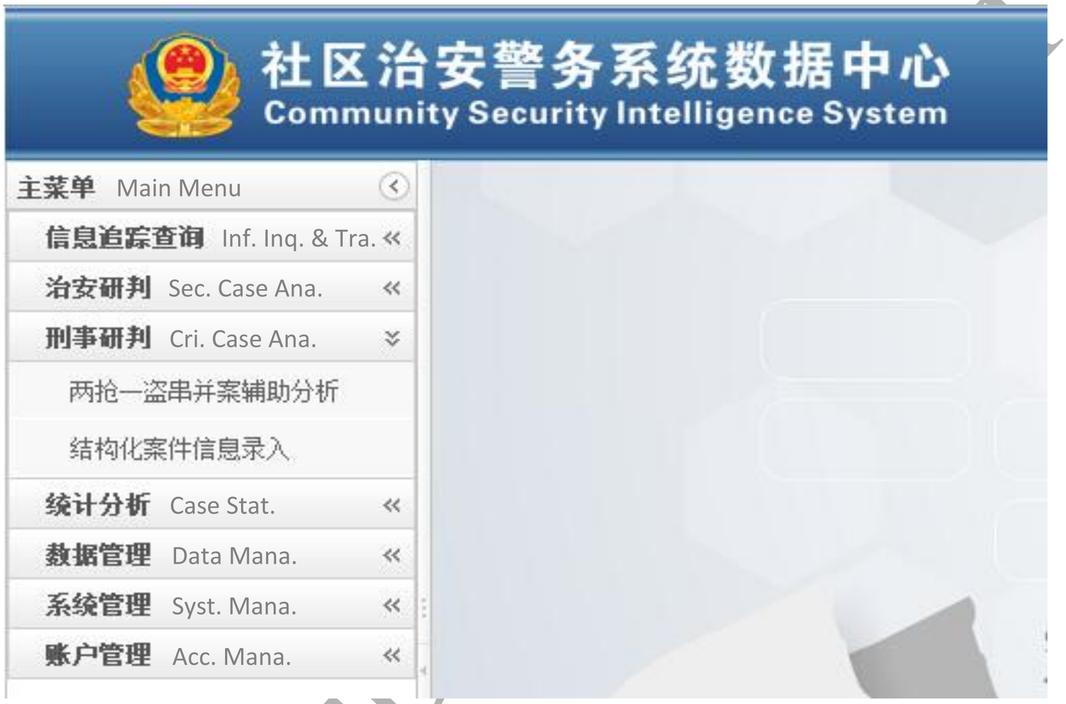


Fig. 1. The graphic user interface of the proposed DSS (More details see Fig. 2 and Fig. 3)

The structure of the paper is listed as follows. A brief summary and review of relevant researches are presented in Section 2. Then we introduce our system and its underlying techniques including a mathematical model of the crime linkage problem, similarity algorithms for different types of features and the parameter learning method for the classification model in Section 3. We introduce the real-world robbery dataset in a city of China as well as performance measures in Section 4. Section 5 presents the results of our method in the case study and provides some comparisons and discussions. Finally, we conclude the article and discuss our future works in Section 6.

2. Literature review

2.1. Decision support system for serial crimes linkage

Motivated by the need to improve the work efficiency of crime linkage and investigation, researchers have developed a number of automated tools or systems to support police operations [2, 10, 27, 35, 42]. We put our emphasis on the serial crime linkage systems. From the very beginning, a system called the Violent Criminal Apprehension Program (ViCAP) was developed by the Federal Bureau of Investigation (FBI) to aid in crime analysis [21]. It compared over 100 MO attributes of an incident with the other cases in a database and returned the top 10 “matches” that were most similar to the case being examined. However, neither its underlying method nor performance was reported. Another system called the Violent Crime Linkage Analysis System (ViCLAS) was first introduced and applied in the Royal Canadian Mounted Police (RCMP) to link crimes such as serial homicide and sexual assaults [23]. It had been adopted by some other countries such as Australia and the Netherlands. However, this system cannot link serial crimes automatically and only trained specialists are able to use it to link crimes. Besides, some test of the ViCLAS showed that the data contained in the system may be unreliable [33, 40].

Furthermore, the Regional Crime Analysis Program (ReCAP) introduced by Brown and used by several police jurisdictions in Virginia could automatically calculate the similarity of case pair to support robbery case linkage decision [11]. Recently, Borg et al. presented a decision support system for data collecting as well as comparing and analyzing residential burglaries [8]. A cut-clustering approach was used to group crimes to reduce the amount of crimes to review for residential burglary analysis based on modus operandi, residential characteristics, stolen goods, spatial similarity, or temporal similarity. The above-mentioned systems all show that this kind of systems could help the crime analysts in serial crimes detection to some extent.

2.2. Serial crimes linkage models and approaches

As a core part of the serial crimes linkage system, the mathematical models and approaches for linking serial crimes have been introduced and evaluated in many researches. The approaches could be classified into at least two types, pairwise case linkage and crime series identification [37]. The former involves identifying whether a pair of cases was committed by the same offender, which could be regarded as a binary classification problem, while the latter is a process of identifying group of crimes that were committed by the same offender, which is similar to a cluster problem.

As a representative of pairwise case linkage researches, Brown and Hagen developed a similarity-based approach for crime association [11]. They used both expert's scores and a statistical method to determine the weight of each attribute, i.e. similarities between behavior features. The results showed that their methods were both faster and more accurate than analysis using the structured query language (SQL). Similarly, Albertetti et al. presented a fuzzy MCM approach combining spatio-temporal, behavioral and forensic information for implementing a tailor-made crime linkage system and applied their method to link residential burglaries [1]. There have been many other studies detailing the performance of pairwise case linkage method across a variety of crime types [3, 32, 43, 44, 51].

As for the crime series detection approaches, Lin and Brown presented an outlier-based cluster method [29]. They argued that a group of incidents is likely to be committed by the same criminal when they are not only similar to each other, but also distinctive. They clustered the crimes according to the outlier score which was designed to evaluate this distinctiveness. More common attributes among a group of incidents and a lower frequency of these attributes' combination lead to high outlier score. However, this distinctiveness-based method relies heavily on the data quality and scale. Borg used a cut-clustering approach to group crimes which reduce the amounts of cases to review while keeping most connected cases [8]. Moreover, Wang et al. designed a semi-supervised method to detect crime series [47]. They first build a model to learn the weights of similarity from the labeled crime data and constructed a similarity graph. Then they used a subspace clustering method to find out all the crime series. Results showed that their method had a better performance than traditional cluster methods such as hierarchical agglomerative clustering. A variety of other crime series detection method have been used to group similar crimes for further investigation or offender profiling [6, 7, 31, 38, 45, 46].

Apart from the above-mentioned case pair classifying approach and serial crimes clustering approach, there were some other researches about serial crime linkage in the literature. For example, Zoete et al. used Bayesian networks to model different evidential structures, analyzed their complex underlying dependencies and how they influence the probability of some crimes to be in the same series [13]. However, evidence such as DNA and shoeprint are seldom found in robbery or burglary. Therefore, the Bayesian networks analysis does not suit for linking these kinds of incidents.

From the methodology viewpoint, because the pairwise case linkage methods use supervised learning to include solved cases and domain knowledge, say, into similarity matrices and classifiers, they often have better performance than unsupervised or semi-supervised learning method. On the other hand, the solved cases can naturally serve as training data so we do not need extra efforts to prepare training data for supervised learning, which is often criticized by unsupervised learning advocates. Besides, the principle of case linkage method conforms to the decision-making process of crime linkage and is easier to be understood by crime analysts. As a result, if some exceptions happen, it is more manageable to tune by the experienced crime analysts.

Since the pairwise case linkage approach has some advantages, such as relatively accurate, interpretable, flexible and easy to develop and tune, we use it as core method in the decision support system. To the best of our knowledge, only a small part of literature build a model with multi-type of features [11, 29]. However, in these papers, the design of categorical attributes is too simple and cannot reflect subtle difference among attribute values. In the robbery incidents, due to the lack of direct evidence, we need to use information from multiple sources to link crimes. Therefore, it is of great significance to build a model with different types of features and taking the knowledge of crime analysts into consideration as well.

2.3. Measures of features' similarity

Former studies have indicated that incidents in a series have similar temporal, spatial and MO features. These features are usually recorded as either quantitative (numeric) or qualitative (categorical) attributes in the criminal case database. There are some research efforts on defining similarity and distance among features, which are grouped together based on the type of attributes.

When data are numeric, distances can be derived starting from the classical Euclidean distance or the Minkowski distance. It has fewer problems in measuring the similarity of quantitative attributes. Brown and Hagen introduced an approach based on the absolute distance between two values to measure the similarity of quantitative attributes such as the suspect's stature [11].

Qualitative attributes are more common in the criminal case database. When data are categorical, they are typically modeled as sets of elements. Hubalek had collected and compared theoretically and empirically 43 similarity coefficients based on qualitative (binary) attribute data used in ecology [20]. These similarity quotients were calculated on the basis of the number of common elements and different elements between two attributes (sets). Several more advanced approaches have been proposed for categorical attributes, such as [28] and [9]. Among these similarity coefficients, Jaccard's coefficient (*J*) is most widely used in crime linkage to define the similarity of categorical attributes. Jaccard's coefficient is defined as $a/(a+b+c)$, where *a* is the overall number of elements shared by the two samples, and *b* and *c* are the numbers of elements unique to sample one and two respectively. Bennell and Canter used the Jaccard's coefficient to measure the similarity of modus operandi features, such as entry behaviors and target selection choices, between a pair of crimes [4].

A similarity index, taxonomic similarity (Δ_S), which was first applied to measure the similarity of marine species, took an expanded view of similarity by utilizing hierarchical information [22]. Woodhams et al. applied it to the task of behaviorally linking serial sexual offenses [49]. They compared Δ_S with *J* and demonstrated that taxonomic similarity had clear advantages over Jaccard's coefficient, especially under conditions of data degradation i.e., missing data.

However, Bennell and Melnyk's results suggested that *J* was as effective as Δ_S when used as a measure of similarity in case linkage, and may slightly outperform Δ_S under certain conditions, e.g., when large samples were examined [5, 34].

Golfarelli et al. summarized and reviewed some definitions of hierarchical distance or similarity. He divided the definitions into two classes: level-based distance and occurrence-based distance [16]. The taxonomic similarity belongs to the former.

Generally speaking, these hierarchical similarity indexes are more rational than common categorical similarity indexes because the former are more effective in measuring the similarity of features with missing data. However, to the best of our knowledge, the existing hierarchical similarity indexes have a relatively high computational complexity (see Section 3.3.3). Besides, the hierarchy weight optimization was not considered in these indexes. Therefore, we need to design an efficient similarity algorithm to take hierarchical information into consideration.

3. System and Method

3.1. Decision Support System description

After around one year investigation, research and discussion with police experts, we have developed a decision support system called Community Security Intelligence System. The aim of this system is to help police office analyze crime related data and support their daily decision making such as crime investigating and police deploying.

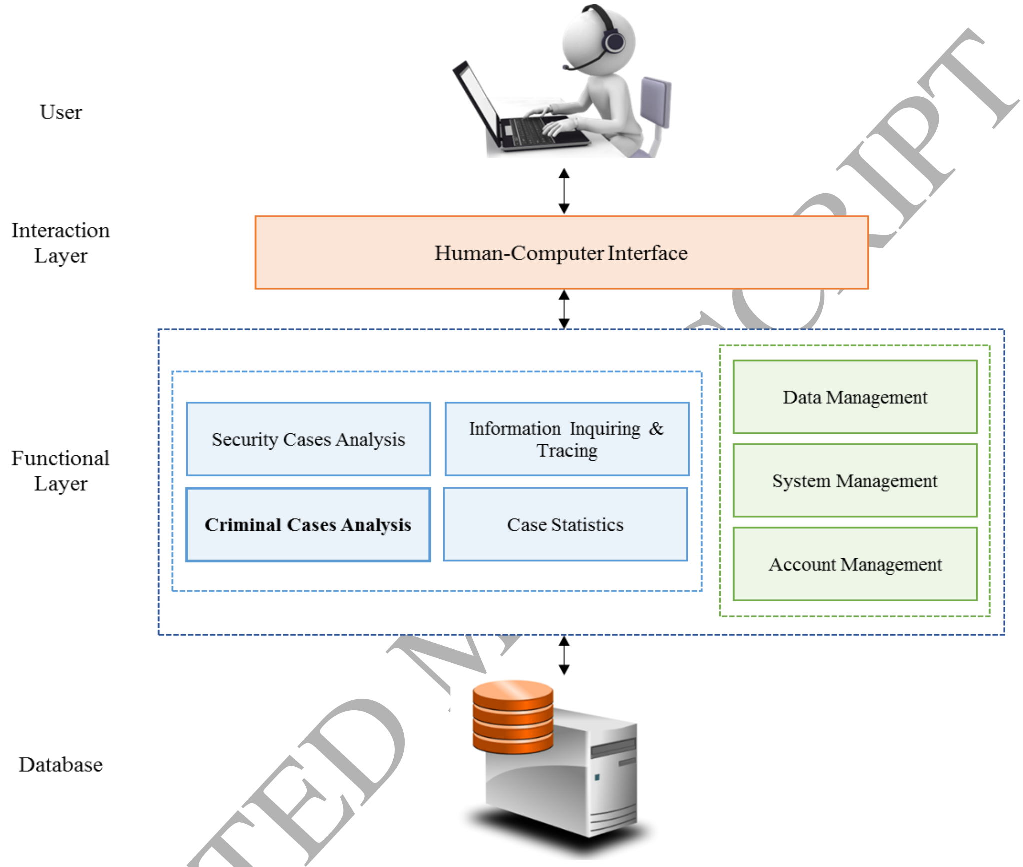


Fig. 2. System framework of the proposed DSS

The system framework is illustrated in Fig. 2. Users access the system function and service modules through the human-computer interface. The crime-related data and other system data are stored in several databases. From the functional perspective, the system consists of four function subsystems/modules including security cases analysis system, criminal cases analysis system, information inquiring & tracing module, and case statistics module; and three service modules including data management module, system management module, and account management. Each subsystem/module has its user interface and exchanges data with databases.

This paper focuses on the criminal cases analysis system. Our robbery case linkage system, as one of developed application examples of the criminal cases analysis system, will be exemplified in detail in this paper.

The robbery case linkage system includes three key modules: data entry and processing, case linkage and model setting. The first two modules are daily-used modules which are designed to collect and process crime data and output case linkage results. Police officers enter case data through the data entry and processing module, and crime analysts use the case linkage module to support the case linkage decision in their daily operation. The third module is a service module which is designed to adjust the case linkage model. When the system is initialized, domain experts and system developer could set some model parameters like similarity matrices and choose the training data through the module. Domain experts also could re-set these parameters and update the training data, if required, after the system is developed.

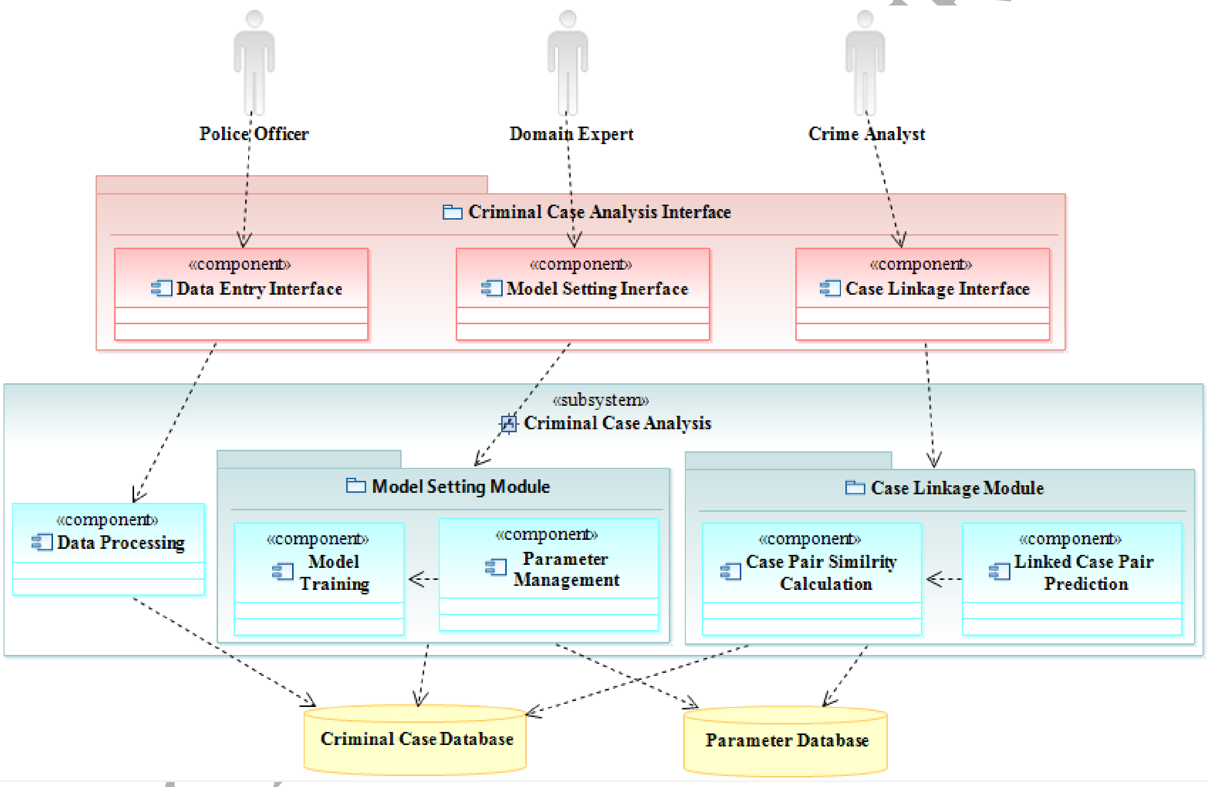

Police Officer
Domain Expert
Crime Analyst
Criminal Case Analysis Interface
«component» Data Entry Interface
«component» Model Setting Inerface
«component» Case Linkage Interface
«subsystem» Criminal Case Analysis
«component» Data Processing
Model Setting Module
«component» Model Training
«component» Parameter Management
Case Linkage Module
«component» Case Pair Similrity Calculation
«component» Linked Case Pair Prediction
Criminal Case Database
Parameter Database

Fig. 3. Component Diagram of our developed robbery case linkage system

We draw the component diagram of the robbery case linkage system to show the relationship among system components and the environment as illustrated in Fig. 3. Broadly speaking, system users interact with the system through the user interface; interface modules then call the corresponding processing modules or packages to realize system functions, and processing modules or packages write data to or read data from databases.

To be specific, police officers could enter the structured case data like modus operandi features, time and locality of cases, and the unstructured case data like texture descriptions through the data entry interface (more details about data formats see Section 18 and Appendix A). The case data would be stored in the criminal case database after being processed and transformed by the data processing module.

Domain experts can interact with the system using the model setting interface. They modify some parameters and adjust the model on the interface if the DSS need to tune. Within the model setting module, the parameter management module not only receive the hype-parameters set by experts but also get the parameters trained on solved criminal case data by the model training module. Then these parameters would be stored in the parameter database.

Crime analysts access to the system through the case linkage interface and use the case linkage results displayed on the interface to support their decision-making. The case linkage interface calls the case linkage module to realize the case linkage function when the use trigger the related command. As a core module of the system, the case linkage module has two key functions including case pair similarity calculation and suspicious linked case pair prediction. On one hand, it could calculate the similarity of selected cases for crime analysts. On the other hand, it could output suspicious linked case pairs according to the search criteria of crime analysts, such as the time period of the crimes or the threshold of similarity. The case linkage module read criminal case data from the criminal case database and model parameter data from the parameter database to complete its operations.

The following sub-sections will introduce the underlying techniques of the case linkage system.

3.2. Problem description

Former studies have indicated that incidents in a series have similar temporal, spatial and MO features. So we can link criminal cases based on some measures of similarity (or dissimilarity). As we discussed above, our system will use a pairwise hybrid similarity-based method. We pair every two cases together, use different similarity algorithms according to the feature's type and build a model to predict whether the pair is committed by the same offender.

Table 1
Notation Description

Notation	Description
N	Number of cases

n	Number of case pairs, $n = N*(N-1)/2$
m	Number of features
r_i	Feature vector of Case *i*, $r_i=\left(r_i^1, r_i^2, \cdots, r_i^m\right)$, where $i \in \{1,2,\cdots,N\}$
S_{ij}	Similarity vector between Cases *i* and *j*, $S_{ij}=\left(S_{ij}^1, S_{ij}^2, \cdots, S_{ij}^m\right)$, where i，$j \in \{1,2,\cdots,n\}$ and $j > i$
Y_{ij}	Indicator of whether Cases *i* and *j* belong to a series, $Y_{ij} \in \{0,1\}$
$\hat{Y}_{ij}$	Result predicted by the model whether Cases *i* and *j* belong to a series
θ	Coefficient vector of features
ω^k	Parameters in the similarity calculation of Feature *k*
n_k	Number of parameters in the similarity calculation of Feature *k*
$f(z)$	Synthesize discriminant function
Sim^k	Similarity algorithm of Feature *k*
$L(\theta,\omega)$	Objective function

Some notations used in this section are listed in Table 1. For the crime linkage problem, suppose that we have *N* solved cases: $\{C_1, C_2, \cdots, C_N\}$, and some of them were committed by the same perpetrator (or same gang), which can be formed $n = N*(N-1)/2$ pairs of cases: $\{P_1, P_2, \cdots, P_n\}$. Each case has *m* features which can be a quantitative or qualitative value or a vector. The feature vector of Case *i* can be denoted as $r_i=\left(r_i^1, r_i^2, \cdots, r_i^m\right)$, where $i \in \{1,2,\cdots,N\}$. The similarity vector between Cases *i* and *j* could be represented as $S_{ij}=\left(S_{ij}^1, S_{ij}^2, \cdots, S_{ij}^m\right)$, where i，$j \in \{1,2,\cdots,n\}$ and $j > i$. Let Y_{ij} be the indicator of whether Cases *i* and *j* belong to a series.

Let $\left(\theta^1, \theta^2, \cdots, \theta^m\right)$ be the coefficient vector of features, and $\omega^k=\left(\omega_1^k, \omega_2^k, \cdots, \omega_{n_k}^k\right)$ be the parameters in the similarity calculation of Feature *k*, where $k \in \{1,2,\cdots,m\}$ and n_k is the number of parameters in the similarity calculation of Feature *k*. The similarity of Feature *k* between Cases *i* and *j* can be represented as $S_{ij}^k = Sim^k\left(r_i^k, r_j^k;\ \omega^k\right)$, where Sim^k is the similarity algorithm of Feature *k*. The discriminant result of whether Cases *i* and *j* belong to a series can be decided by Eq. (1).

$$\hat{Y}_{ij} = f\left(\sum_{k=1}^{m} \theta^k \times S_{ij}^k\right) \tag{1}$$

Let $f(z)=1/\left(1+e^{-z}\right)$, then the synthetic discriminant function of a case pair can be expressed as:

$$\hat{Y}_{ij} = \frac{1}{1+e^{-\sum_{k=1}^{m}\theta^k \times S_{ij}^k}} \quad \text{or}$$
$$\hat{Y}_{ij} = \frac{1}{1+e^{-\sum_{k=1}^{m}\theta^k \times Sim^k\left(r_i^k, r_j^k\,;\,\omega^k\right)}} \tag{2}$$

Our goal is to design similarity algorithms for different types of features and decide the coefficients θ and ω to maximize the following likelihood function.

$$L(\theta,\omega) = \prod_{i,j,j>i}^{N}\left(\hat{Y}_{ij}\right)^{Y_{ij}} \times \left(1-\hat{Y}_{ij}\right)^{\left(1-Y_{ij}\right)} \tag{3}$$

The value range of $\hat{Y}_{ij}$ is between 0 and 1. It can be explained as the similarity between two cases or the probability of whether the two cases belong to a series. The similarity algorithms and coefficient learning methods will be introduced in the following section.

3.3. The similarity algorithms

According to the description in Section 11, we need to determine the similarity algorithms of different types of features before coefficient learning. The features can be divided into three types: continuous (quantitative) attributes, categorical (qualitative) attributes and hierarchical attributes. Thus, we design similarity algorithms for them separately.

3.3.1. The similarity of categorical attributes

A simple and commonly used approach for a categorical attribute adopts a binary similarity measure. Specifically, the similarity of Attribute *k* is 1 if the attribute values exactly match and 0 otherwise. We can make an extension of the basic measure to use some similarity tables to specify the similarity between different values for each categorical variable. Taking the somatotype of the suspect as an example, we could create a similarity table as Table **2**.

Table 2

Example similarity table for somatotype of suspect

Somatotype	Very Thin	Thin	Medium	Fat	Very Fat	NA
Very Thin	0.8	0.6	0.3	0.1	0	0.5
Thin	0.6	0.8	0.6	0.3	0.1	0.5
Medium	0.3	0.6	0.8	0.6	0.3	0.5
Fat	0.1	0.3	0.6	0.8	0.6	0.5
Very Fat	0	0.1	0.3	0.6	0.8	0.5
NA	0.5	0.5	0.5	0.5	0.5	0.5

We have created this table through several interviews with crime analysts in a city of China.

The range of similarity is 0 to 1. These similarity could be changed by experts during operation. Since the similarity of categorical attribute has been solicited from experts, there are no coefficients to learn from data. The similarity of categorical attribute can be expressed as:

$$Sim_{cat}^{k}\left(r_i^k, r_j^k\right) = table^k\left(loc\left(r_i^k\right), loc\left(r_j^k\right)\right) \tag{4}$$

The left side of the equation denotes the similarity between Cases *i* and *j* on categorical Attribute *k*, and the right side of the equation denotes the value obtained from the corresponding position in the table.

3.3.2. The similarity of quantitative attributes

For a quantitative attribute, we can use the absolute difference between two attribute values to denote their distance, i.e. $Dist_{qua}^{k}\left(r_i^k, r_j^k\right) = \left|r_i^k - r_j^k\right|$. The similarity between two attribute values can be expressed as a function of distance. When any of these two attribute values misses, the similarity would be set as 0.5.

$$Sim_{qua}^{k}\left(r_i^k, r_j^k\right) = QSF\left(\left|r_i^k - r_j^k\right|\right) \tag{5}$$

QSF is a function to make the range of similarity be 0 to 1. It could be Eq. (6) or other expressions.

$$QSF\left(x\right) = \left(max - x\right) / \left(max - min\right) \tag{6}$$

Here max and min are the maximum and the minimum of all the possible *x*, respectively. Care needs to be taken in the calculation of time and location’s similarity. Generally speaking, if two cases belong to a series, the temporal-spatial relationship between them should meet a special constraint, i.e. the time lag between them multiplied by the movement speed of offenders should be equal or greater than the distance between them. If two cases do not meet this condition, the similarity between them would be decreased to the minimum.

3.3.3. The similarity of hierarchical attributes

A hierarchical attribute is a special categorical attribute which has the hierarchical characteristic. First, one case may have multiple values in a hierarchical attribute. For example, the suspect may conduct more than one action in a criminal case, like threatening, beating the victim and snatching a property from the victim in the same case. Secondly, values in a hierarchical attribute may have the hierarchical relationship. Taking the locality of China as an example, a set of villages comprise a town, some towns, sub-districts, and counties form a city and some cities form a province. In the robbery incidents, all the modus operandi features can be designed as hierarchical attributes after discussing with experienced crime analysts.

We design a hierarchical similarity algorithm according to this characteristic. For example, Villages A and B belong to a town while Village C is in another town. Generally speaking, the similarity of locality between Villages A and B is often greater than that between Villages A and C.

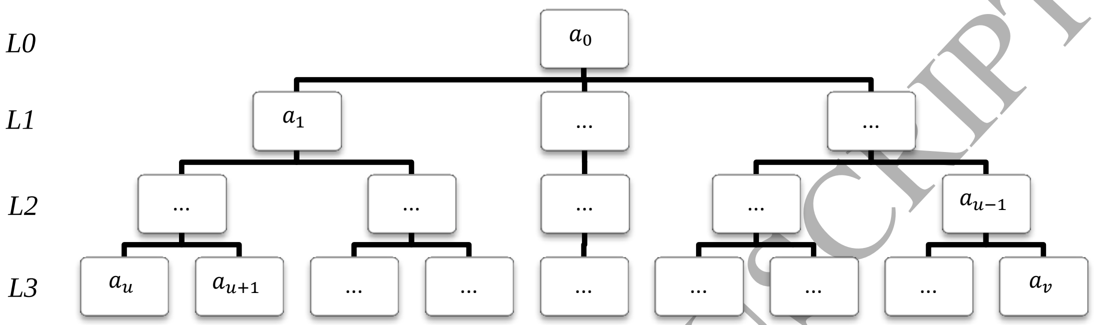


Fig. 4. Hierarchy trees

Given a set $A=\{a_u, a_{u+1}, \cdots, a_v\}$ as the range of a hierarchical attribute, the elements in set *A* could be gradually aggregated by a hierarchical tree structure. As shown in Fig. 4, $\{a_u, a_{u+1}, \cdots, a_v\}$ are leaf nodes of the tree, i.e. nodes of the lowest level *L3*, and $\{a_0, a_1, \cdots, a_{u-1}\}$ are nodes of levels *L0* to *L2*.

We set the roll-up rule in the tree structure as OR operation, i.e., the value of a parent node is equal to the logical addition of values of all of its child nodes. According to this rule, if we know the vector of leaf nodes, the vector of all nodes in the tree can be calculated accordingly. Thus, known a vector of leaf nodes $V=(v_u, v_{u+1}, \cdots, v_v)$, the hierarchical tree structure and the roll-up rule, a vector $V'=(v_0, v_1, \cdots, v_{u-1}, v_u, v_{u+1}, \cdots v_v)$ of all nodes in the tree can be obtained, where $v_l \in \{0,1\}$. Hence, we can use a binary vector V to denote a hierarchical attribute of a case, and V' contains hierarchy information of that attribute.

The values of Case *i* in a hierarchical Attribute *k*, r_i^k, is a subset of $\{a_u, a_{u+1}, \cdots, a_v\}$, so we can obtain the corresponding vector $\left(v_{i,u}^k, v_{i,u+1}^k, \cdots, v_{i,v}^k\right)$. Then we transform r_i^k to a vector $\left(v_{i,0}^k, v_{i,1}^k, \cdots, v_{i,u-1}^k, v_{i,u}^k, v_{i,u+1}^k, \cdots, v_{i,v}^k\right)$ which includes hierarchy information. The hierarchical distance between r_i^k and r_j^k could be defined as $Dist_{hie}^k\left(r_i^k, r_j^k\right)=\sum_{l=1}^{v}\omega_l^k \times \left|v_{i,l}^k - v_{j,l}^k\right|$. The hierarchical similarity algorithm can be expressed as:

$$Sim_{hie}^{k}\left(r_i^k, r_j^k\right) = HSF\left(\sum_{l=1}^{v}\omega_l^k \times \left|v_{i,l}^k - v_{j,l}^k\right|\right) \tag{7}$$

where *HSF* is a conversion function like *QSF*. When any of r_i^k and r_j^k is empty, which represents the missing data, the similarity would be set as 0.5. In this similarity, the weight ω_l^k are the parameters required to be determined. They could be given by experts. However, it is more reasonable to learn the parameters from the dataset if the data support the learning. Generally speaking, all the attribute values of all the hierarchical attributes are more than 100 while the solved robbery cases of one year in a city of china are relatively less. We need to reduce the parameters. So according to the characteristic of the tree structure, we assume that the weights of sibling nodes are identical.

We compare the hierarchical similarity in this paper with the commonly-used hierarchical similarity in literature, the taxonomic similarity. A regular hierarchical tree which has *L* levels is taken as an example, given that every non-leaf nodes have *D* child nodes, the tree has D^L leaf nodes.

When all the weights ω_l^k are equal to 1 and only one leaf node is equal to 1, the hierarchical distance proposed in this paper is the same as the taxonomic distance. When all the weights ω_l^k are equal to 1 and more than one leaf node are equal to 1, the values of our distances are a little different from the taxonomic distances. The reason is that we use the logical addition operation in the process of roll-up.

Commonly, the weights ω_l^k in our algorithm are not equal to 1. As a result, the values of distance calculation by our algorithm are different from the taxonomic algorithm. Some comparisons are listed as follows.

A. Time Complexity of similarity algorithm

In the processing stage we convert the vector of leaf nodes into the vector of all nodes in the tree that contains the structure information, so the time complexity of our algorithm is lower than the taxonomic algorithm. The time complexity of the taxonomic algorithm is proportional to the level number multiplied by the number of leaf nodes, i.e., $O\left(L*D^L\right)$. The time complexity of our hierarchical similarity algorithm is proportional to the number of all nodes in the tree and can be expressed as $O\left(\left(D^L - D\right)/\left(D-1\right)\right)$, which could be simplified as $O\left(D^{L-1}\right)$. It is clear that the complexity of our algorithm is lower than that of the taxonomic algorithm by one order of magnitude of *D*.

B. Discriminant capacity of similarity algorithm

According to the definition proposed by Golfarelli's et al., the discriminant capacity of the distance function is bounded by the number of distinct distances and the distribution of such values [16]. Compared to the taxonomic distance, the weights of different nodes in our distance algorithm could be different. The number of distinct distances of taxonomic distance is *L+1*, and the number of our distance is up to $\left(D^L * \left(D^L - 1\right)/2\right)$. So the discriminant capacity of our hierarchical similarity is greater than that of taxonomic similarity.

3.4. Classification model

According to the characteristic of our research problem, we could use a three-layer similarity network model to represent it (see Fig. 5). The input units are the absolute differences between two cases' attribute vectors. The middle units are the similarity of each attribute. The output unit is the similarity of a case pair, and if we set a decision similarity threshold, the output could be converted to the class of case pair (linked or unlinked). Different from the common neuron network's structure, the input units are partly connected with middle units in our network instead of fully connected. To be specific, every input unit (an element of the attribute vector) is only connected to the corresponding middle unit (similarity). This network structure conforms to this real-world problem and has a strong interpretability, because every layer has a corresponding physical meaning. With the help of the model, users not only know whether a case pair is linked or not but also know the reasons, i.e., some similar features of this pair.

Let $\left(\left(X_{11}, X_{12}, \cdots, X_{1v}\right), \left(X_{21}, X_{22}, \cdots, X_{2v}\right), \ldots, \left(X_{m1}, X_{m2}, \cdots, X_{mv}\right)\right)$ be the input vector and $\left(S_1, S_2, \cdots, S_m\right)$ be the middle vector. The connection parameter vector between the input layer and the middle layer is $\left(\left(\omega_1^1, \omega_2^1, \cdots, \omega_{np_1}^1\right), \left(\omega_1^2, \omega_2^2, \cdots, \omega_{np_2}^2\right), \cdots, \left(\omega_1^m, \omega_2^m, \cdots, \omega_{np_m}^m\right)\right)$ and the connection parameter vector between the middle layer and the output layer is $\left(\theta^1, \theta^2, \cdots, \theta^m\right)$. The sigmoid function is used to restrict the range of output. Thus, S_k equals $\frac{1}{1+e^{-\sum_{l=1}^{v}\omega_l^k \times X_{kl}}}$ and $\hat{Y}$ equals $\frac{1}{1+e^{-\sum_{k=1}^{m}\theta^k \times S_k}}$. The objective function of parameter learning is Eq. (8).

$$l(\theta,\omega)=\sum_{i=1}^{n}\left(Y_i\times\log\left(\hat{Y}_i\right)+\left(1-Y_i\right)\times\log\left(1-\hat{Y}_i\right)\right) \quad (8)$$

Then we could use the gradient descent algorithm to learn the optimal parameters, which would be introduced briefly in Section 5.2.

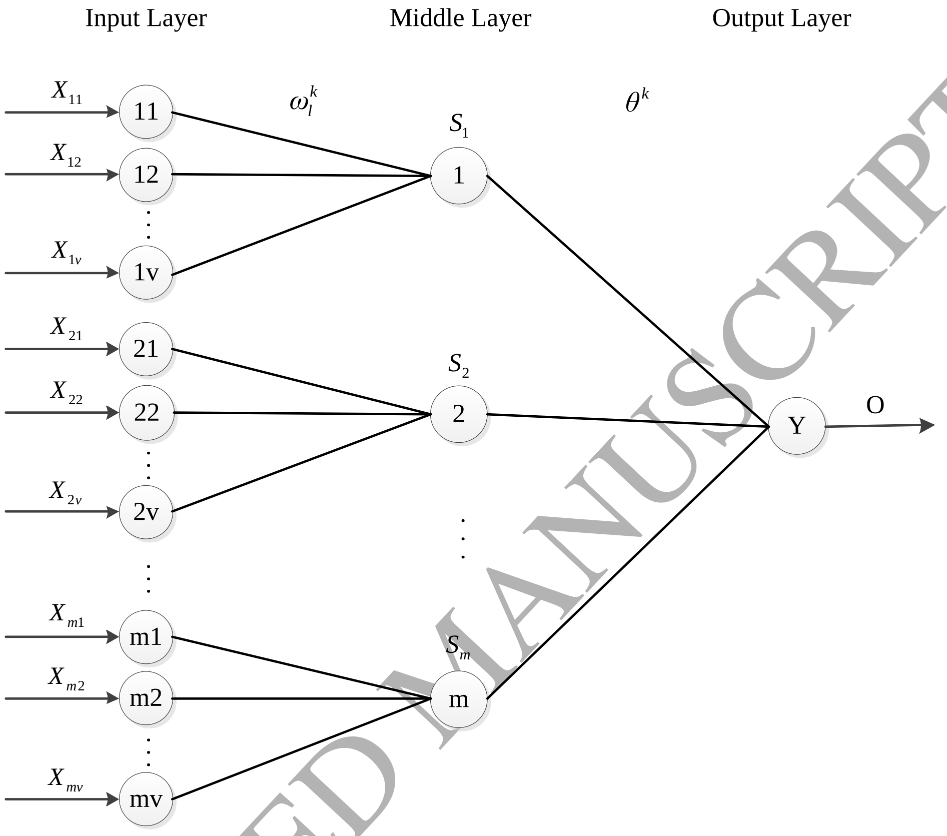


Fig. 5. A Three-Layer Similarity Network

4. Case Study

4.1. Data description

A real-world crime dataset is used to verify the performance of our proposed system as well as its underlying techniques. The dataset consists of robbery incidents occurred from January 2012 to May 2014. The incident information was collected by the police officers mainly through the checkbox-based form and the drop-down-box-based form (See Fig. 10 in Appendix A). The checkbox-based form collects modus operandi features and consists of 5 sections and about 100 checkboxes. In addition to the modus operandi features, information about time, date and locality etc. of the incident is also gathered. If required, a field for unstructured textual descriptions or observations also presents. This field allows police officers to enter additional information.

The dataset comprises 92 solved robbery incidents committed by 45 offenders (or gangs). Twelve offenders committed more than one incident and the remaining 33 offenders committed only one (see Table 3). So we can construct 4186 pairs of cases. There are 168 pairs committed by the same offender and 4018 pairs committed by different offenders.

Table 3

Case number of every series

Series	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	Other	Total
Case number	4	2	4	2	3	3	5	6	7	13	7	3	33	92

Every incident contains 22 features. Some features are denoted by categorical or quantitative value and some features are denoted by a vector. Categorical attributes include number of gang members, somatotype of the suspect, gender of the victim, etc. Quantitative attributes include height of the suspect, age of the suspect, time etc. Hierarchical attributes include actions taken by the suspects, tools used by the suspects, state of the victim before the crime was committed, etc. Every hierarchical attribute is denoted by a binary vector, where every element denotes an attribute value. Names and classification of these 22 features could be found in Table 12 in Appendix A.

4.2. Evaluation criteria of results

4.2.1. Measures of dataset's separability in features

For the binary classification problem, the separability of an input dataset denotes the separation degree of two classes of data samples. The separability reflects the complexity of a dataset itself and has an influence on the classification accuracy, regardless of which classifier we use [12]. Measures which describe the difference between two distributions can be used to describe the separability of an input dataset in a single feature. These measures focus on the effectiveness of a single feature in separating the classes. The measure can be used to choose input variables in the neural network model and be used to evaluate the effectiveness of a similarity algorithm as well. We list two measures as follows.

1) The two-sample Kolmogorov-Smirnov (KS) test value

The two-sample KS test is used to check whether there is a significant difference between two distributions [24, 41]. The KS test value is:

$$M1 = \max_x \left| F_{1,n_1}(x) - F_{2,n_2}(x) \right| \tag{9}$$

where F_{1,n_1} and F_{2,n_2} are the empirical distribution function of two classed respectively, and n_1 and n_2 are the sample size of the two samples respectively.

The null hypothesis of the KS test is that there is no significant difference between the two distributions. The null hypothesis is rejected at level α [39] if:

$$M1 > c(\alpha)\sqrt{\frac{n_1 + n_2}{n_1 \times n_2}} \tag{10}$$

M1 is used to describe the separability of a dataset. The range of the *M1* metric is 0 to 1. *M1=0* indicates that the distributions of two classes coincide completely; *M1=1* indicates that the distributions of two classes separate completely. The value of KS test reflects the separation degree of two classes of data samples.

2) The percentage of non-overlapped data samples

This is a measure describing how much each feature contributes to the separation of the two classes. This measure is defined as the fraction of all data points separable by a feature, namely, the percentage of data samples which are not in the overlap region [12]. The overlap region can be defined as follows:

$$R = \left[\max\left(\min_i\left(f_i^1\right), \min_i\left(f_i^2\right) \right), \min\left(\max_i\left(f_i^1\right), \max_i\left(f_i^2\right) \right) \right] \tag{11}$$

$$\operatorname{sgn}\left(f_i^c\right) = \begin{cases} 1, & f_i^c \notin R \\ 0, & others \end{cases} \tag{12}$$

$$M2 = \frac{\sum_{c=1}^{2} \sum_{i}^{n_c} \operatorname{sgn}\left(f_i^c\right)}{n_1 + n_2} \tag{13}$$

where $\min_i\left(f_i^c\right)$ and $\max_i\left(f_i^c\right)$ are the minimum and maximum values of a given feature in class *c*, and *c*=1, or 2 for the binary problem. For a data sample *i*, if f_i^c does not belong to *R*, $\operatorname{sgn}\left(f_i^c\right)$ is equal to 1. So *M2* denotes the percentage of non-overlapped data points. The range of *M2* is 0 to 1. Small *M2* indicates high overlap and low separability.

Compared to *M1*, the *M2* measure is easy to calculate. However, it is sensitive to outliers.

4.2.2. Metrics for Evaluating Classifier Performance

We take the widely used confusion matrix and corresponding statistical indexes: precision and recall rate to measure classification accuracy or effectiveness [17]. As mentioned in Section 11, our model would predict a value $\hat{Y}_{ij}$ which represents the similarity of the case pair. It is a continuous value between 0 and 1. So, a threshold between 0 and 1 is needed and the case pair is predicted as the positive tuple (linked pair) above this threshold. Given the threshold, the predicted classes of all the data samples could be obtained and the confusion matrix could be generated. As listed in Table 4, *TP* refers to the positive tuples that were correctly labeled by the classifier. *TN* indicates the negative tuples that were correctly labeled by the classifier. *FP* refers to the negative tuples that were incorrectly labeled as positive. *FN* indicates the positive tuples that were mislabeled as negative.

Table 4

Confusion Matrix

predicted actual	No	Yes	Total
No	*TN*	*FP*	*N*
Yes	*FN*	*TP*	*P*
Total	*N'*	*P'*	*P+N*

Precision can be thought of as a measure of exactness (i.e., what percentage of tuples labeled as positive are actually such), whereas recall is a measure of completeness (what percentage of positive tuples are labeled as such).

These measures can be computed as:

$$precision = TP/(TP+FP) \tag{14}$$

$$recall = TP/(TP+FN) \tag{15}$$

The accuracy of the classifier is:

$$accuracy = (TP+TN)/(P+N) \tag{16}$$

5. Case Study Results and Discussion

In this section, in order to assess the efficiency and effectiveness of our approach, we present three groups of results and make discussions about the results of our case study on a real world robbery dataset.

5.1. Comparisons of taxonomic and hierarchical similarity

In order to evaluate the effectiveness of our hierarchical similarity algorithm, we conduct an experiment to compare our hierarchical similarity algorithm with the commonly-used taxonomic similarity algorithm. The indicators used to compare are the above-mentioned separability indexes. The following table shows the comparison results.

Table 5

Comparisons of results' separability of taxonomic and hierarchical similarity

	M_Action		M_Tool		M_State		M_Prop		M_Result	
	M1	*M2*	*M1*	*M2*	*M1*	*M2*	*M1*	*M2*	*M1*	*M2*
taxonomic	0.65	0.01	0.53	0.01	0.65	0.38	0.69	0.00	0.30	0.09
HS1	0.66	0.12	0.55	0.19	0.65	0.20	0.70	0.00	0.38	0.05
HS2	0.72	0.20	0.57	0.43	0.65	0.34	0.70	0.00	0.38	0.09

In Table 5, *M1* and *M2* are the above-mentioned separability indexes. Taxonomic represents the calculation results of the taxonomic algorithm; HS1 represents the calculation results of hierarchical similarity algorithm when all the weights equal 1 and HS2 represents the calculation results of hierarchical similarity algorithm when the weights are optimized. M_Action etc. respectively represent five hierarchical attributes and they are the modus operandi features.

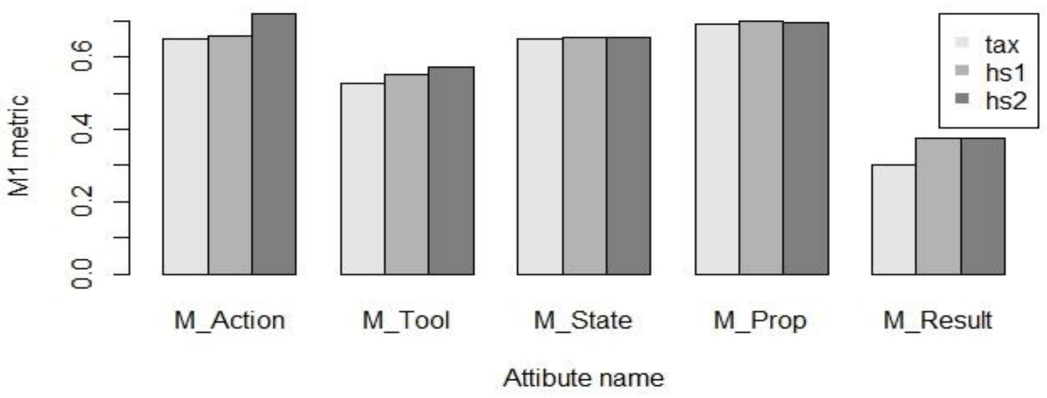


Fig. 6. Bar charts of *M1* Metric

As shown in Fig. 6, apart from attribute M_Result, the *M1* measure of the results of taxonomic and HS1 are almost identical, which illustrates that the effectiveness of the taxonomic similarity algorithm and our hierarchical similarity algorithm when all the weights equal 1 are almost identical. Compared to taxonomic, HS2 has a higher *M1* measure in three attributes:

M_Action, M_Tool and M_Result. This result illustrates that our hierarchical similarity algorithm is more effective after optimizing. The results above indicate that our similarity algorithm is no worse than taxonomic similarity algorithm in addressing the crime linkage problem.

In order to further compare HS2 with the taxonomic algorithm, we display distribution statistics and boxplots of similarities in

Table ***6***.

Table 6

Descriptive statistics for taxonomic and HS2 (Take M_Action for example)

	Unlinked		Linked	
	taxonomic	HS2	taxonomic	HS2
Min	0.000	0.000	0.333	0.000
Median	0.667	0.004	1.000	1.000
Mean	0.717	0.198	0.943	0.875
Max	1.000	1.000	1.000	1.000
SD	0.200	0.383	0.166	0.300

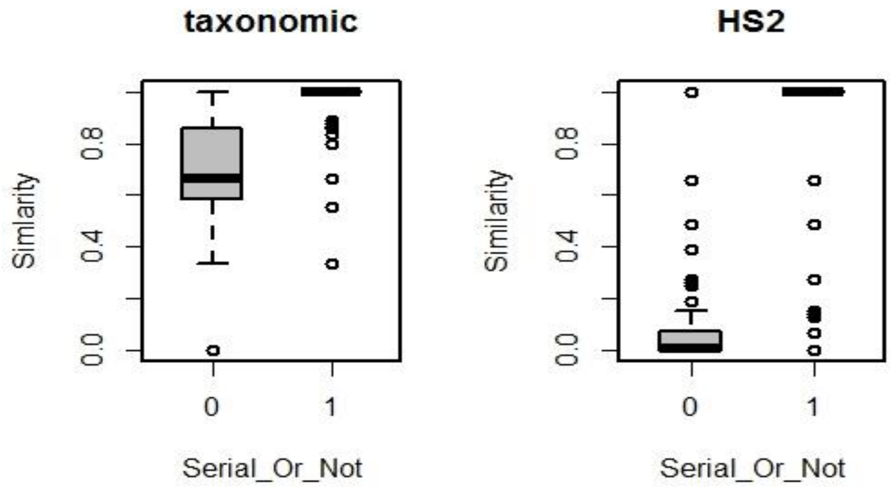


Fig. 7. Boxplot for the results of taxonomic and HS2

As shown in the first figure (the results of taxonomic) of Fig. 7, a large proportion (about 80%) of linked pairs' similarities are close to 1 and a large proportion (about 70%) of unlinked pairs' similarities are under 0.85. It is obvious that there is a significant difference between two classes of data samples. But some linked pairs' similarities are near to 0.3, which results in that the overlap between two classes is high and *M2* index is low. This result illustrates that *M2* index is sensitive to the outliers. Therefore, *M1* is a better index in measuring data separability than *M2*.

Similarly, we can draw the same conclusion from the right-hand figure (the results of HS2) of Fig. 7.

Comparing the similarities' distribution calculated by the taxonomic algorithm and HS2 algorithm, we find that the linked pairs' similarities calculated by two algorithms almost have no difference, while the unlinked pairs' similarities calculated by the HS2 algorithm are significantly lower than those calculated by the taxonomic algorithm. This result, as well as all the *M1* index, illustrate that the similarities calculated by HS2 algorithm has better separability and HS2 algorithm have more contribution to case linkage than the taxonomic algorithm.

5.2. Results of classification with 5 operandi features

In practice, crime analysts mainly link the serial cases using the similarities of the modus operandi features and the relationship of the temporal and spatial features among cases. And the studies in literature also show the importance of modus operandi features for the case linkage. In this paper, modus operandi features are hierarchical. In the previous section we analyze the contribution of each modus operandi feature to the case linkage after using the hierarchical similarity algorithm. However, it is not enough to link serial cases using a single modus operandi feature. The classification model introduced in Section 3.4 is a case pairs' classifier combining multiple modus operandi features. The input vector of the network, $\left(\left(X_{11},X_{12},\cdots,X_{1v}\right),\left(X_{21},X_{22},\cdots,X_{2v}\right),\ldots,\left(X_{m1},X_{m2},\cdots,X_{mv}\right)\right)$, is the difference vector of hierarchical features between cases, and the output *Y* is a flag of whether the case pair is linked.

Before parameter learning, we analyze the influence of every input component *X* to the output *Y*, and select some useful input components according to the data separability index *M1*. That is, for each component *X*, we divide it into two groups according to the value of the corresponding *Y*, then calculate the KS test value. If the KS test value is less than a certain threshold value, which indicates that the component has little contribution to the classification of *Y*, then we delete it. After removing non-significant input nodes and the corresponding input data, we can obtain the post-pruning network and corresponding dataset.

Such pre-pruning offers advantages in terms of simpler networks, faster training and better generalization ability to avoid overfitting due to the oversized network. Furthermore, pruning a network by removing insignificant input nodes may increase the predictive accuracy. The flowchart of our algorithm is listed as below.

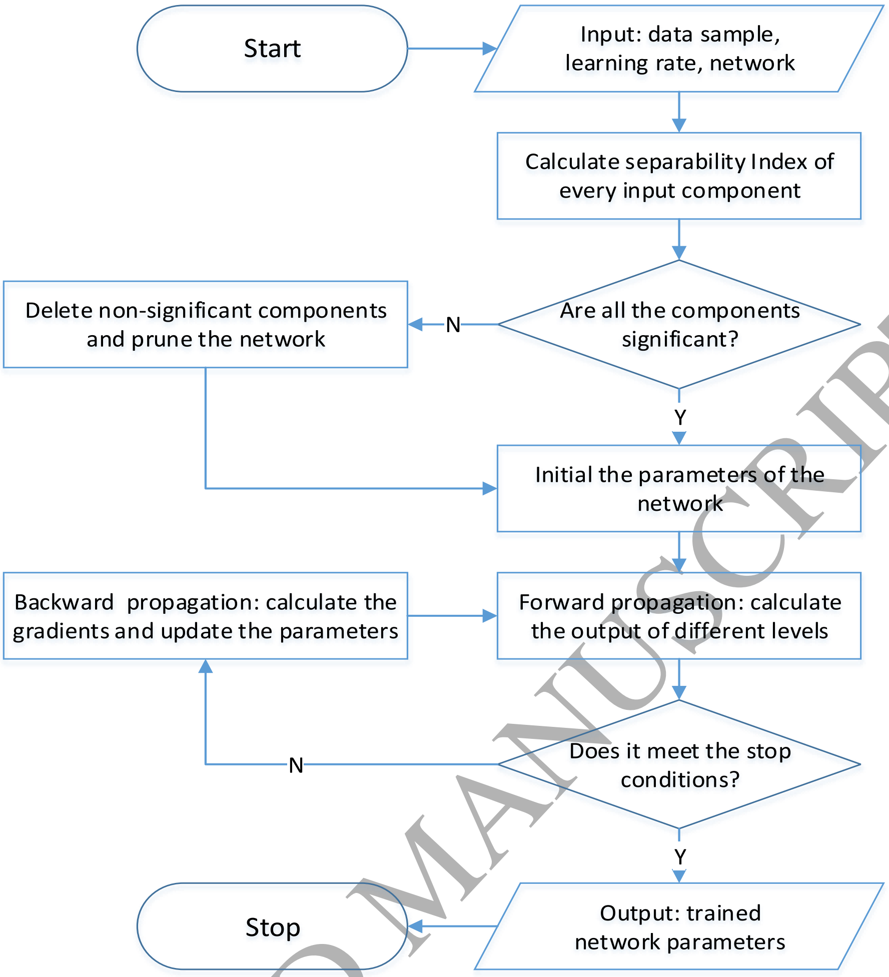

Start
Input: data sample, learning rate, network
Calculate separability Index of every input component
Are all the components significant?
N
Delete non-significant components and prune the network
Y
Initial the parameters of the network
Forward propagation: calculate the output of different levels
Backward propagation: calculate the gradients and update the parameters
Does it meet the stop conditions?
N
Y
Output: trained network parameters
Stop

Fig. 8. Flow chart of the pre-pruning and backpropagation algorithm

We conduct experiments using the data introduced in Section 4. Before pruning, there are 47 components in the input vector of the network. After filtering out some non-significant components, it only remains 28 components in the input vector. We compare the effectiveness and efficiency of these two classification model. As a comparison group, the results of the expert weighting model are also displayed. In the experts weighting model, weights of all the features are given by the experienced crime analysts instead of being learned from the data.

Table 7

The goodness of fit of the three model by comparing separability index

	Similarity network model (before pruning)	Similarity network model (after pruning)	Experts weighting model
M1	0.89	0.88	0.72
M2	0.82	0.80	0.48

Table 8

The goodness of fit of the three model by comparing the confusion matrix

Similarity network model (before pruning)			Similarity network model (after pruning)			Experts weighting model		
predicted actual	No	Yes	predicted actual	No	Yes	predicted actual	No	Yes
No	3913	105	No	3913	105	No	3923	95
Yes	46	122	Yes	50	118	Yes	83	85
precision=0.54; *recall*=0.73			*precision*=0.53; *recall*=0.71			*precision*=0.47; *recall*=0.51		

As shown in the above table, the separability index, precision and recall of the classifier (after pruning) are not significantly different from those of classifier (before pruning), which indicates that our feature selection method is reasonable and the pruning does not filter out too much useful information. Besides, compared with the comparison group, the separability index, precision and recall of the first two models are all higher than those of experts weighting model, which indicates our model significantly improve the effectiveness of linking serial cases.

As for the efficiency, the parameter learning speed of the after-pruning model is faster than that of before-pruning model because the 19 components are removed and the numbers of network connections are reduced correspondingly. Likewise, the calculation speed of the model is boosted after pruning when we apply the model to classify the case pairs.

In order to test the generalization ability of the classifier, a 4-folds cross-validation of the classifier has been done with results in Table 9.

Table 9

4-folds cross-validation results of the similarity network classifier

	test1	test2	test3	test4	average
M1	0.86	0.83	0.85	0.88	0.86
precision	0.45	0.49	0.53	0.49	0.49
recall	0.60	0.64	0.69	0.50	0.61

As shown in Table 9, the average *M1*, precision and recall of the 4-folds cross validation are 0.86, 0.49 and 0.61 respectively, while these indexes of fitting results are 0.88, 0.53 and 0.71. These accuracy indexes of cross validation results are only within 15% less than those of the fitting results, which indicates that the classifier has a good generalization ability. All of the above results show that the overall effect of the pruning classifier is not bad.

Further analyzing the confusion matrix of the fitting results of the network model, we can find that the number of FP, the negative samples that were incorrectly labeled as positive, in the matrix is relatively large, which results in the precision of the model is not high. In order to explain this outcome, we analyze the raw data of the dataset. There are 93 pairs in which all the modus operandi features are the same between the cases, that is, the input vector of the sample is zero vector. As a result, no matter what model we use, these pairs would be classified to the same class. These pairs consist of 43 linked pairs (positive tuples) and 50 unlinked pairs (negative tuples). This characteristic of dataset inevitably lead to that the FP of classification results is greater than 50 when we only use the 5 modus operandi features. Thus, if we want to improve the precision of the model, we need to add other features into the model.

5.3. Results of classification with all the features

Based on the analysis of former section, in order to further improve the accuracy of the classifier, we use the hierarchical modus operandi features, combined with the similarities of time, locality and some other features as the input vector of the model. We build a new similarity network to classify the case pairs. Before the classification, we also calculate the *M1* index of all the input attributes and select the attributes with good separability. The selected attributes of the input vector are 5 modus operandi features which consist of 28 components and similarities of other features like number of gang members, gender of the victim, time, moment and locality.

The goodness of fit of our classifier is listed as below. Since our dataset is unbalanced, the accuracy of the model could be misleading. Compared with the confusion matrix in Section 5.2, the precision and recall of the classification results are all improved significantly.

Table 10
The goodness of fit of similarity network model

predicted / actual	No	Yes
No	3984	34
Yes	34	134
precision=0.80; *recall*=0.80; *accuracy*=0.983		

When we change the threshold, different precision and recall of the model could be obtained. As shown in Fig. 9, the horizontal axis represents the position of threshold in all the predicted similarities. For example, 0.95 means that we set the threshold at the top 5% position of all the similarities. The two lines denote the change of precision and recall respectively. In the application of our decision support system, we set a default threshold and output all the predicted positive data samples as possible linked case pairs. Considering the caseload, after discussing with police officers, we set the default threshold at the top 4% position of all the similarities. Meanwhile, the crime analysts could change the threshold in the system interface to meet their actual demand. If they want to dig out most of the linked pairs, they could set a small threshold (moving toward left on the horizontal axis). The recall would be large, but at the same time the precision would be small, which means that they need more workload to analyze these pairs, and vice versa. It a trade-off between precision and recall, or workload and the number of the true linked pairs.

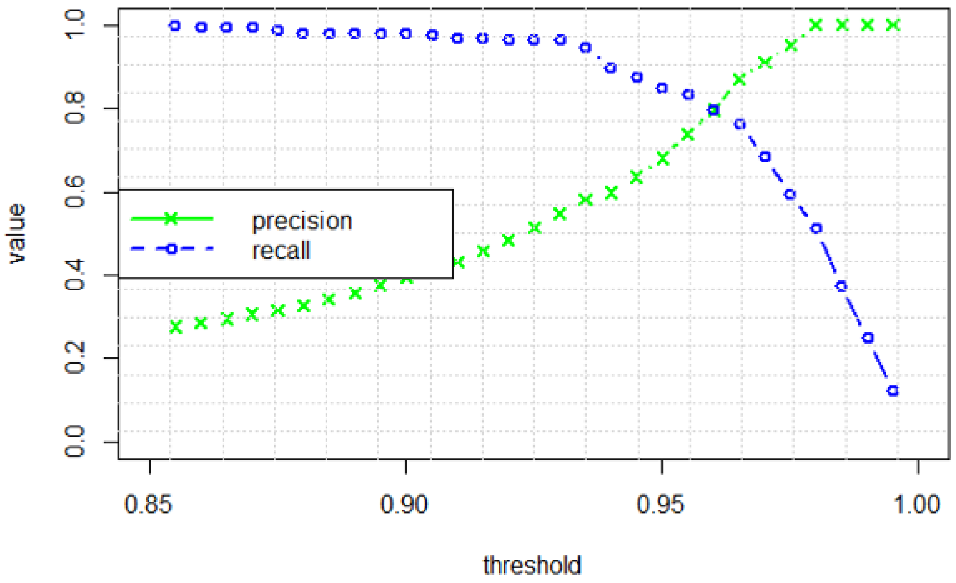


Fig. 9. The change of precision and recall with threshold

To test the generalization ability of the model, a 4-folds cross-validation of the classifier has also been done and the results is listed in Table 11.

Table 11
4-folds cross-validation results of the similarity network classifier

	test1	test2	test3	test4	average
M1	0.96	0.88	0.94	0.92	0.92
precision	0.83	0.69	0.74	0.79	0.76
recall	0.83	0.69	0.74	0.79	0.76
accuracy	0.99	0.98	0.98	0.98	0.981

Compared with the results of Section 5.2, the precision and recall of cross validation are all improved significantly, which indicate that the newly added attributes are effective in improving the accuracy of the model. Compared with the fitting results, the average precision, recall and accuracy of cross-validation are very close to those of the fitting results, which show that the generalization ability of the model is promising.

In conclusion, the above results imply that our model, together with the pre-pruning and backpropagation algorithm have a good efficiency and effectiveness in addressing the crime linkage problem. As for the case study on robbery case dataset, it is a good solution of the case pairs' classification problem by building the similarity network model with 5 modus operandi features and similarities of other features including of number of gang members, gender of the victim, time, moment and locality. When the training dataset change with time, the attributes used in our model would be adjusted automatically.

6. Conclusion

In this paper, a decision support system for detecting serial crimes has been presented. The system not only allow common users to link serial crimes automatically through its front-end modules, but also provide flexibility in training data update and domain experts' interaction, including adjusting the key components like similarity matrices, and decision thresholds to reach a good tradeoff between caseload and number of true linked pairs.

The underlying method framework of this system is also introduced. The framework consists of similarity algorithms, a classification model, a feature selection and parameter learning algorithm used to link the serial crimes. As for the similarity, we have designed similarity algorithms according to the characteristic of different types of features, especially the hierarchical characteristic of some modus operandi features. As for the classification model, since the parameters in the similarity calculation and the coefficients of different features are all required to specify, we have used a similarity network structure to formulate this model in which the two-phase parameters could be learned simultaneously. Besides, this structure is interpretable and easy to be understood by system users such as crime analysts. As for the parameter learning algorithm, we have first pruned the network structure according to the separability index of input attributes and then learned these parameters using the commonly-used gradient descent algorithm. We have not only aimed at making the classification accurately but also improved the efficiency of the method.

The theoretical analysis and the real-world case study results have demonstrated that our hierarchical similarity is superior to the taxonomic similarity algorithm in terms of both effectiveness and efficiency. The linkage results imply that our method can classify the case pairs at a high accuracy with good generalization ability and reduce the calculation complexity by pre-pruning as well. Above all, the theoretical analysis and real world case study have indicated that our method framework with default setting perform well on linking accuracy as well as interpretability and computational efficiency.

The system has been deployed in a public security bureau in China. After running for more than one year, it receives positive feedback from users with more than ten detected series of crimes having been confirmed. It allows law enforcement agency to save resources and improves the work efficiency of the individual police officers with an accuracy of crime linkage comparable with the human discriminant.

However, there are still some limitations of this study. First, our approach cannot directly deal with the textual data currently. Secondly, we only verify its validity in linking serial robbery cases. It is cannot be applied to other crime types without adaptation.

In the future, we will further improve the performance of the system according to feedback from the operation. One direction is to use text mining techniques like text classification [25, 26, 36] and topic modelling of sentences or short notes [14, 15]. Besides, despite the fact that it may deteriorate the interpretability of linkage results, we will explore some non-linear classification models as a possible alternative for better linkage accuracy.

Reference

[1] F. Albertetti, P. Cotofrei, L. Grossrieder, O. Ribaux, K. Stoffel, The CriLiM Methodology: Crime Linkage with a Fuzzy MCDM Approach, Intelligence and Security Informatics Conference, (2013) 67-74.

[2] K. Baumgartner, S. Ferrari, G. Palermo, Constructing Bayesian networks for criminal profiling from limited data, Knowledge-Based Systems, 21(7) (2008) 563-572.

[3] C. Bennell, S. Bloomfield, B. Snook, P. Taylor, C. Barnes, Linkage analysis in cases of serial burglary: comparing the performance of university students, police professionals, and a logistic regression model, Psychology, Crime & Law, 16(6) (2010) 507-524.

[4] C. Bennell, D.V. Canter, Linking commercial burglaries by modus operandi: tests using regression and ROC analysis, Science & Justice, 42(3) (2002) 153-164.

[5] C. Bennell, D. Gauthier, D. Gauthier, T. Melnyk, E. Musolino, The impact of data degradation and sample size on the performance of two similarity coefficients used in behavioural linkage analysis, Forensic science international, 199(1-3) (2010) 85-92.

[6] M. Boldt, A. Borg, Clustering residential burglaries using multiple heterogeneous variables, International Journal of Information Technology & Decision Making, (2015).

[7] A. Borg, M. Boldt, Clustering Residential Burglaries Using Modus Operandi and Spatiotemporal Information, International Journal of Information Technology & Decision Making, (2016) 1550033.

[8] A. Borg, M. Boldt, N. Lavesson, U. Melander, V. Boeva, Detecting serial residential burglaries using clustering, Expert Systems with Applications, 41(11) (2014) 5252-5266.

[9] S. Boriah, V. Chandola, V. Kumar, Similarity measures for categorical data: A comparative evaluation, Proceedings of the 2008 SIAM International Conference on Data Mining, (2008) 243-254.

[10] J.W. Brahan, K.P. Lam, H. Chan, W. Leung, AICAMS: artificial intelligence crime analysis and management system, Knowledge-Based Systems, 11(5) (1998) 355-361.

[11] D.E. Brown, S. Hagen, Data association methods with applications to law enforcement, Decision Support Systems, 34(4) (2003) 369-378.

[12] J.-R. Cano, Analysis of data complexity measures for classification, Expert Systems with Applications, 40(12) (2013) 4820-4831.

[13] J. de Zoete, M. Sjerps, D. Lagnado, N. Fenton, Modelling crime linkage with Bayesian networks, Science & Justice, (2014).

[14] L. Du, W. Buntine, H. Jin, A segmented topic model based on the two-parameter Poisson-Dirichlet process, Machine Learning, 81(1) (2010) 5-19.

[15] L. Du, W. Buntine, H. Jin, C. Chen, Sequential latent Dirichlet allocation, Knowl Inf Syst, 31(3) (2012) 475-503.

[16] M. Golfarelli, E. Turricchia, A characterization of hierarchical computable distance functions for data warehouse systems, Decision Support Systems, 62(2014) 144-157.

[17] J. Han, M. Kamber, J. Pei, Data mining: concepts and techniques, (Elsevier, 2011).

[18] N. Han, W. Chen, Case Linkage based on Cluster Analysis, Journal of Chinese People's Public Security University (Scinece and Technology), 71(1) (2012) 53-58.

[19] H. Hao, Scientific Apply of the Condition for Investigating Merged Cases, Journal of Chinese People's Public Security University (Social Science Edition), 22(4) (2006).

[20] Z. Hubalek, Coefficients of Association and Similarity, Based on Binary (Presence Absence) Data

- an Evaluation, Biol Rev, 57(Nov) (1982) 669-689.

[21] D.J. Icove, Automated crime profiling, FBI Law Enforcement Bulletin, 55(12) (1986) 27-30.

[22] C. Izsak, A.R.G. Price, Measuring beta-diversity using a taxonomic similarity index, and its relation to spatial scale, Mar Ecol Prog Ser, 215(2001) 69-77.

[23] G. Johnson, VICLAS: Violent crime linkage analysis system, RCMP Gazette, 56(10) (1994) 9-13.

[24] A. Justel, D. Pena, R. Zamar, A multivariate Kolmogorov-Smirnov test of goodness of fit, Stat Probabil Lett, 35(3) (1997) 251-259.

[25] N. Kalchbrenner, E. Grefenstette, P. Blunsom, A convolutional neural network for modelling sentences, arXiv preprint arXiv:1404.2188, (2014).

[26] Y. Kim, Convolutional neural networks for sentence classification, arXiv preprint arXiv:1408.5882, (2014).

[27] P.W. Kwan, M.C. Welch, J.J. Foley, A knowledge-based Decision Support System for adaptive fingerprint identification that uses relevance feedback, Knowledge-Based Systems, 73(2015) 236-253.

[28] D. Lin, An Information-Theoretic Definition of Similarity, International Conference on Machine Learning, (1998).

[29] S. Lin, D.E. Brown, An outlier-based data association method for linking criminal incidents, Decision Support Systems, 41(3) (2006) 604-615.

[30] Y. Liu, G. Zhao, Priciples and Mehods of Investigating merge cases, (China University of Political science and Law Press, 2013).

[31] L. Ma, Y. Chen, H. Huang, AK-Modes: A weighted clustering algorithm for finding similar case subsets, in: Intelligent Systems and Knowledge Engineering (ISKE), 2010 International Conference on, (IEEE, 2010), pp. 218-223.

[32] L. Markson, J. Woodhams, J.W. Bond, Linking serial residential burglary: Comparing the utility of modus operandi behaviours, geographical proximity, and temporal proximity, Journal of Investigative Psychology and Offender Profiling, 7(2) (2010) 91-107.

[33] M.M. Martineau, S. Corey, Investigating the Reliability of the Violent Crime Linkage Analysis System (ViCLAS) Crime Report, Journal of Police and Criminal Psychology, 23(2) (2008) 51-60.

[34] T. Melnyk, C. Bennell, D.J. Gauthier, D. Gauthier, Another look at across-crime similarity coefficients for use in behavioural linkage analysis: an attempt to replicate Woodhams, Grant, and Price (2007), Psychology, Crime & Law, 17(4) (2011) 359-380.

[35] G. Oatley, B. Ewart, J. Zeleznikow, Decision support systems for police: Lessons from the application of data mining techniques to “soft” forensic evidence, Artificial Intelligence and Law, 14(1-2) (2006) 35-100.

[36] G. Pang, H. Jin, S. Jiang, CenKNN: a scalable and effective text classifier, Data Min Knowl Disc, 29(3) (2015) 593-625.

[37] M.D. Porter, A Statistical Approach to Crime Linkage, The American Statistician, (2015) 1-38.

[38] B.J. Reich, M.D. Porter, Partially supervised spatiotemporal clustering for burglary crime series identification, Journal of the Royal Statistical Society: Series A (Statistics in Society), 178(2) (2015) 465-480.

[39] N. Smirnov, Table for Estimating the Goodness of Fit of Empirical Distributions, Ann Math Stat, 19(2) (1948) 279-279.

[40] B. Snook, K. Luther, J.C. House, C. Bennell, P.J. Taylor, The Violent Crime Linkage Analysis System: A Test of Interrater Reliability, Criminal Justice and Behavior, 39(5) (2012) 607-619.

[41] M.A. Stephens, Edf Statistics for Goodness of Fit and Some Comparisons, J Am Stat Assoc,

69(347) (1974) 730-737.

[42] P.E. Taylor, S.J. Huxley, A break from tradition for the San Francisco police: Patrol officer scheduling using an optimization-based decision support system, Interfaces, 19(1) (1989) 4-24.

[43] M. Tonkin, J. Woodhams, R. Bull, J.W. Bond, Behavioural case linkage with solved and unsolved crimes, Forensic science international, 222(1-3) (2012) 146-153.

[44] M. Tonkin, J. Woodhams, R. Bull, J.W. Bond, E.J. Palmer, Linking Different Types of Crime Using Geographical and Temporal Proximity, Criminal Justice and Behavior, 38(11) (2011) 1069-1088.

[45] T. Wang, C. Rudin, D. Wagner, R. Sevieri, Detecting Patterns of Crime with Series Finder, in: AAAI (Late-Breaking Developments), (2013).

[46] T. Wang, C. Rudin, D. Wagner, R. Sevieri, Learning to detect patterns of crime, in: Machine Learning and Knowledge Discovery in Databases, (Springer, 2013), pp. 515-530.

[47] T. Wang, C. Rudin, D. Wagner, R. Sevieri, Finding Patterns with a Rotten Core: Data Mining for Crime Series with Cores, Big Data, 3(1) (2015) 3-21.

[48] J. Woodhams, R. Bull, C.R. Hollin, Chapter 6: Case linkage: identifying crimes committed by the same offender. In: Criminal Profiling: International Theory, Research, and Practice., (New Nork: Humana Press Inc., 2007).

[49] J. Woodhams, T.D. Grant, A.R.G. Price, From marine ecology to crime analysis: Improving the detection of serial sexual offences using a taxonomic similarity measure, Journal of Investigative Psychology and Offender Profiling, 4(1) (2007) 17-27.

[50] J. Woodhams, C.R. Hollin, R. Bull, The psychology of linking crimes: A review of the evidence, Legal and Criminological Psychology, 12(2007) 233-249.

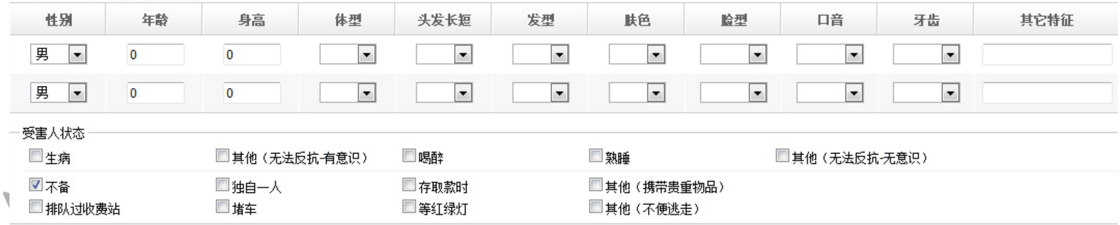
[51] J. Woodhams, G. Labuschagne, A Test of Case Linkage Principles with Solved and Unsolved Serial Rapes, Journal of Police and Criminal Psychology, 27(1) (2011) 85-98.

[52] J. Woodhams, K. Toye, An empirical test of the assumptions of case linkage and offender profiling with serial commercial robberies, Psychology, Public Policy, and Law, 13(1) (2007) 59-85.

Appendix A

Table 12
Attributes' description and type

No.	Attribute	Attribute Description	Type
1	Gan_Num	Number of gang members	categorical
2	Sus_Gen	Gender of the suspect	categorical
3	Time	Time of the crime	numeric
4	M_Action	Actions taken by the suspects	hierarchical
5	M_Tool	Tools used by the suspects	hierarchical
6	M_State	State of the victim before the crime was committed	hierarchical
7	M_Prop	Properties robbed by the suspects	hierarchical
8	M_Result	Result of the crime	hierarchical
9	Wek	Whether the crime happened at weekends	categorical
10	Mom	Moment of the crime	numeric
11	Vic_Gen	Gender of the victim	categorical
12	M_Esc	Method of escaping the scene	categorical
13	Loc	Location of the crime	categorical
14	Vic_Age	Age of the victim	numeric
15	Sus_Age	Age of the suspect	numeric
16	Sus_Wei	Somatotype of the suspect	categorical
17	Sus_Sta	Body stature of the suspect	numeric
18	Sus_Len	Hair length of the suspect	categorical
19	Sus_Acc	Accent of the suspect	categorical
20	Sus_Fac	Face shape of the suspect	categorical
21	Sus_Hair	Hair style of the suspect	categorical
22	Sus_Col	Skin color of the suspect	categorical



性别	年龄	身高	体型	头发长短	发型	肤色	脸型	口音	牙齿	其它特征
男	0	0								
男	0	0								

受害人状态

□生病 □其他（无法反抗-有意识） □喝醉 □熟睡 □其他（无法反抗-无意识）

☑不备 □独自一人 □存取款时 □其他（携带贵重物品）

□排队过收费站 □堵车 □等红绿灯 □其他（不便逃走）

Fig. 10. Example of the checkbox-based form and drop-down-box-based form in the system