Mining Popular Crime Patterns from Crime Datasets

[1] BVS. Varma, [2] V. Valli Kumari

[1] CSE Department, Ideal Institute of Technology
Kakinada, Andhra Pradesh, India

[2] CS&SE Department, A U College of Engineering, Andhra University
Visakhapatnam, Andhra Pradesh, India

Abstract - **Criminal investigation plays an important role in law enforcement agencies while analyzing crimes. This can help in finding suspects and for better attribution of past crimes. So the use of pattern based approaches will have the potential to assist crime analyze experts in discovering new patterns of criminal activities. So the research is extended in this area in finding new crime patterns. Since frequent pattern mining problem has been introduced, researchers have been developed many ways and also extended to different useful emerging patterns such as closed, maximal, cyclic, periodic, and popular patterns. In this paper, we introduced popular crime patterns which give the popularity of each crime or incident among the entire crime database. The process used for mining popular crime patterns is discussed as PCrime-growth algorithm. The compact data structure called PCrime-tree is implemented in this algorithm. Finally our experiment results have been shown which gives information of compact, space efficient and time efficient of our proposed algorithm.**

Keywords - *Crime Patterns, Popular Patterns, Crime databases, PCrime-growth, PCrime-tree.*

1. Introduction

Criminals commit crimes at any place and in any form. As the law-enforcement insists crime is an offence against the society that is often prosecuted and punishable. Data mining approaches such as pattern mining is an essential component of discovering various patterns in this domain of interest (crime). For the sake of public safety researchers are investigating various ways to handle crimes [1-6] and also improving on existing algorithms used to develop crime patterns. So many algorithms have been proposed after the introduction of frequent pattern mining. These proposed algorithms have been classified into two categories. The first category is mainly focused on algorithm efficiency, to avoid the candidate generation-and-test approach of the Apriori algorithm [7], a tree-based algorithm called FP-growth[8] was proposed to capture the content of transactional database so that frequent patterns can be mined recursively by using FP-tree [8] structure.

In the second category it is mainly focused on the extending work of frequent patterns to other interesting or useful metrics which are further used in research. These mining patterns are based on support/frequency calculations of the given data. Though these metrics are useful, the support-based frequent patterns are not sufficient to discover many interesting knowledge in transactional database among patterns. Due to this reason the problem of rising some interesting measures and their corresponding such as correlated patterns [9], periodic patterns [10,11], constrained patterns [12,13], hyperclique patterns [14], high utility patterns[15], popular patterns [16] [17] etc.

Law-enforcement may wants to find popular crime patterns from large group of criminals and are most popular in doing crime. Similarly criminals also want to get connected easily to a popular crime person or making association with a crime person. This search method can be done easily if the popular criminals are been grouped and placed in one popular crime pattern. Some well known crimes will happen frequently in almost all areas of the world such as robbery, rape, murder, dacoit, injuring, cyber crimes etc. All these crimes we see frequently in news from so long time. Such frequently happening crimes may affect not only the victims but also the public as a whole. Therefore, to check on crimes and target to the criminals are inevitable that need to be performed by the law enforcement officials to give protection for the public. Law enforcement agencies along with computer data analysts are responsible to be competent in crime investigation from the voluminous crime datasets. So, in this paper we propose an algorithm for extracting popular crime patterns in the betterment of the society by helping law enforcement agencies in reducing crimes.

The rest of the paper is organized as follows. In section 2 we define the problem of popular crime patterns in crime databases. We describe the process of mining popular crime patterns by PCrime-growth algorithm in section 3.

In section 4 we describe how to draw the pop-tree structure for mining popular crime patterns. In section 5 related work is discussed. Our experimental results have been shown in section 6. Finally the paper is been concluded in section 7.

2. Related Work

In the study and analysis of Data mining the criminology can be categorized into crime control and crime suppression. Crime control tends to use knowledge from the analyzed data to control and prevent the occurrence of crime, whereas the criminal suppression tries to catch a criminal by using the history of a person recorded in databases.

Brown et.al [19] developed an application framework called ReCAP (Regional Crime Analysis Program) for mining data in order to catch professional criminals using data fusion and data mining techniques.

The Data fusion technique was used to manage, fuse and interprets information from various sources. The main purpose was to overcome confusion from conflicting reports and cluttered or noisy backgrounds. Data mining was used to automatically discover patterns and relationships in large databases. Crime detection and prevention techniques are applied to different applications ranging from cross-border security, Internet security to household crimes.

De Bruin et. al. [20] introduced a framework for crime trends using a new distance measure for comparing all individuals based on their profiles and then clustering them accordingly.

This method also provided a visual clustering of criminal careers and identification of classes of criminals. Abraham et al. [21] proposed a method to employ computer log files as history data to search some relationships by using the frequency occurrence of incidents. Then, they analyzed the result to produce profiles, which can be used to perceive the behavior of criminal. From the literature study, it could be concluded that crime data is increasing to very large quantities running into zota bytes (1024bytes).

This in turn is increasing the need for advanced and efficient techniques for analysis. Data mining as an analysis and knowledge discovery tool has immense potential for crime data analysis.

As is the case with any other new technology, the requirement of such tool changes, which is further augmented by the new and advanced technologies used by criminals. All these facts confirm that the field is not yet mature and needs further investigations.

3. Problem Statement

In this section we describe the basic problem definitions related to popular crime patterns. Let us consider Table 1 as a typical crime database (*CDB*) [18] with categorical crime attributes which comprises incident location, suspect and victim information, day of the week, time and date, weapons used, and crime scene status etc., a few are mention below.

The first step is to manage and identify the attributes of interest that are available in the crime database, can see in Table 1. Then the database is queried to extract the attributes of interest, creating the Crime Transaction Database (*CTDB*) which is to be mined.

The mining phase takes place in the given crime database as follows, we consider a simple analogy from Table 1. Suppose we passed a query "involving location information and corresponding crime incidents that occurred there", to obtain the crime transaction database (*CTDB*) in Table 2, then it is further encoded or transformed with location as Crime Identification (*CrimeID*) being the unique identifier for each crime transaction and Crime Incidents that happen in that location on a particular day or a week etc.

Each crime incident and crimeID has been encoded with abbreviations in order to reduce processing time. *CTDB* is a set of transactions projected over *CDB*. Now, we can able to mine popular crime patterns from Table 2.

IJCSN International Journal of Computer Science and Network, Volume 4, Issue 5, October 2015
ISSN (Online) : 2277-5420 www.IJCSN.org
Impact Factor: 0.417

Table 1: A Crime Database (CDB)

Location Category	*Crime Incident*	*VictimID*	*Gender*	*Time*	*Culprit*	. . .
Area 1	Robbery	A2	M	Noon	gang	. . .
Area 2	Kidnapping	D4	M	Noon	gang	. . .
Area 3	Robbery	Z8	F	Night	individual	. . .
Area 4	Rape	P3	M	Night	gang	. . .
-	-	-	-	-	-	
-	-	-	-	-	-	
-	-	-	-	-	-	. . .

Table 2: Crime Transactional database (CTDB)

Crime Id	*Crime Incidents*
ct_1	c_2, c_4
ct_2	c_2, c_3, c_6, c_7, c_8
ct_3	$c_2, c_3, c_4, c_5, c_6, c_7$
ct_4	c_3, c_5, c_7, c_8
ct_5	c_1, c_4
ct_6	c_1, c_2, c_9
ct_7	c_1, c_4, c_5

Let Crime-item={c_1 , c_2 , c_3, ...c_m } be a set of *m* domain items. A set of *n* crime transactions in crime transactional database (CTDB) are {ct_1, ct_2, ct_3,......ct_n} where each crime transaction ct_j in CTDB is a subset of each Crime-item in crime transactional database. $|ct_j|$ is used to represent the length of crime transaction ct_j. Let C = {c_1 , c_2 , ...c_k} $\subseteq$ Crime-item or Crime-pattern consisting of k items (k-itemset), where $|C|=k \leq m$. The projected database of C denoted as DB_C is a set of CTDB transaction that contain C. Here *max*CTL(C) and *sum*CTL(C) are used to represent maximum crime length and total crime length of all crime transactions in DB_C. From the above description we draw the following definitions:

Crime Transaction Popularity, Pop(C, ct_j) of a pattern C in crime transaction ct_j measures the membership degree of C in ct_j. We compute the membership degree based on the difference between the crime transaction length $|ct_j|$ and pattern size $|C|$.

$$Pop(C, ct_j) = |ct_j| - |C| \quad \text{--- } 1$$

Long Crime Transaction Popularity, Pop(C, $ct_{maxCTL(C)}$) of a pattern C in crime transaction $ct_{maxCTL(C)}$ measures the membership degree of C in ct_{maxCTL}, where $ct_{maxCTL(C)}$ is the crime transaction having maximum length in DB_C .

$$Pop(C, ct_{maxCTL(C)}) = \left(\max_{ctj \in DB_C} |ctj|\right) - |C| \quad \text{--- } 2$$

Popularity, Pop(C) of a pattern C in the CTDB measures an aggregated membership degree of C in all crime transaction in

the CTDB. It is defined as an average of all crime transaction popularities of C.

$$Pop(C) = \frac{1}{|DB_C|} \sum_{ct_j \in DB_C} Pop(C, ct_j) \quad \text{--- } 3$$

Popular Crime A user specified minimum popularity threshold min_pop is given, a crime C is considered popular if its popularity is atleast min_pop (i.e $Pop(C) \geq$ minpop).

4. Mining Popular Crime Patterns

During the process of mining the frequent patterns, the calculation of frequency satisfies the downward closure property i.e., if one item is considered as infrequent then its superset is also considered as infrequent which helps in reducing the search space by deleting the infrequent patterns and thus the process of mining patterns will be faster. Sometimes when mining popular patterns we can observe that from Table 2, a crime pattern {c_2} is unpopular but its superset {c_2, c_3} can be popular which does not satisfy downward closure property. Therefore the mining of popular crime patterns may be a challenging step. We describe step by step procedure how the process of mining popular patterns takes place in this section. For mining popular crime patterns we are proposing a PCrime-growth algorithm which is having two phases like FP-growth algorithm. The two key phases are (a) construction of a PCrime-tree (b) mining of popular crime patterns from PCrime-tree.

IJCSN International Journal of Computer Science and Network, Volume 4, Issue 5, October 2015
ISSN (Online) : 2277-5420 www.IJCSN.org
Impact Factor: 0.417

Let us consider the CTDB shown in Table 2, which consists of n=7 crime transactions and m=9 domain items $c_1, c_2, \ldots, c_9$. For pattern $C = \{c_2, c_3\}$, its projected database $DB_{\{c2,c3\}}=\{ct_2, ct_3\}$. Hence, $|DB_{\{c2,c3\}}| = 2$, $|ct_2| = |\{c_2, c_3, c_6, c_7, c_8\}| = 5$, $|ct_3| = |\{c_2, c_3, c_4, c_5, c_6, c_8\}| = 6$, $maxCTL(\{c_2, c_3\}) = \max\{|ct_2|, |ct_3|\} = \max\{5, 6\} = 6$, and $sumCTL(\{c_2, c_3\}) = |ct_2| + |ct_3| = 5 + 6 = 11$.

The crime transaction popularity of pattern $\{c_2, c_3\}$ in ct_2 can be computed as $Pop(\{c_2, c_3\}, ct_2) = |ct_2|-|\{c_2, c_3\}| = 5-2 = 3$. Similarly, $Pop(\{c_2, c_3\}, ct_3) = |ct_3|-|\{c_2, c_3\}| = 6-2 = 4$. Recall from Example 1 that $DB_{\{c2,c3\}}=\{ct_2, ct_3\}$ (i.e., $\{c_2, c_3\}$ appears only in ct_2 and ct_3). As ct_3 is the longest crime transaction in $DB_{\{c2,c3\}}$ (because $maxCTL(\{c_2, c_3\}) = 6$), the long transaction popularity of crime pattern $\{c_2, c_3\}$ in $ct_{maxCTL(\{c2,c3\})}$ can be computed as $Pop(\{c_2, c_3\}, ct_{maxCTL(\{c2,c3\})}) = \max\{|ct_2|, |ct_3|\} - |\{c_2, c_3\}| = 6 - 2 = 4$. Hence, the popularity of crime pattern $\{c_2, c_3\}$ is

$$\frac{1}{|DB_{\{c2,c3\}}|}\left(Pop(\{c_2, c_3\}, ct_2) + Pop(\{c_2, c_3\}, ct_3)\right) = \frac{1}{2}(3+4) = 3.5$$

The sum of crime transaction length is $sumCTL(\{c_2, c_3\}) = 11$. Then, the popularity of crime pattern $\{c_2, c_3\}$ is $\frac{sumCTL(\{c_2,c_3\})}{|\{ct_2,ct_3\}|} - |\{c_2, c_3\}| = \frac{11}{2} - 2 = 3.5$. In the same way, the popularity of crime pattern $\{c_2\}$ is

$$\frac{sumCTL(\{c_2\})}{|\{ct_1, ct_2, ct_3, ct_6\}|} - |\{c_2\}| = \frac{|ct_1| + |ct_2| + |ct_3| + |ct_6|}{|\{ct_1, ct_2, ct_3, ct_6\}|} - |\{c_2\}| = \frac{16}{4} - 1 = 3$$

If the user specified min_pop is 3.3, then crime pattern $\{c_2, c_3\}$ is popular in the CTDB shown in Table 2 because $Pop(\{c_2, c_3\}) = 3.5 \geq 3.3 = min\_pop$. However, crime pattern $\{c_2\}$ is *not* popular because $Pop(\{c_2\}) = \frac{1}{|DB_{\{c_2\}}|}$ $(Pop(\{c_2\},ct_1) + Pop(\{c_2\}, ct_2) + Pop(\{c_2\},ct_3)+Pop(\{c_2\},ct_6))$

$= \frac{1}{4}(1 + 4 + 5 + 2) = 3 < 3.3 = min\_pop$. So crime pattern $\{c_2\}$ is unpopular but its superset $\{c_2, c_3\}$ is popular which does not satisfy downward closure property. To handle such type of challenge we can redefine the popularity of a pattern C.

Popularity Pop(C) of a pattern C in the CTDB measures an aggregate membership degree of C in the CTDB. It is defined in terms of $sumCTL(C) = \sum_{ct_j \in DB_C} |ct_j|$ as follows.

$$Pop(C) = \frac{1}{|DB_C|}\sum_{ct_j \in DB_C} Pop(C, ct_j)$$

$$= \frac{1}{|DB_C|}\sum_{ct_j \in DB_C} (|ct_j| - |C|)$$

$$= \frac{sumCTL(C)}{|DB_C|} - |C| \qquad \text{--- } 4$$

We notice that $sumCTL(\{c_2, c_3\}) = 11 \leq 16 = sumCTL(\{c_2\})$. The definition of $sumCTL(C)$ further confirms that the total crime transaction length $sumCTL(C)$ of C satisfies the downward closure property (i.e., $sumCTL(C) \geq sumCTL(C')$ if $C \subseteq C'$). For mining popular crime patterns we are proposing a PCrime-growth algorithm which is having two phases like FP-growth algorithm. The unpopular crime patterns should be in PCrime-tree because it is possible that its supersets may be popular. In this process, we first build a tree structure called Popular Crime pattern tree (PCrime-tree) to capture the essential information from the CTDB with only two database scans of the CTDB. Fortunately, recollect from Section 2 that $sumCTL(C)$ satisfies the downward closure property. So, not all unpopular items need to be kept. Some of them can be pruned by using following properties [16].

1. *The popularity of a crime pattern C is always less than or equal to its long crime transaction popularity, i.e.,* $Pop(C) \leq Pop(C, ctmaxCTL(C))$.
2. *For* $C \subseteq C'$, $Pop(C')$ *cannot exceed* $maxCTL(X) - |C'|$.

Based on the above two equations, the following equation provides us with an upper bound of the popularity $Pop(C')$ of a pattern C' (in terms of $maxCTL(C)$), where $C \subseteq C'$:

$$Pop^{UB}(C') = maxCTL(X) - |C'|. \qquad \text{--- } 5$$

Based on *Equation* (5), we can calculate the popularity upper bound of a pattern C' from $maxCTL(C)$ (where $C \subseteq C'$, and $|C'| = |C| + 1 = k+1$), and prune unpopular crime patterns. We call this *super-crime pattern popularity check*. Similar to FP-tree [8], each node of a Pop-tree contains the parent and child pointers as well as horizontal node traversal pointers. To facilitate popular crime pattern mining, we keep (i) a crime pattern c, (ii) support of $Y \cup \{c\}$, (iii) $sumCTL(Y \cup \{c\})$, and (iv) $maxCTL(Y \cup \{c\})$, where Y represents the set of items above c (i.e., ancestor nodes of c) in the form of { c: *support*(c), *maxCTL*(c), *Pop*(c) }.

To construct a PCrime-tree we need to scan the CTDB to find the *support*(c), maximum crime transaction length *maxCTL*(c) and the popularity *Pop*(c) for each item c in the CTDB. Then, we can perform the super-pattern popularity check and safely delete a pattern c if $PopUB(c')$ < *min_pop* (where c' is an extension of c). We then scan the CTDB the second time to insert each transaction into the PCrime-tree in a similar fashion as the insertion process of FP-tree.

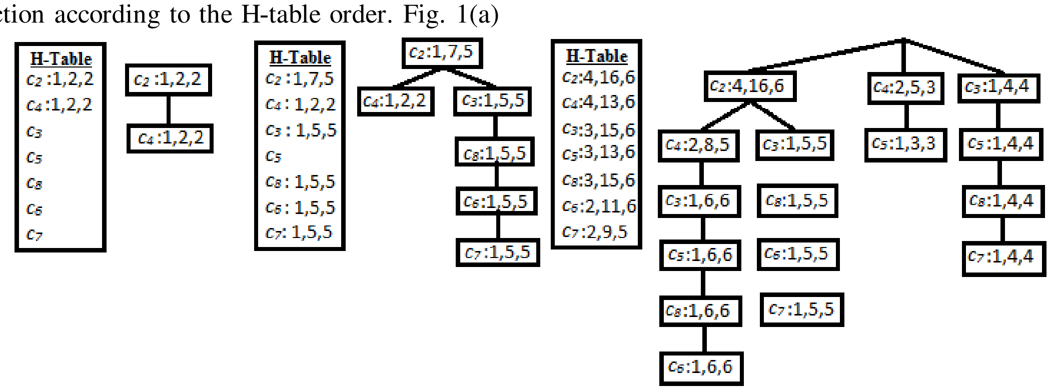
Example Let us show how to construct a PCrime-tree for the CTDB shown in Table 2 with *min_pop* = 2.4. From the first database scan we find the following information in the form of c: *support*(c), *maxCTL*(c), *Pop*(c) for each of the m=9 domain items, i.e., (c_1:3,3,1.66), (c_2:4,6,3.0), (c_3:3,6,4.0), (c_4:4,6,2.25), (c_5:3,6,3.33), (c_6:2,6,4.5), (c_7:2,5,3.5), (c_8:3,6,4.0), (c_9:1,3,2.0). All the items except c_1, c_4 & c_9 are popular crimes (i.e., with popularity at least 2.4). The crimes c_1, c_4 & c_9 are unpopular but their super-patterns may be popular. So we cannot delete these three crimes without checking their super-pattern popularity. The popularity upper bounds of the extensions of c_1, c_4 & c_9 are 3 – 2 = 1, 6 – 2 = 4 and 3 – 2 = 1 respectively. As the value for c_4 is greater than *min_pop*, we keep c_4 but safely delete c_1 & c_9. Now we sort and insert crime items c_2, c_3, c_4, c_5, c_6, c_7 and c_8 into a header table (H-table) in support descending order of crimes: { c_2, c_3, c_4, c_5, c_6, c_7, c_8 }.

Once again we scan the CTDB database to compute the length of each transaction and to remove all items that are not in the H-table and also sort the remaining items in each transaction according to the H-table order. Fig. 1(a) shows the contents of the H-table (c: *support*(c), *sumCTL*(c), *maxCTL*(c)) and the PCrime-tree structure after inserting first crime transaction ct_1 of CTDB. As ct_1 and ct_2 share a common prefix (i.e., {c_2}), we increase the occurrence count of the common node {c_2 : 1, 2, 2} by one, and its total crime transaction length (*sumCTL*) by the crime transaction length of ct_2 (i.e., $|ct_2|$= 5) and update its *maxCTL*. For the remaining uncommon nodes of ct_2 we set *support* =1, *sumCTL*=$|ct_2|$ and *maxCTL*=$|ct_2|$. The contents of the PCrime-tree after insertion of ct_2 are shown in Fig. 1(b). The final PCrime-tree after capturing all the transactions in the CTDB is shown in Fig. 1(c).

The PCrime-growth finds popular crime patterns from the PCrime-tree in which each tree node captures the number of occurrences, total crime transaction length, and maximum crime transaction length. Then our algorithm finds popular crime patterns by constructing the projected database for potential popular itemsets and recursively mining their extensions. While constructing the conditional database from a projected database, a super-pattern popularity check for extensions of any unpopular crimes will take place, and deletes the crime when it fails the check. Such pruning technique is called as the *lazy pruning*. This lazy pruning technique ensures that no popular crime patterns will be missed by PCrime-growth algorithm and popular crime patterns can be found by mining our PCrime-tree.

5. Experiment Results

Let us consider the distinct crime database attributes in the CTDB



(a) PCrime-tree captures ct_1 (b) PCrime-tree captures ct_1 &ct_2 (c) PCrime-tree captures $ct_1 – ct_7$

Figure 1. The PCrime-tree Construction

5.1 LAPD Crime and Collision Raw Data - 2014

In order to do experiments we mostly use the datasets which are commonly used in frequent pattern mining experiments as the characteristics of these transactional datasets are well known. Including these datasets we also use IBM synthetic datasets from www.almaden.ibm.com/cs/quest and real datasets from frequent itemset mining dataset repository fimi.cs.helsinki.fi/data. By using this data we obtained consistent results. With the help of space constraint, we represent here the experimental results on a subset of these datasets.

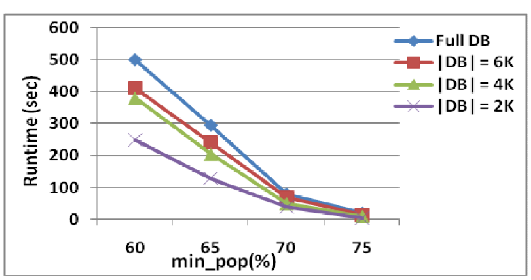
Table 3: Datasets

dataset	*transactions*	*items*	*maxTL*	*avg TL*	*Data density*
T10I4D100K	**1,00,000**	**870**	**29**	**10.10**	**Sparse**
T20I4D100K	**99,996**	**871**	**42**	**19.81**	**Sparse**
mushroom	**8,124**	**119**	**23**	**23.00**	**dense**

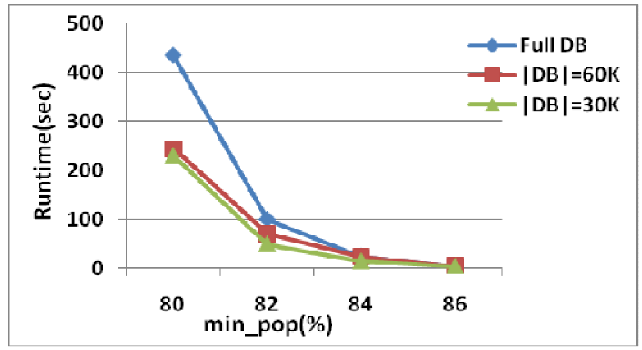
Generally all the programs are written in Java and run in windows environment with 1.3 GHz processor. The programs which describes the run time indicates the total execution time. The results that are reported are based on average multiple run in every case. In the below experiments the pcrime-trees were constructed using descending order of support counts. The performance of pcrime-tree structure and pcrime-growth algorithm when varying the mining parameters such as popularity threshold and dataset characteristics has been presented.

Runtime

The execution time which the pcrime-growth requires for mining popular crime patterns over datasets of different types and changes in min_pop has been reported. The execution time includes all the steps of H-table construction, the pcrime-tree



(a). Mushroom



(b). T2014D100K

Figure 2. Runtime of Mushroom and T2014D100K

building and the corresponding mining process. In the above figure we represent the results on one sparse dataset and other result on dense dataset. The effect of mining on the variations in size of such datasets can be observed by performing popular pattern mining while increasing the size of datasets.

Compactness of pop-tree

In this section we report the compactness of pcrime-tree in terms of number of pcrime-tree nodes. As the mushroom dataset has a fixed transaction length, the maximum transaction length for every possible pattern is always same. Since every item undergoes lazy pruning phase so each item tree size will be same with varying the min_pop. With the increase of min_pop the size of tree structure gradually reduced in T20I4D100K. With the increase in size of database, the number of nodes gets increased in both the datasets.

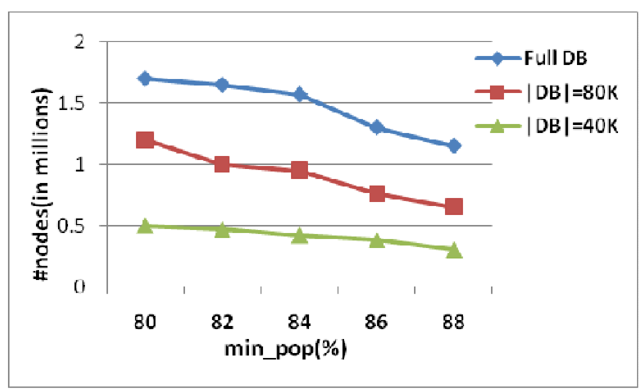


Figure 3. Compactness of the PCrime-tree on T20I4D100K

The results demonstrate that pcrime-tree can mine the set of popular patterns on these dataset for reasonable small value of popularity threshold with particular amount of execution time.

6. Conclusion

In this paper we introduced a new algorithm called PCrime-growth algorithm that finds popular crime patterns from Crime datasets. With the first database scan it constructs PCrime-tree and also computes support of a crime, maximum crime transaction length and popularity of each crime. After that the algorithm performs for super-pattern popularity for unpopular crimes to prune the crime database. After second scan it's going to compute length of each crime transaction by eliminates unpopular crimes and will extracts popular crime patterns from the crime databases. Our experimental results showed that our PCrime-tree is time and space efficient for both sparse and dense datasets. In addition these results are also affected on crime datasets on construction of PCrime-tree and mining of popular crime patterns to be time efficient.

References

[1] N. G. Khan, V. Bhaga, Effective data mining approach for crime-terrorpattern detection using clustering algorithm technique, Engineering Research and Technology International Journal Vol 2 (4) (2013), pp. 2043–2048.

[2] H. Chen, W. Chung, J. Xu, G. Wang, Y. Qin, M. Chau, Crime data mining: A general framework and some examples, IEEE Computer Journal 37 (4) (2004) 50–56.

[3] Mohammad, J. Mohsen, E. Martin, G. Uwe, F. Richard, Crimewalker: A recommendation model for suspect investigation, in: Proc. fifth ACM conference on Recommender systems, ACM, 2011, pp. 1–8.

[4] O. Isafiade, A. Bagula, Citisafe: Adaptive spatial pattern knowledge using fp-growth algorithm for crime situation recognition, in: Proc. IEEE International Conference on Ubiquitous Intelligence and Computing, IEEE, 2013, pp. 551–556.

[5] P. Phillips, I. Lee, Mining co-distribution patterns for large crime datasets, Expert Systems with Applications International Journal 39 (14) (2012) 11556–11563.

[6] D. Wang, W. Ding, H. Lo, T. Stepinski, J. Salazar, M. Morabito, Crime hotspot mapping using the crime related factors- a spatial data mining approach, Applied Intelligence Journal 39 (4) (2013) 772–781.

[7] Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: VLDB, 1994, pp. 487–499 (1994).

[8] Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: ACM SIGMOD 2000, pp. 1–12 (2000).

[9] Lee, Y.-K., Kim, W.-Y., Cai, Y.D., Han, J.: CoMine: efficient mining of correlated patterns. In: IEEE ICDM 2003, pp. 581–584 (2003).

[10] Rasheed, F., Alshalalfa, M., Alhajj, R.: Efficient periodicity mining in time series databases using suffix trees. IEEE TKDE 23(1), 79–94 (2011).

[11] Zhang, M., Kao, B., Cheung, D.W., Yip, K.Y.: Mining periodic patterns with gap requirement from sequences. ACM TKDD, 1(2), art. 7 (2007).

[12] Lakshmanan,L.V.S., Leung, C.K.-S.,Ng,R.T.:Efficient dynamic mining of constrained frequent sets. ACM TODS 28(4), 337–389 (2003).

[13] Leung, C.K.-S., Sun, L.: A new class of constraints for constrained frequent pattern mining. In: ACM SAC 2012, pp. 199–204 (2012).

[14] Xiong, H., Tan, P.-N., Kumar, V.: Hyperclique pattern discovery. Data Mining and Knowledge Discovery 13(2), 219–242 (2006).

[15] Yao, H., Hamilton, H.J.: Mining itemset utilities from transaction databases. DKE 59(3), 603–626 (2006).

[16] Leung C.K.-S., Tanbeer S.K.: Mining Popular Patterns from Transactional Databases. Springer DaWak 2012, 291-302 (2012).

[17] Leung C.K.-S., Tanbeer S.K.: Finding Popular Friends in Social Networks. IEEE Second International Conference on Cloud and Green Computing, 501-508 (2012).

[18] Omowunmi Isafiade, Antoine Bagula, Sonia Berman.: A Revised Frequenrt Pattern Model for Crime Situation Recognition Based on Floor-Ceil Quartile Function. Elsevier Science Direct, Procedia Computer Science 55 (2015), 251-260.

[19] Brown, D.E. The regional crime analysis program (RECAP): A frame work for mining data to catch criminals," in Proceedings of the

IEEE International Conference on Systems, Man, and Cybernetics, Vol. 3, Pp. 2848-2853 (1998).

[20] De Bruin, J.S. , Cocx, T.K. , Kosters, W.A. , Laros, J. And Kok, J.N. Data mining approaches to criminal career analysis," in Proceedings of the Sixth International Conference on Data Mining (ICDM‟06), Pp. 171-177 (2006).

[21] Abraham, T. and de Vel, O. Investigative profiling with computer forensic log data and association rules," in Proceedings of the IEEE International Conference on Data Mining (ICDM'02), Pp. 11 – 18 (2006).

Authors -

B.V.S.Varma is Associate Professor in the department of Computer Science & Engineering at Ideal Institute of Technology, India. He obtained M.Tech degree in CSE from Dr. M.G.R.University and B.Tech in CSE from Madras University. He has 13 years of Teaching Experience He research interest includes Data Mining.

V.Valli Kumari is professor in the department of Computer Science & Systems Engineering and Director of Computer Center at Andhra University, India. She has received best Researcher Award from the same university. She has 17 years of teaching experience and guided several Ph.Ds. she is an active researcher and executed several research projects. She has published more than 120 research papers in international conferences and journals, authored several book chapters and chaired several international conferences.