IMPACT: International Journal of Research in Engineering & Technology (IMPACT: IJRET)
ISSN(E): 2321-8843; ISSN(P): 2347-4599
Vol. 2, Issue 4, Apr 2014, 129-136

CLUSTERING FOR FORENSIC ANALYSIS

SAHIL KHENAT, PRATIK KOLHATKAR, SARANG PARIT & SHARDUL JOSHI

Department of Computer Engineering, UCOER, Pune University, Pune, Maharashtra, India

ABSTRACT

In today's digital world, information in computers has great importance and this information is very essential in context for future references and studies irrespective of various fields. So surveying of such information is critical and important task. In computer forensic analysis, a lot of information present in the digital devices (computers in context of our paper) is examined to extract information. And computer consists of hundreds of thousands of files which contain unstructured text or information, so clustering algorithms are of great interest. Clustering helps to improve analysis of documents under consideration. This document clustering analysis is very useful for crime investigations to analyze the information from seized digital devices like computers, laptops, hard disks, tablets. There are total six algorithms used for clustering of documents like K-means, K-medoids, single link, complete link, Average Link, CSPA. These six algorithms are very efficiently used to cluster the digital documents. These algorithms make the analysis very fast by clustering very close documents in one cluster. Also two validity index are used to find out how many clusters are formed.

KEYWORDS: Clustering, CSPA, K-Mean, K-Medoids

INTRODUCTION

Estimates that are proposed by IT IS are digital data density increased 18 times in latest 5 to 6 years. Here in year 2006, data density that is volume which is 161 hexabytes raised to 988 hexabyte in year 2011. And is growth is going exponentially. So as digital world contains very important, complex and unstructured data, clustering algorithms play important role in forensic analysis of such digital documents

In particular domain related to paper, hundreds of thousands of documents are examined. And this surveillance which exceeds capabilities of expertise which monitors or analyze such documents. So it is very prime requirement to make data simple and to use some techniques which boosts the analysis of complex, unstructured documents. And here Data mining techniques as well as pattern recognizing techniques are of great importance. Because clusters are going to form according to their pattern also Clustering algorithms are used to explore data without knowledge of data.

The main purpose behind why to perform clustering is to group the documents which having some sort of similar data. It is like grouping of similar things together so searching or finding can perform efficiently. The same concept is carried out here in clustering. So that particular cluster *Ci* consist of documents *Di* contains some sort of similar content. So that now experts can focus on particular cluster for any document rather than analyze all documents. Also there are in all 6 clustering algorithms namely K-means, K-medoids, Single link, complete link, Average Link, CSPA. With different parameters, when algorithms were executed gives sixteen new instantiations. Silhouette and its simplified version are two relative indexes which are used to finding out number of clusters that will be made after algorithm implementation.

In any digital forensic analysis, number of clusters is very important and critical factor which is mainly unknown and also there is no investigation has been made on cluster numbers. But as clusters are very useful for fast investigation of digital documents, our paper consist of classical clustering algorithms also newest researches like consensus partitions. Here our paper is organized in following main points:

- Introduction
- Classical Clustering Algorithms and Pre-processing Steps
- Proposed System
- Limitations
- Conclusion
- References

CLASSICAL CLUSTERING ALGORITHMS AND PRE-PROCESSING STEPS

The important step before running the clustering algorithms is the preprocessing. Pre-processing consisting of stopword removal and stemming. Stopword removal is process in which particular words are removed which will not affect the meaning of document. This consists of removal of prepositions, pronouns, articles and other irrespective words. After preprocessing, the main statistical approach for text mining is adopted. In this Statistical approach, document represented in vector model format, each document is represented as vector model consisting of words according to their frequencies of occurrence.

Reduction technique known as Term Variance (TV) is also used to increase efficiency of clustering algorithms. As clusters are formed which containing documents, TV are used to estimate top *n* words which have greatest occurrences over documents within clusters. So that this is very important factor for formation of cluster. Also it is important to find out distances between two documents when they are resides in different clusters. And for finding out distances between them, cosine-based distance and Levenshtein -based distance

As to find out number of clusters, by looking for best set of partitions of data set from different clusters which gives best result by relative index or different combinations of attributes are selected and after various runs(k-mean), best is selected. So by considering that, there are various sets of data partitions with different clusters, from which we have to choose best one. And one of the ways to finding out best partition from sets is Levenshtein-based algorithms so it is a important component in our studies.

Suppose there is object x in cluster A. And dissimilarity of x with other objects in same luster that is A is a (i). Now consider cluster C. And average dissimilarity of object x to cluster C is d(x, C). As we have to find o dissimilarities within neighbors clusters following technique is used. After computing the dissimilarity d(x, C) with all clusters except A, smallest one is selected that is b(x).

Value of dissimilarity neighbors is finding out by formula

$S(x) = b(x)-a(x)/ \max \{a(x), b(x)\}$

Value S(x) is verified in between age of -1 to 1. If value of S(x) is higher, object x belongs to particular cluster but if S(x) is zero, then it is not clear that whether object belongs to current cluster or adjacent one. S(x) is carried out over i=1, 2,…,n where n are no of objects and then average is computed. And best clustering is maximum S(x).

Hence to finding out effective S(x), i.e. called *simplified Silhouette*, one can compute only the distances among the objects and the centroids of the clusters. So a(x) is dissimilarity correspond to or simply belongs to cluster (A) centroid. Now it is very easy to get only one distance rather than finding out whole dissimilarity between all objects within cluster. Also rather than finding out d (x C). C not equal to a, we will find only distance with centroid.

Table 1: Information Found in Clusters

Cluster	Information
C1	6 LIC policies
C2	2 bank accounts
C3	5 grocery list
C4	1 loan agreement 2 check receipt
C5	5 office applications
C6	3 financial transactions
C7	5 investment club status

PROPOSED SYSTEM

Consider one folder as an input. It contains six files. Our software will take one file at a time. And then it will go under following processes:

Consider one folder as an input. It contains six files. Our software will take 1st file at a time. And then it will go under following processes.

- **Pre-Processing Module**
  - **Fetch the Content of File:** This will be our 1st sub step in pre-processing module in which content of our input file will be fetched by our software for further processing.
  - **Stopword Removing:** Fetched content of input file contains lot of stopwords i.e. the words which don't have important meaning.

For example, suppose our input file contains sentence as

"Here we are learning java".

In this sentence, we have words like "here", "we", "are" which are not important for further processing i.e. stemming. So we will remove those words from our original sentence and we just pass words like "java", "learning" to further step i.e. stemming.

- **Stemming:** This is the step where we bring down the word to its original base form. Consider the same example of sentence.

Example- "Here we are learning java". In this sentence, after removing of stopwords, we get words "learning", "java" for stemming.

In stemming, we will bring the word “learning” to its base form as “to learn”.

For this, we use algorithms ‘Port stemmer’, but problem with this already existing algorithm is it don’t return some words to its bas form with correct spelling or with correct meaning.

For Example: The word Studied, It returns “to studi”, and not “To study”. And also words like ‘String’, which are already in base form containing ‘-ing’. That after removing ‘-ing’, words become meaningless.

- **Preparing Cluster Vector**

From first module, we get pre-processed content of the input. We will calculate weight i.e. frequency (no of occurrences) of each word. And we will arrange these words in descending order according to their weights. Out of them, we take certain number of words *as* Top *n* words. And we will maintain its array, say Array A.

In second module, usually the first sentence is the most important and which is probably it is most suitable s the title of the document. In such a way every document has its first sentence as its title which results in formation of Array B. Our input document may contain few numerical values which can provide us very important information, so we will gather those numerical values and we will maintain its different array, Array C.

Now, we have three different arrays A, B, C of ‘Top *n* words’, ’title’, ’numerical values’ respectively. So we will combine all three arrays and we will form one master vector *Mv*.

Likewise we will create master vectors for all the remaining files.

- **Forensic Analysis**

Now, we have master vectors for all the files in that particular input folder. So, here we going to compare all this master vectors with each others to find out the similarities between master vectors on the basis of accuracy. This process is named as mapping of master vectors. The accuracy is given by user according to their requirement.

For example, consider file folder consisting of 5 text files, 2 image files, 3 video files. File formats other than text files will get filtered out. Only .txt, .doc, .docx files are considered for analysis of documents. Therefore 5 text files will have their respective 5 master vectors {MV1, MV2, MV3, MV4, MV5}. Now we apply the process of mapping on these master vectors. User will enter accuracy on the basis of which the further process will be executed.

Constraint of accuracy is that 0<accuracy<1.

Table 2: Statistical Mapping of Clusters

	MV1	MV2	MV3	MV4	MV5
MV1	0	0.56	0.98	0.45	0.3
MV2	0.56	0	0.63	0.52	0.41
MV3	0.98	0.63	0	0.75	0.18
MV4	0.45	0.52	0.75	0	0.23
MV5	0.3	0.41	0.18	0.23	0

Suppose, user enters accuracy as 0.5. Then all master vectors having accuracy quotient equal to or more than 0.5 will be combined together to form revised master vector (R.M.V.). So, in this case revised master vectors will be formed in following manner.

- MV1 and MV2→ RMV12
- MV1 and MV3→ RMV13
- MV2 and MV3→ RMV23
- MV2 and MV4→ RMV24
- MV3 and MV4→ RMV34

In such a way, we get five revised master vectors which contain data depending on accuracy entered by user.

Note: Why accuracy should not be 1?

If user enters accuracy as 1, then that means while mapping master vectors of files data from those master vectors should be exactly same, which is not practical and it is inefficient.

Due to this, our system will skip very crucial information might help in investigation and carries important data.

Algorithm for Cluster Vector Creation

CL = [D, S, SVSM, Sim (d, s_i), C]

Input

D = {d_1, d_2; : : : ; $d_{|D|}$}

Where set of input documents to be clustered

S = {s_1; s_2; : : : ; s_n}

set of n subjects, where each subject si is a set of weighted terms

Sim (d, s_i) = similarity function

C = {C_1, C_2…… C_n}

Where c is set of n overlapping output clusters

Cluster Vector

T_w=Term weight

D = {d_0, d_1, d_2,.,..,……..d_n}

Where,

D=Document Set

d_0, d_1,…….,d_n=words of documents

N_d=Numerical data

Output

V_c = cluster vector

Algorithm

Step 0: Get T_w and D

Step 1: Sort T_{100} in descending order

Step 2: Add top 100 words to T_{100}

Step 3: Fetch 1st sentence as T_t

Step 4: For each d_i

Step 5: Check for numeric data

Step 6: Then add into $N_d = \{n_1, n_2, n_3, \ldots\ldots, n_n\}$

Step 7: Merge T_{100}, T_t, and N_d

Step 8: $V_c = \{T_{100}, T_t, N_d\}$

Algorithm for Forensic Analysis

Input

V_c = Cluster Vector

Output

$F_r = \{c_1, c_2, c_3, \ldots\ldots, c_n\}$

Algorithm

Step 0: Get $V_c = \{T_t, T_{100}, N_d\}$

Step 1: Get T_t

Step 2: for each sentence s_i

Step 3: Find title word of T_t

Step 4: Rank top 5 sentences

Step 5: Get T_{100}

Step 6: For each sentence s_i

Step 7: Find frequency of T_{100} words

Step 8: Rank top 5 sentences R_{100}

Step 9: Get N_d

Step 10: for each sentence s_i

Step 11: Find N_d in each sentences

Step 12: Rank top 5 sentences, R_{nd}

Step 13: Merge R_t, R_{100}, R_{nd}

Step 14: Remove repeated sentences to get important (Imp) vectors

Step 15: for each document

Step 16: Compare Imp vector

Step 17: If comparison is more than 50% then cluster the documents

Step 18: Stop

LIMITATIONS

It is known that very well, that working of any clustering algorithm depends on data, but for estimated datasets some of our versions of clustering algorithms have shown good results. One of the prominent issues is scalability. In order to handle this issue, sampling and other techniques can be used. Also algorithms like bisecting k-means and associated approaches can be used. These algorithms can persuade dendograms. They have a similar inductive bias with respect to the various hierarchical methods in our work. More precisely, aiming at evading the computational difficulties, partitional clustering algorithms can be used for computing a hierarchical clustering solution by using repeated cluster bisectioning techniques. For illustration, bisecting k-means has comparatively low computational requirements. i.e. it is O(N.log N), versus the overall time complexity O(N^2.logN) for specially agglomerative methods. If the number of documents is extremely high for running an agglomerative algorithm, in that case, bisecting k-means and other related approaches can be used as a solution.

When we consider the cost of computation for estimating the number of clusters, the silhouette introduced in [1] is based mainly on all the distances between objects. This leads to a computational cost of O (N^2.D), where N is number of objects in dataset and D is number of attributes. As a solution to this potential difficulty, especially when very large size datasets come into the picture, a simplified version of silhouette can be used. Simplified silhouette includes computation of distance between objects and cluster centroids. This reduces computational cost from O (N^2.D) to O (k.N.D) where k is the number of clusters. Here value of k is very less than value of N. In our work we have adopted silhouettes. Instead there are several relative criteria which can be used as a replacement or alternative for it. Such criteria have abilities that make each of them to outperform others in particular classes of problems. Practically, one can use or test different criteria to compute number of clusters by considering both the quality of data partitions and respective computational cost. We would also like to mention that practically it is not that important to use scalable methods. In our case, there are no severe restrictions to get data partitions. Instead of that, domain experts can utilize a lot of time to analyze their input data before reaching up to a particular conclusion.

CONCLUSIONS

We have introduced an approach which can become an ideal application for document clustering to forensic analysis of computers, laptops and hard disks which are seized from criminals during investigation of police. There are several practical results based on our work which are extremely useful for the experts working in forensic computing department. In our work, the algorithms known as Average Link and Complete Link yielded the best results. In spite of these algorithms having high computational costs, they are suitable for our work domain because dendrograms provides a

neat summary of documents which are being inspected. All the textual documents are scanned thoroughly and corresponding output is given. When proper initialization is done, the partitional K-means and K-medoids algorithms also have satisfactory results.

When estimation of number of clusters is to be done and approaches for doing the same are considered, at that time the simplified version of silhouette in less efficient as compared to its relative validity criterion which is far more accurate than simpler version. Additionally, in some results it was observed that making utilization of the file names along with the actual document information or content would prove to be very useful for document ensemble algorithms. Another important observation which we came across was that clustering algorithms tend to bring about clusters formed by either relevant or irrelevant document set. This leads to enhancement of expert examiner's job or task. Further ahead, our proposed approach has the capacity to boost the speed of computer inspection by an impressive factor.

REFERENCES

1. Document Clustering for Forensic Analysis: An Approach for Improving Computer Inspection Luís Filipe da Cruz Nassif and Eduardo Raul Hruschka.
2. L. Kaufman and P. Rousseeuw, *Finding Groups in Gata: An Introduction to Cluster Analysis*. Hoboken, NJ: Wiley-Interscience, 1990.
3. B. S. Everitt, S. Landau, and M. Leese, *Cluster Analysis*. London, U.K.: Arnold, 2001.
4. B. Mirkin, *Clustering for Data Mining: A Data Recovery Approach*. London, U.K.: Chapman & Hall, 2005.