ISSN : 2231-0711

Document Clustering for Forensic Analysis via Cosine Similarity

Priyanka kasote
G.H.R.I.E.T.
B.E (I.T.)

Shruti singh
G.H.R.I.E.T.
B.E (I.T.)

Ms. Chhaya Varade
G.H.R.I.E.T.

***Abstract*-In study of computer forensic, hundreds and thousands of files are usually inspected. Most of the data is present in amorphous form. In this framework mechanized methods of investigation are of great interest. In specific, algorithms used for clustering documents can ease the innovation of fresh and useful information from the document under study. We use a methodology that relates the document clustering algorithms to forensic analysis of computer apprehended in police investigations. We demonstrate our method by using the well-known clustering algorithms (CSPA and cosine similarity matrix). Research have been implemented with different combinations of parameters, causing in 16 different instantiations of algorithm, connected study in literature are more restricted than our study. In our study we use CSPA algorithm for clustering which will include various sub algorithms such as cosine similarity matrix and text processing algorithm.**

I. INTRODUCTION

It is projected that the capacity of records in the digital world has amplified to an abundant range. This huge quantity of records has a straight impact in computer forensics, which can be generally defined as the restraint that syndicates components of law and computer science to gather and examine records from computer systems in a way that is adequate as proof in a court of law. In our specific application domain, it typically comprises inspecting hundreds of thousands of records per computer. This action leads to a challenge in examining and understanding the records. To decrease this problem numerous automated data analysis techniques were used. Technique like those broadly used for machine learning and data mining are of supreme importance. Specific algorithms for pattern recognition from the information in text document are favorable, as it will become evident later in paper.

Clustering algorithms are typically used for examining data analysis, where there is very scarce information about the records. This case is encountered in many application of computer forensics. Our datasets consists of unstructured objects, the classes or categories of document that can be found are a priori unidentified. Whereas, even structured datasets acquired from previous investigation, have no hope to have valid classes for the upcoming data, which is gathered from other investigation processes. Which means that, new data sample would come from different population. Here we can use the clustering algorithms, which are capable of finding latent patterns from text documents found in apprehended computer, can improve the analysis executed by the forensic examiners.

In clustering algorithms the objects within a valid cluster are more alike to each other than they are to objects belonging to different cluster. Hence, if once a data partition has been brought from data, the expert might focus initially on reviewing representative documents from the obtained set of clusters. After the prelim examination the forensic examiner may decide to view the detailed report.

Practically, domain experts are scarce and have limited time available for performing examinations. Thus, after finding a relevant document, the examiner could prioritize the analysis of other documents belonging to the cluster of interest, because it is likely that these are also relevant to the investigation.

Clustering algorithms have been studied from many years, and the literature is huge. Our reference paper deals with 6 different well known algorithms namely: the partition algorithm K-means and K-medoids, the hierarchical single/complex/average link. Whereas, we are focusing to use only CSPA algorithm. In addition we will also use cosine similarity matrix and text processing algorithm. This is the additional part of the context

II. LITERATURE SURVEY

There are only a few studies reporting the use of clustering algorithms in the *Computer Forensics* field. Essentially, most of the studies describe the use of classic algorithms for clustering data—e.g., Expectation-Maximization (EM) for unsupervised learning of Gaussian Mixture Models, K-means, Fuzzy C-means (FCM), and Self-Organizing Maps (SOM). These algorithms have well known properties and are widely used in practice. For instance, K-means and FCM can be seen as particular cases of EM. Algorithms like SOM, in their turn, generally have inductive biases similar to K-means, but are usualness' computationally efficient. In, SOM-based algorithms were used for clustering files with the aim of making the decision-making process performed by the examiners more efficient. The files were clustered by taking into account their creation dates/times and their extensions. This kind of algorithm has also been used in order to cluster the results from keyword searches.

The underlying assumption is that the clustered results can increase the information retrieval efficiency, because it would not be necessary to review all the documents found by the user anymore. An integrated environment for mining e-mails for forensic analysis, using classification and clustering algorithms, was presented. In a related application domain, e-mails are grouped by using lexical, syntactic, structural, and domain-specific features. Three clustering algorithms (K-means, Bisecting K-means and EM) were used. The problem of clustering e-mails for forensic analysis was also addressed, where a Kernel-based variant of K-means was applied. The obtained results were analyzed subjectively, and the authors concluded that they are interesting and useful from an investigation perspective.

More recently [13], a FCM-based method for mining association rules from forensic data was describe

III. PROJECT MODULE

A. User Authentication

This module will enable the system to validate users and enable only valid users to access the application. The users will be facilitated to register themselves and access the application.

B. Document Upload

This module will facilitate the users to upload text documents (txt, rtf formats) which will be analyzed by the system for the clustering mechanism. This module also enables users to view uploaded files.

C. Admin Module

This module will facilitate admin to view the complete documents in the system. Admin performs the main job of clustering with the help of cosine similarity matrix.

D. Document Clustering

This section will be executed by the admin to segregate the information populated by the user into a set of cluster and will also include the proposed mechanism with **cosine similarity matrix** analysis which will include procedures like stemming, tokenization, stop word removal and filtering as well as clustering process which will be finalized later.

E. View Clusters & Download

The clusters created will be visible here in segregated format for identifying the output of the above process. The user can view and download the required cluster of the provided file.

IV. ALGORITHMS

Cosine similarity matrix

It is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. The cosine of 0° is 1, and it is less than 1 for any other angle. It is thus a judgment of orientation and not magnitude: two vectors with the same orientation have a Cosine similarity of 1, two vectors at 90° have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of their magnitude. Cosine similarity is particularly used in positive space, where the outcome is neatly bounded

Note that these bounds apply for any number of dimensions, and Cosine similarity is most commonly used in high-dimensional positive spaces. For example, in Information Retrieval and text mining, each term is notionally assigned a different dimension and a document is characterized by a vector where the value of each dimension corresponds to the number of times that term appears in the document.

The technique is also used to measure cohesion within clusters in the field of data mining.

Cosine distance is a term often used for the complement in positive space, that is:

$$D_C(A,B) = 1 - S_C(A,B)$$

It is important to note, however, that this is not a proper distance metric as it does not have the triangle inequality property and it violates the coincidence axiom; to repair the triangle inequality property whilst maintaining the same ordering, it is necessary to convert to Angular distance.

One of the reasons for the popularity of Cosine similarity is that it is very efficient to evaluate, especially for sparse vectors, as only the non-zero dimensions need to be considered.

Cluster-based Similarity Partitioning Algorithm (CSPA)

Essentially, if two objects are in the same cluster then they are considered to be fully similar, and if not they are dissimilar. This is the simplest heuristic and is used in the Cluster-based Similarity Partitioning Algorithm (CSPA). With this viewpoint, one can simply reverse engineer a single clustering into a binary similarity matrix. Similarity between two objects is 1 if they are in the same cluster and 0 otherwise. For each clustering, a n×n binary similarity matrix is created. The entry-wise average of r such matrices representing the r sets of groupings yields an overall similarity matrix. *Figure 1* illustrates the generation of the cluster-based similarity matrix for the example given in table

Alternatively, and more concisely, this can be interpreted as using k binary cluster membership features and defining similarity as the fraction of clustering's in which two objects are in the same cluster. The entire n×n similarity matrix $\mathbf{S}$ can be computed in one sparse matrix multiplication

$$\mathbf{S} = \frac{1}{r}\mathbf{H}\mathbf{H}^{\dagger}$$

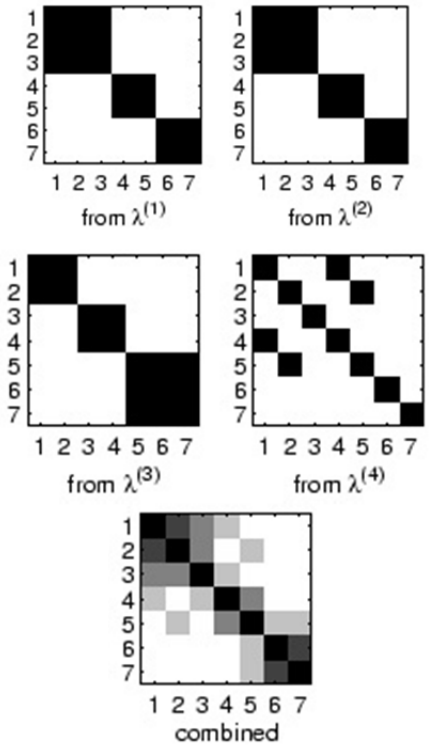

Figure 1: Illustration of Cluster-based Similarity Partitioning Algorithm (CSPA) for the cluster ensemble example problem. Each clustering contributes a similarity matrix (matrix entries are shown by darkness proportional to similarity). Their average is then used to re-cluster the objects to yield consensus.

V. STEPS FOLLOWED

Start with reading each and every document. Remove special characters and punctuation marks are from the plain text document.

Split sentences into individual tokens or words. Reduce the word by removing unwanted word like 'at', 'the', and etc. and compare it with stop word.

Calculate weight for each word by applying TF-IDF (Term Frequency Inverse Document Frequency) scheme. Where ,

TFIDF= (No. of occurrences of term t in this document D) * log((total No. of documents)/(No. of documents with mention of term t)).

Now calculate **cosine** value based on TFIDF for each file in following set format:

$$\begin{bmatrix} 1 & d(1,2) & d(1,3) \ldots & d(1,n) \\ d(2,1) & 1 & d(2,3) & d(2,n) \\ d(3,1) & d(3,2) & 1 & d(3,n) \\ . & . & . & . \\ . & . & . & . \\ d(n,1) & d(n,2) & d(n,3) \ldots & 1 \end{bmatrix}$$

Here each document is compared with every document to get cosine and if its cosine is same then it will return 1 else 0. If cosines of 2 documents are bit similar then it returns value between 0 to 1.

Take the mean (i.e. average) of i'th row and of j'th column , then calculate the mean of the obtained means , the resultant value will be a **threshold**

It would be:

im = a/ n and ***jm= b/n***

Suppose, a= sum of ith row and b= sum of jth column ith & jth

mean=(im+jm)/2

If (documents in ith row and jth column >threshold) then put that document in that cluster

Else repeat steps 6-8.

VI. LIMITATIONS

The history on computer forensic informs that use of algorithms which measured that the number of cluster is known and fixed a priori by the user. Taking into consideration the computational cost of estimating the number of clusters, the silhouette proposed depends on the computation of all distances between objects, leading to an estimated computational cost of $O(N^2 \cdot D)$, where N is the number of objects in the dataset and D is the number of attributes, respectively. As already mentioned in the paper, to alleviate this potential difficulty, especially when dealing with very large datasets, a simplified silhouette can be used.

The simplified silhouette is based on the computation of distances between objects and cluster centroids, thus making it possible to reduce the computational cost from $O(N^2 \cdot D)$ to $O(k \cdot N \cdot D)$, where k, the number of clusters, is usually significantly less than N. It is also worth mentioning that there are several different relative validity criteria that can be used in place of the silhouettes adopted in our work. As discussed in [14], such criteria are endowed with particular features that may make each of them to outperform others in specific classes of problems. Also, they present different computational requirements. In this context, in practice one can try different criteria to estimate the number of clusters by taking into account both the quality of the obtained data partitions and the associated computational cost.

VII. CONCLUSION

By using this proposed 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 the proposed work which are extremely useful for the expert working in forensic computing department

REFERENCES

[1] Shi Na, Liu Xumin and Guan Yong, "Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm" Intelligent Information Technology and Security Informatics (IITSI), vol 74, 2010.

[2] J. F. Gantz, D. Reinsel, C. Chute, W. Schlichting, J. McArthur, S. Minton, I. Xheneti, A. Toncheva, and A. Manfrediz, "The expanding digital universe: A forecast of worldwide information growth through 2010," *Inf. Data, vol. 1, pp. 1–21, 2007.*

[3] A. Strehl and J. Ghosh, "Cluster ensembles: A knowledge reuse framework for combining multiple partitions," *J. Mach. Learning Res., vol.* 3, pp. 583–617, 2002.

[4] E. R. Hruschka, R. J. G. B. Campello, and L. N. de Castro, "Evolving clusters in gene-expression data," *Inf. Sci., vol. 176, pp. 1898–1927,* 2006.

[5] N. L. Beebe and J. G. Clark, "Digital forensic text string searching: Improving information retrieval effectiveness by thematically clustering search results," *Digital Investigation, Elsevier, vol. 4, no. 1, pp. 49–54,* 2007.

[6] R. Hadjidj, M. Debbabi, H. Lounis, F. Iqbal, A. Szporer, and D. Benredjem, "Towards an integrated e-mail forensic analysis framework," *Digital Investigation, Elsevier, vol. 5, no. 3–4, pp. 124–137,* 2009