Volume 6, Issue 5, May 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

# Document Clustering for Computer Forensic Analysis

**Apoorva Jain A, Monisha K Raj, Nisarga S U, Regina Vaz, Neelaja K**
Computer Science, NIEIT, Mysuru, Karnataka,
India

***Abstract:*** ***In computer forensic analysis, hundreds of thousands of files are usually examined. Much of the data in those files consists of unstructured text, whose analysis by computer examiners is difficult to be performed. In this context, automated methods of analysis are of great interest. In particular, algorithms for clustering documents can facilitate the discovery of new and useful knowledge from the documents under analysis. We present an approach that applies document clustering algorithms to forensic analysis of computers seized in police investigations. We illustrate the proposed approach by carrying out extensive experimentation with six well-known clustering algorithms (K-means, K-medoids, Single Link, Complete Link, Average Link, and CSPA) applied to five real-world datasets obtained from computers seized in real-world investigations. Experiments have been performed with different combinations of parameters, resulting in 16 different instantiations of algorithms. In addition, two relative validity indexes were used to automatically estimate the number of clusters. Related studies in the literature are significantly more limited than our study. Our experiments show that the Average Link and Complete Link algorithms provide the best results for our application domain. If suitably initialized, partitional algorithms (K-means and K-medoids) can also yield to very good results. Finally, we also present and discuss several practical results that can be useful for researchers and practitioners of forensic computing.***

***Keywords: CSPA, EAC, SOM, DFD***

## I. INTRODUCTION

Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.While large-scale information technology has been evolving separate transaction and analytical systems, data mining provides the link between the two. Data mining software analyzes relationships and patterns in stored transaction data based on open-ended user queries. Several types of analytical software are available: statistical, machine learning, and neural networks.

Generally, any of four types of relationships are sought:

- Classes: Stored data is used to locate data in predetermined groups. For example, a restaurant chain could mine customer purchase data to determine when customers visit and what they typically order. This information could be used to increase traffic by having daily specials.
- Clusters: Data items are grouped according to logical relationships or consumer preferences. For example, data can be mined to identify market segments or consumer affinities.
- Associations: Data can be mined to identify associations. The beer-diaper example is an example of associative mining.
- Sequential patterns: Data is mined to anticipate behavior patterns and trends. For example, an outdoor equipment retailer could predict the likelihood of a backpack being purchased based on a consumer's purchase of sleeping bags and hiking shoes.

Data mining consists of five major elements:

1) Extract, transform, and load transaction data onto the data warehouse system.
2) Store and manage the data in a multidimensional database system.
3) Provide data access to business analysts and information technology professionals.
4) Analyze the data by application software.
5) Present the data in a useful format, such as a graph or table.

## II. RELATED WORK

**1) Cluster ensembles: A knowledge reuse framework for combining multiple partitions**

**AUTHORS:** A. Strehl and J. Ghosh

This paper introduces the problem of combining multiple partitionings of a set of objects into a single consolidated clustering without accessing the features or algorithms that determined these partitionings. We first identify several

application scenarios for the resultant 'knowledge reuse' framework that we call cluster ensembles. The cluster ensemble problem is then formalized as a combinatorial optimization problem in terms of shared mutual information. In addition to a direct maximization approach, we propose three effective and efficient techniques for obtaining high-quality combiners (consensus functions). The first combiner induces a similarity measure from the partitionings and then reclusters the objects. The second combiner is based on hypergraph partitioning. The third one collapses groups of clusters into meta-clusters which then compete for each object to determine the combined clustering. Due to the low computational costs of our techniques, it is quite feasible to use a supra-consensus function that evaluates all three approaches against the objective function and picks the best solution for a given situation. We evaluate the effectiveness of cluster ensembles in three qualitatively different application scenarios: (i) where the original clusters were formed based on non-identical sets of features, (ii) where the original clustering algorithms worked on non-identical sets of objects, and (iii) where a common data-set is used and the main purpose of combining multiple clusterings is to improve the quality and robustness of the solution. Promising results are obtained in all three situations for synthetic as well as real data-sets.

**2) Evolving clusters in gene-expression data**

**AUTHORS:** E. R. Hruschka, R. J. G. B. Campello, and L. N. de Castro

Clustering is a useful exploratory tool for gene-expression data. Although successful applications of clustering techniques have been reported in the literature, there is no method of choice in the gene-expression analysis community. Moreover, there are only a few works that deal with the problem of automatically estimating the number of clusters in bioinformatics datasets. Most clustering methods require the number k of clusters to be either specified in advance or selected a posteriori from a set of clustering solutions over a range of k. In both cases, the user has to select the number of clusters. This paper proposes improvements to a clustering genetic algorithm that is capable of automatically discovering an optimal number of clusters and its corresponding optimal partition based upon numeric criteria. The proposed improvements are mainly designed to enhance the efficiency of the original clustering genetic algorithm, resulting in two new clustering genetic algorithms and an evolutionary algorithm for clustering (EAC). The original clustering genetic algorithm and its modified versions are evaluated in several runs using six gene-expression datasets in which the right clusters are known a priori. The results illustrate that all the proposed algorithms perform well in gene-expression data, although statistical comparisons in terms of the computational efficiency of each algorithm point out that EAC outperforms the others. Statistical evidence also shows that EAC, is able to outperform a traditional method based on multiple runs of k-means over a range of k.

**3) Exploring forensic data with self-organizing maps**

**AUTHORS:** B. K. L. Fei, J. H. P. Eloff, H. S. Venter, andM. S. Oliver

This paper discusses the application of a self-organizing map (SOM), an unsupervised learning neural network model, to support decision making by computer forensic investigators and assist them in conducting data analysis in a more efficient manner. A SOM is used to search for patterns in data sets and produce visual displays of the similarities in the data. The paper explores how a SOM can be used as a basis for further analysis. Also, it demonstrates how SOM visualization can provide investigators with greater abilities to interpret and explore data generated by computer forensic tools.

**4) Digital forensic text string searching: Improving information retrieval effectiveness by thematically clustering search results**

**AUTHORS:** N. L. Beebe and J. G. Clark

Current digital forensic text string search tools use match and/or indexing algorithms to search digital evidence at the physical level to locate specific text strings. They are designed to achieve 100% query recall (i.e. find all instances of the text strings). Given the nature of the data set, this leads to an extremely high incidence of hits that are not relevant to investigative objectives. Although Internet search engines suffer similarly, they employ ranking algorithms to present the search results in a more effective and efficient manner from the user's perspective. Current digital forensic text string search tools fail to group and/or order search hits in a manner that appreciably improves the investigator's ability to get to the relevant hits first (or at least more quickly). Thisresearch proposes and empirically tests the feasibility and utility of post-retrieval clustering of digital forensic text string search results – specifically by using Kohonen Self-Organizing Maps, a self-organizing neural network approach.

This paper is presented as a work-in-progress. A working tool has been developed and experimentation has begun. Findings regarding the feasibility and utility of the proposed approach will be presented at DFRWS 2007, as well as suggestions for follow-on research.

**5) Towards an integrated e-mail forensic analysis framework**

**AUTHORS:** R. Hadjidj, M. Debbabi, H. Lounis, F. Iqbal, A. Szporer, and D. Benredjem

Due to its simple and inherently vulnerable nature, e-mail communication is abused for numerous illegitimate purposes. E-mail spamming, phishing, drug trafficking, cyber bullying, racial vilification, child pornography, and sexual harassment are some common e-mail mediated cyber crimes. Presently, there is no adequate proactive mechanism for securing e-mail systems. In this context, forensic analysis plays a major role by examining suspected e-mail accounts to gather evidence to prosecute criminals in a court of law. To accomplish this task, a forensic investigator needs efficient automated tools and techniques to perform a multi-staged analysis of e-mail ensembles with a high degree of accuracy, and in a timely fashion. In this article, we present our e-mail forensic analysis software tool, developed by integrating existing state-of-the-art statistical and machine-learning techniques complemented with social networking techniques. In this framework we incorporate our two proposed authorship attribution approaches; one is presented for the first time in this article.

## III. SYSTEM ANALYSIS

**Existing System**

Clustering algorithms are typically used for exploratory data analysis, where there is little or no prior knowledge about the data. This is precisely the case in several applications of Computer Forensics, including the one addressed in our work. From a more technical viewpoint, our datasets consist of unlabeled objects—the classes or categories of documents that can be found are a priori unknown. Moreover, even assuming that labeled datasets could be available from previous analyses, there is almost no hope that the same classes (possibly learned earlier by a classifier in a supervised learning setting) would be still valid for the upcoming data, obtained from other computers and associated to different investigation processes. More precisely, it is likely that the new data sample would come from a different population. In this context, the use of clustering algorithms, which are capable of finding latent patterns from text documents found in seized computers, can enhance the analysis performed by the expert examiner. The rationale behind clustering algorithms is that objects within a valid cluster are more similar to each other than they are to objects belonging to a different cluster. Thus, once a data partition has been induced from data, the expert examiner might initially focus on reviewing representative documents from the obtained set of clusters. Then, after this preliminary analysis, (s) he may eventually decide to scrutinize other documents from each cluster. By doing so, one can avoid the hard task of examining all the documents (individually) but, even if so desired, it still could be done.

**Proposed System**

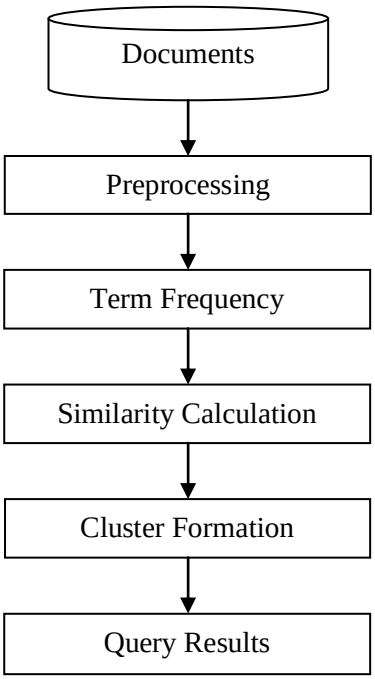
Clustering algorithms have been studied for decades, and the literature on the subject is huge. Therefore, we decided to choose a set of (six) representative algorithm in order to show the potential of the proposed approach, namely: the partitional K-means and K-medoids, the hierarchicalSingle/Complete/Average Link, and the cluster ensemble algorithm known as CSPA. These algorithms were run with different combinations of their parameters, resulting in sixteen different algorithmic instantiations. Thus, as a contribution of our work, we compare their relative performances on the studied application domain—using five real-world investigation cases conducted by the Brazilian Federal Police Department. In order to make the comparative analysis of the algorithms more realistic, two relative validity indexes have been used to estimate the number of cluster automatically from data.

System Design

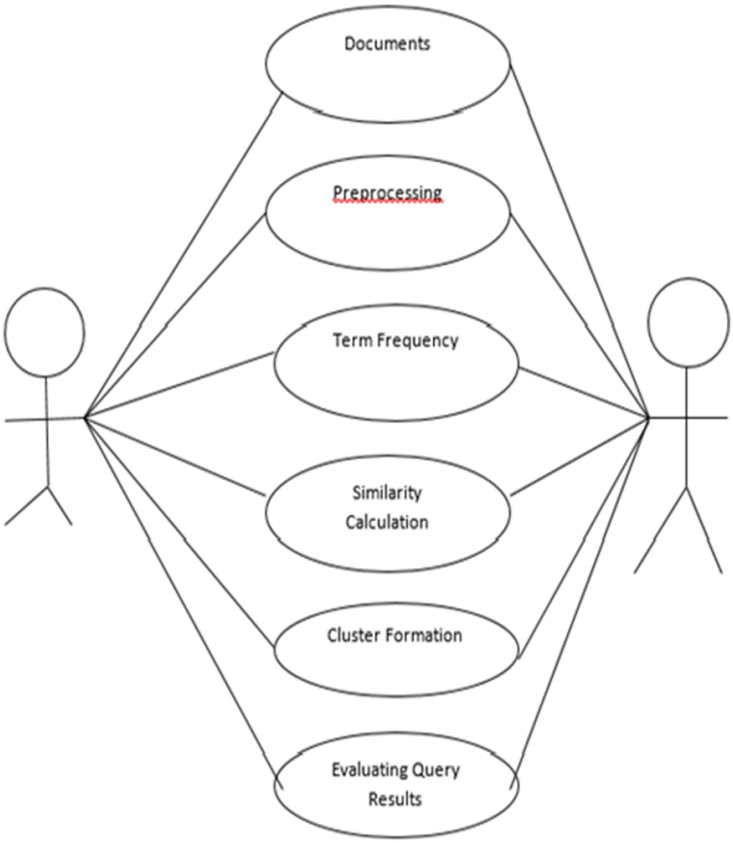
Data Flow Diagram /Use Case Diagram/Flow Diagram

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

**Data Flow Diagram**






## IV. MODULES

### *A. Preprocessing Module:*

Before running clustering algorithms on text datasets, we performed some preprocessing steps. In particular, stopwords (prepositions, pronouns, articles, and irrelevant document metadata) have been removed. Also, the Snow balls temming algorithm for Portuguese words has been used. Then, we adopted a traditional statistical approach for text mining, in which documents are represented in a vector space model. In this model, each document is represented by a vector containing the frequencies of occurrences of words, which are defined as delimited alphabetic strings, whose number of characters is between 4 and 25. We also used a dimensionality reduction technique known as Term Variance (TV) thatcan increase both the effectiveness and efficiency of clustering algorithms. TV selects a number of attributes (in our case 100 words) that have the greatest variances over the documents. In order to compute distances between documents, two measures have been used, namely: cosine-based distance and Levenshtein-based distance. The later has been used to calculate distances between file (document) names only.

### *B. Calculating the number of Clusters:*

In order to estimate the number of clusters, a widely used approach consists of getting a set of data partitions with different numbers of clusters and then selecting that particular partition that provides the best result according to a specific quality criterion (e.g., a relative validity index). Such a set of partitions may result directly from a hierarchical clustering dendrogram or, alternatively, from multiple runs of a partitional algorithm (e.g., K-means) starting from different numbers and initial positions of the cluster prototypes.

### *C. Clustering Techniques:*

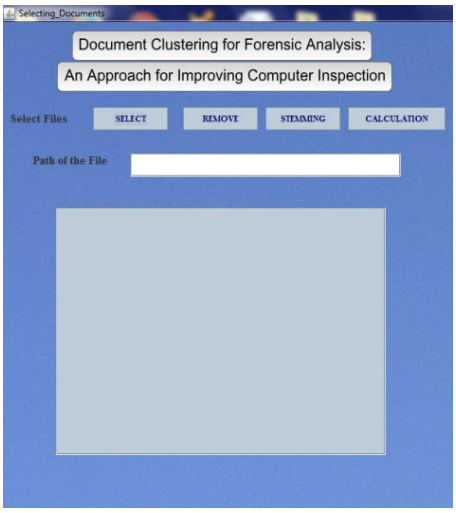
The clustering algorithms adopted in our study—the partitional K-means and K-medoids, the hierarchical Single/Complete/Average Link, and the cluster ensemble based algorithm known as CSPA—are popular in the machine learning and data mining fields, and therefore they have been used in our study. Nevertheless, some of our choices regarding their use deserve further comments. For instance, K-medoids is similar to K-means. However, instead of computing centroids, it uses medoids, which are the representative objects of the clusters. This property makes it particularly interesting for applications in which (i) centroids cannot be computed; and (ii) distances between pairs of objects are available, as for computing dissimilarities between names of documents with the Levenshtein distance.

### *D. Removing Outliers:*

We assess a simple approach to remove outliers. This approach makes recursive use of the silhouette. Fundamentally, if the best partition chosen by the silhouette has singletons (i.e., clusters formed by a single object only), these are removed. Then, the clustering process is repeated over and over again—until a partition without singletons is found. At the end of the process, all singletons are incorporated into the resulting data partition (for evaluation purposes) as single clusters.

**Snapshot**




## V. CONCLUSION

We presented an approach that applies document clustering methods to forensic analysis of computers seized in police investigations. Also, we reported and discussed several practical results that can be very useful for researchers and practitioners of forensic computing. More specifically, in our experiments the hierarchical algorithms known as Average Link and Complete Link presented the best results. Despite their usually high computational costs, we have shown that they are particularly suitable for the studied application domain because the dendrograms that they provide offer summarized views of the documents being inspected, thus being helpful tools for forensic examiners that analyze textual documents from seized computers. As already observed in other application domains, dendrograms provide very informative descriptions and visualization capabilities of data clustering structures [5].

The partitional K-means and K-medoids algorithms also achieved good results when properly initialized. Considering the approaches for estimating the number of clusters, the relative validity criterion known as silhouette has shown to simplified version. In addition, some of our results suggest that using the file names along with the document content information may be useful for cluster ensemble algorithms. Most importantly, we observed that clustering algorithms indeed tend to induce clusters formed by either relevant or irrelevant documents, thus contributing to enhance the expert examiner's job. Furthermore, our evaluation of the proposed approach in five real-world applications show that it has the potential to speed up the computer inspection process. Aimed at further leveraging the use of data clustering algorithms in similar applications, a promising venue for future work involves investigating automatic approaches for cluster labeling. The assignment of labels to clusters may enable the expert examiner to identify the semantic content of each cluster more quickly—eventually even before examining their contents. Finally, the study of algorithms that induce overlapping partitions (e.g., Fuzzy C-Means and Expectation-Maximization for Gaussian Mixture Models) is worth of investigation.

## REFERENCES

[1] 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.

[2] B. S. Everitt, S. Landau, and M. Leese, Cluster Analysis. London, U.K.: Arnold, 2001.

[3] A. K. Jain and R. C. Dubes, Algorithms for Clustering Data. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[4] L. Kaufman and P. Rousseeuw, Finding Groups in Gata: An Introduction to Cluster Analysis. Hoboken, NJ: Wiley-Interscience, 1990.

[5] R. Xu and D. C.Wunsch, II, Clustering. Hoboken, NJ: Wiley/IEEE Press, 2009.

[6] 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.

[7] 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.

[8] B. K. L. Fei, J. H. P. Eloff, H. S. Venter, andM. S. Oliver, "Exploring forensic data with self-organizing maps," in Proc. IFIP Int. Conf. Digital Forensics, 2005, pp. 113–123.

[9] 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.

[10] 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.

[11] F. Iqbal, H. Binsalleeh, B. C. M. Fung, and M. Debbabi, "Mining writeprints from anonymous e-mails for forensic investigation," Digital Investigation, Elsevier, vol. 7, no. 1–2, pp. 56–64, 2010.

[12] S. Decherchi, S. Tacconi, J. Redi, A. Leoncini, F. Sangiacomo, and R. Zunino, "Text clustering for digital forensics analysis," Computat. Intell. Security Inf. Syst., vol. 63, pp. 29–36, 2009.