ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 7, July 2014

# CLUSTERING OF DOCUMENTS IN FORENSIC ANALYSIS FOR IMPROVING COMPUTER INSPECTION

**K.Pallavi[1], S.NagarjunaReddy[2], Dr.S.Sai Satyanarayana Reddy[3]**

M.Tech, CSE, LBRCE, Mylavaram, India[1]

Assistant Professor, CSE, LBRCE, mylavaram, India[2]

Professor, CSE, LBRCE, Mylavaram, India[3]

**Abstract:** In Forensic Analysis thousands of files are usually examined. Data in those files consists of unstructured text analyzing it by examiners is very difficult. Algorithms for clustering documents can facilitate the discovery of new and useful knowledge from the documents under analysis. Cluster analysis itself is not one specific algorithm but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster. Here we propose an approach that applies to clustering of Documents seized in police investigations. We define the proposed approach with K-Means algorithm. Our experiment shows that the performance of computer for inspecting several files is improved. Finally we also present and discuss several practical results that can be useful for researchers and practitioners of forensic computing.

**Index Terms:** Clustering, forensic computing, mining.

## 1. INTRODUCTION

The volume of data in the digital world increased from 161 hexabytes in 2006 to 988 hexabytes in 2010.this large amount of data has a direct impact in *computer Forensics.* In our particular application domain it usually involves examining hundreds of thousands of files per computer. This activity exceeds the expert's ability of analysis and interpretation of data. Therefore methods for automated data analysis. Like those widely used for machine learning and data mining are of paramount importance. In particular, algorithms for pattern recognition from the information present in text documents are promising as it will hopefully become evident later in the paper. The concept of clustering has been around for a long time. It has several applications, particularly in the context of information retrieval and in organizing web resources. The main purpose of clustering is to locate information and in the present day context, to locate most relevant electronic resources. The research in clustering eventually led to automatic indexing --- to index as well as to retrieve electronic records. Clustering is a method in which we make cluster of objects that are somehow similar in characteristics. The ultimate aim of the clustering is to provide a grouping of similar records. Clustering is often confused with classification, but there is some difference between the two.

Clustering algorithms are typically used for exploratory data analysis. 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 labelled datasets could be available from previous analyses. 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 but, even if so desired, it still could be done.

In a more practical and realistic scenario, domain experts are scarce and have limited time available for performing examinations. Thus, it is reasonable to assume that, 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. Such an approach, based on document clustering, can indeed improve the analysis of seized computers, as it will be discussed in more detail later.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 7, July 2014

Clustering algorithms have been studied for decades, and the literature on the subject is huge.

## 2. RELATED WORK

In database management, data clustering is a technique in which, the information that is logically similar is physically stored together. In order to increase the efficiency of search and the retrieval in database management, the number of disk accesses is to be minimized. In clustering, since the objects of similar properties are placed in one class of objects, a single access to the disk can retrieve the entire class. If the clustering takes place in some abstract algorithmic space, we may group a population into subsets with similar characteristic, and then reduce the problem space by acting on only a representative from each subset. Clustering is ultimately a process of reducing a mountain of data to manageable piles. For cognitive and computational simplification, these piles may consist of "similar" items. There are two approaches to document clustering, particularly in information retrieval; they are known as term and item clustering. Term clustering is a method, which groups redundant terms, and this grouping reduces, noise and increase frequency of assignment

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. The literature on *Computer Forensics* only reports the use of algorithms that assume that the number of clusters is known and fixed *a priori* by the user. Aimed at relaxing this assumption, which is often unrealistic in practical applications, a common approach in other domains involves estimating the number of clusters from data. Essentially, one induces different data partitions (with different numbers of clusters) and then assesses them with a relative validity index in order to estimate the best value for the number of clusters.

## 3. CLUSTERING ALGORITHMS AND PRE-PROCESSING STEPS

Clustering is a division of data into groups of similar objects. Representing the data by fewer clusters necessarily loses certain fine details, but achieves simplification. It models data by its clusters. Data modelling puts clustering in a historical perspective rooted in mathematics, statistics, and numerical analysis. From a machine learning perspective clusters correspond to hidden patterns, the search for clusters is unsupervised learning, and the resulting system represents a data concept. From a practical perspective clustering plays an outstanding role in data mining applications such as scientific data exploration, information retrieval and text mining. Clustering is the subject of active research in several fields such as statistics, pattern recognition, and machine learning. This survey focuses on clustering in data mining. Data mining adds to clustering the complications of very large datasets with very many attributes of different types. This imposes unique computational requirements on relevant clustering algorithms. In this paper we use K-Means Algorithm.

### A. K-Means Algorithm:

K-Means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as bary centers of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done.

In other words centroids do not move anymore finally, this algorithm aims at minimizing an *objective function*, in this case a squared error function. The objective function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 ,$$

Where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre $c_j$, is an indicator of the distance of the *n* data points from their respective cluster centres.

The algorithm is composed of the following steps:

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 7, July 2014

Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

Although it can be proved that the procedure will always terminate, the k-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centres. The k-means algorithm can be run multiple times to reduce this effect.k-means is a simple algorithm that has been adapted to many problem domains. As we are going to see, it is a good candidate for extension to work with fuzzy feature vectors. Here is some of the steps for Clustering Of Documents.

### B. Preprocessing

Before running clustering algorithms on text datasets, we performed some preprocessing steps. In particular, stop words (prepositions, pronouns, articles, and irrelevant document metadata) have been removed. Also, the Snow balls stemming 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) that can 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.

### C. 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 dendogram or, alternatively, from multiple runs of a partitional algorithm (e.g., K-means) starting from different numbers and initial positions of the cluster prototypes.

### D. Clustering Techniques

Traditionally clustering techniques are broadly divided into hierarchical and partitioning. Hierarchical clustering is further subdivided into agglomerative and divisive.

a) ***Agglomerative:*** Start with the points as individual clusters and, at each step, merge the most similar or closest pair of clusters. This requires a definition of cluster similarity or distance.
b) ***Divisive:*** Start with one, all-inclusive cluster and, at each step, split a cluster until only singleton clusters of individual points remain.

In this case, we need to decide, at each step, which cluster to split and how to perform the split. Hierarchical techniques produce a nested sequence of partitions, with a single, all inclusive cluster at the top and singleton clusters of individual points at the bottom. Each intermediate level can be viewed as combining two clusters from the next lower level. The result of a hierarchical clustering algorithm can be graphically displayed as tree, called a dendro gram. This tree graphically displays the merging process and the intermediate clusters. For document clustering, this dendrogram provides a taxonomy, or hierarchical index.

### E. 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.

## 4. CONCLUSION

Clustering has a number of applications in every field of life. We are applying this technique whether knowingly or unknowingly in day-to-day life. One has to cluster a lot of thing on the basis of similarity either consciously or unconsciously. Clustering is often one of the first steps in data mining analysis. The partitioned K-means algorithm 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.It identifies groups of related records that can be used as a starting point for exploring further relationships. 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.

## 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: Improvinginformation 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.