International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Information Retrieval of K-Means Clustering For Forensic Analysis

Prachi Gholap[1], Vikas Maral[2]

[1]Pune University, K.J College of Engineering Kondhwa, 411043, Pune, Maharashtra, India

[2]Pune University, K.J College of Engineering Kondhwa, 411043,Pune, Maharashtra, India

Abstract: ***Throughout computer forensic analysis, tons of files regarding information usually are analyzed. Much regarding the results within those people information consists of unstructured data, where examination by way of personal computer analyzers is difficult to be performed. Within this data, intelligent types of analysis usually are regarding good interest. In particular, algorithms for clustering documents could aid the invention of latest in addition to beneficial awareness on the documents below analysis. We provide a technique which does apply report clustering algorithms to be able to forensic examination regarding personal computers arrested within cops' investigations. Most of us show the recommended method by way of doing considerable testing along with 6-8 well-known clustering algorithms (K-means, K-medoids, Single Link, Complete Link, Average Link, in addition to CSPA) put on to 5 real-world datasets obtained from personal computers arrested within real-world investigations. Tests are actually performed with some other combinations of factors, contributing to 16 different instantiations regarding algorithms. Also, a pair of general applicability indexes was utilized to be able to on auto-pilot appraisal the sheer numbers of clusters. Relevant researches inside reading usually are far more constrained when compared with the study. Our own findings show that the Average Link in addition to Complete Link algorithms delivers greatest results for your application domain. In the event that superbly initialized, partitioned algorithms (K-means in addition to K-medoids) can also render to be able to excellent results. Eventually, we provide in addition to analyze many sensible benefits which helps in scientists in addition to practitioners regarding forensic computing.***

Keywords: Clustering algorithms, datasets, forensic computing, text mining

1. Introduction

The volume of data in digital world has seen huge increment in recent few years, and it will continue to grow exponentially. This massive amount data has an immediate impact in Computer Forensics, which is often broadly understood to be the discipline that mixes portions of law and computer science to gather and analyze data from computers in a fashion that is admissible as evidence inside a court of law. Within our particular application domain, it usually involves analyzing tons of files per computer. This activity exceeds the expert's ability of analysis and interpretation of data. Therefore, strategies for automated data analysis, like those widely used for machine learning and data mining, are of paramount importance. Specifically, algorithms for pattern recognition from the data contained in text documents are promising, mainly because it will hopefully become evident later within the paper. Clustering algorithms usually are employed for exploratory data analysis, in which there is a minimum of prior understanding of the info [4], [1]. This is the precise case of applications of Computer Forensics, for example the one addressed inside our work. From a much more technical viewpoint, our datasets consist of unlabeled objects, the classes or groups of documents that are available can be a priori unknown. Moreover, even in the event that labeled datasets could accumulate from previous analyses, there is almost no hope that the same classes (possibly learned earlier by a classifier inside a supervised learning setting) could well be still valid for any upcoming data, from other computers and associated to various investigation processes. More precisely, chances are that the newest data sample would come from an alternative population. On this context, the use of clustering algorithms, which are designed for finding latent patterns from text documents present in seized computers, can boost the analysis done by the expert analyzer. The rationale behind clustering algorithms is the fact objects within a sound cluster are definitely more similar together compared to they are to objects belonging to an alternative cluster [4], [1]. Thus, after a data partition is induced from data, the expert analyzer might initially center on reviewing representative documents on the obtained pair of clusters. Then, so next preliminary analysis, they may eventually opt to examine other documents from each cluster. In so doing, it's possible to stop the hard task of analyzing all the documents (individually) but, even if that's so desired, nonetheless might be done. In a more practical and realistic scenario, domain experts (e.g., forensic analyzers) are scarce and possess very limited time for performing examinations. Thus, it is reasonable to visualize that, after getting a relevant document; the analyzer could prioritize the analysis of other documents from cluster appealing, because it is likely that these types highly relevant to the investigation. Such a blueprint, dependant on document clustering, can indeed help the analysis of seized computers, as it'll be discussed in more detail later.

Actually, we can't even locate one work that is definitely reasonably near the coast its application domain knowing that reports the employment of algorithms efficient at estimating the sheer numbers of clusters. Even perhaps more surprising is the possible lack of studies on hierarchical clustering algorithms, which date back for the sixties. Our study considers such classical algorithms, in addition to recent developments in clustering; just like the make use of consensus partitions [2]. This current paper extends our previous work [9], where nine different instantiations of algorithms were analyzed. As previously mentioned, in this current work we employ sixteen instantiations of algorithms. Furthermore, we provide more insightful quantitative and

qualitative analyses of the experimental results in our application domain.

Directed 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 analyzer to spot the semantic content of every cluster more quickly, eventually even before analyzing their contents. Finally, the analysis of algorithms that induce overlapping partitions (e.g., Fuzzy C-Means and Expectation-Maximization for Gaussian Mixture Models) may be worth of investigation. The remaining paper is organized as follows: Section II shows the algorithms used. The section III presents the different topics related to these studies and their key points. Whereas, the section IV concludes the paper and gives some future studies that are bound to be done in near future.

2. Methods for Clustering

For the purpose of clustering the data, we have to use some clustering algorithms. We have studied 6 different clustering algorithms. They are:

A. *k*-means Algorithms

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 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 bury 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 centers.

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.
4. 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 optima configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centers. 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 are some of the steps for Clustering of Documents.

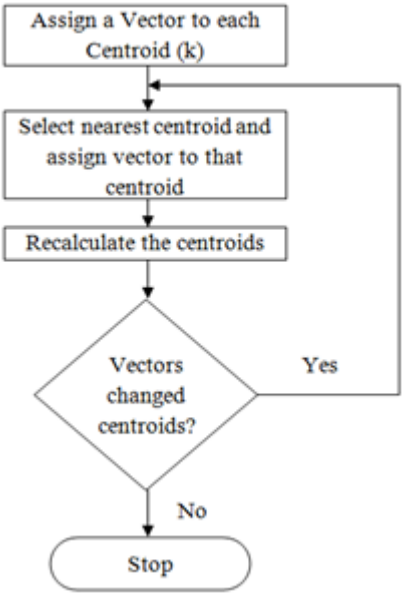

Figure 1: k-means flowchart

B. *k*-medoids Algorithm

The k-Means algorithm has main disadvantage that it is sensitive to outliers since an object with an extremely large value may distort the distribution of data. Instead of taking the mean value of the objects in a cluster as a reference point, a medoid can be used, which is the most centrally located object in a cluster. Thus, the partitioning method can still be performed based on the principle of minimizing the sum of the dissimilarities between each object and its corresponding reference point. This concept forms the basis of the k-Medoids method. The basic strategy of k- Medoids clustering algorithms is to find k clusters in n objects by first arbitrarily finding a representative object (the medoids) for each cluster. Each remaining object is clustered with the medoid to which it is the most similar. The k-Medoids method uses representative objects as reference points instead of taking the mean value of the objects in each cluster. The algorithm takes the input parameter k, the number of clusters to be partitioned among a set of n objects[11]. K-medoid is a classical partitioning technique of clustering that clusters the data set of n objects into k number of clusters. This k: the number of clusters required is to be given by user. This algorithm works on the principle of

minimizing the sum of dissimilarities between each object and its corresponding reference point. The algorithm randomly chooses the k objects in dataset D as initial representative objects called medoids. A medoid can be defined as the object of a cluster, whose average dissimilarity to all the objects in the cluster is minimal i.e. it is a most centrally located point in the given data set. Then for all medoid, after every assignment of a data object to particular cluster the new medoid is decided. The problem is K-medoids does not generate the same result with each run,

Algorithm: Document Clustering: *k*-medoids
Input:
A Collection of Documents $\{D_i\}$,
Number of Representatives *K*.
Output:
A set of medoid documents $D_{C1},\ldots, D_{Ck.}$

1. Randomly select *k* documents as the initial cluster centers.
2. For each document D_i, do, Assign its membership to the cluster C_j that has the largest similarity. sim(D_i, D_{cj});
3. Find the most centrally located document in each cluster.
4. Repeat 2 & 3 till small change in total sum of similarity.
5. Return.

C. Single Link Algorithm

Single Link algorithms uses bottom-up strategy. It compares each point with each point. In this, initially, every object belongs to the different cluster. With iteration, we merge the closest clusters, till some condition is satisfied. Fig (3) explains this algorithm.

- The similarity between a pair of clusters:
- The similarity between the most similar pair of documents, one of which appears in each cluster
- Each cluster member will be more similar to at least one member in that same cluster than to any member of another cluster
- Single-link clustering tends to produce a small number of large, poorly linked clusters

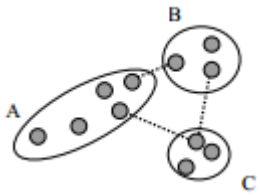

Figure 2: Clustering in single link algorithm

We combine the two clusters whose shortest distance is the smallest: A and B

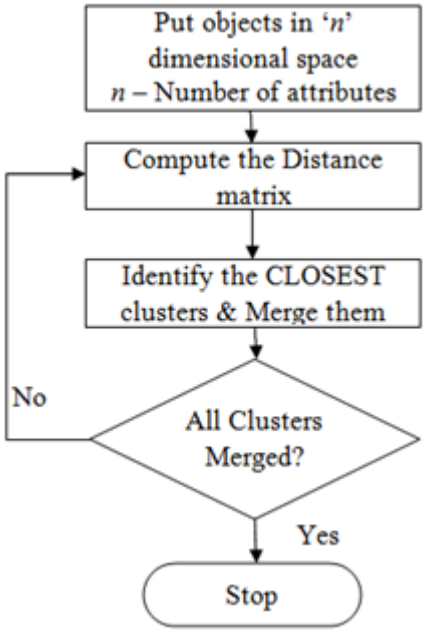

Figure 3: Single Link Algorithm flowchart

D. Complete Link Algorithm

Complete Link algorithm is almost identical to the single link algorithm. The only difference is that, complete link algorithm chooses the distant pair of clusters to merge with iteration. Fig (4) shows this algorithm.

- The similarity between the least similar pair of documents from the two clusters
- Each cluster member is more similar to the most dissimilar member of that cluster than to the most dissimilar member of any other cluster
- Complete-link clustering produces a larger number of small, tightly linked clusters.

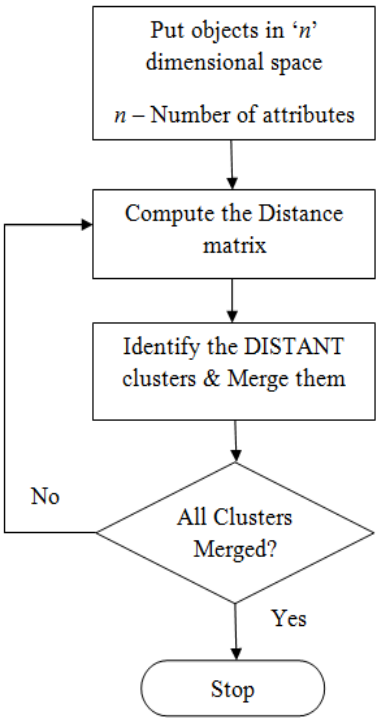

Figure 4: Complete Link algorithm flowchart

We combine the two clusters whose longest distance is the smallest: B and C as shown in the fig (5).

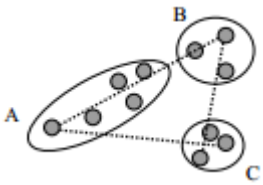

Figure 5: Clustering in Complete Link algorithm

E. Average Link Algorithm

In Average link algorithm, the distance needed to merge the clusters is the average distance between all the objects from one cluster to every object from other cluster. Fig (5) shows the flow of algorithm.

Each cluster member has a greater average similarity to the other members of its cluster than it does to all members of any other cluster

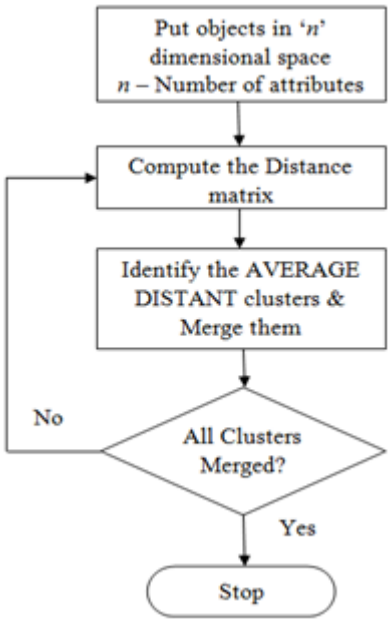

Figure 6: Average Link Algorithm flowchart

F. Cluster-based Similarity Partitioning Algorithm (CSPA)

If two objects belong to the same cluster, they considered as the similar object, if not, then considered as dissimilar. This is the simple logic behind CSPA. In similarity matrix of this algorithm, similarity of object is denoted by 1, and otherwise 0. Thus, for every clustering, an *n*n* binary similarity matrix is created.

3. Literature Review

There are just a couple of studies reporting the utilization of grouping calculations in the Machine Legal sciences field. Basically, the vast majority of the studies portray the utilization of excellent calculations for grouping information e.g., Desire Boost (EM) for unsupervised learning of Gaussian Mixture Models, K-implies, Fuzzyc-implies (FCM), and Orchestrating toward oneself Maps (SOM). These calculations have well-known properties and are generally utilized as a part of practice. For example, K-means and FCM can be seen as specific instances of EM [CM Bishop]. Calculations like SOM [16], in their turn, by and large have inductive inclinations like K-means, however are generally less computationally proficient. In [3], SOM-based calculations were utilized for bunching records with the point of settling on the choice making procedure performed by the analysts more effective. The records were bunched by considering their creation dates/times and their augmentations. This sort of calculation has additionally been utilized as a part of [12] with a specific end goal to bunch the results from essential word looks. The underlying presumption is that the bunched results can build the data recovery productivity, on the grounds that it would not be important to survey all the reports found by the client any longer.

A coordinated environment for mining messages for scientific examination, utilizing order and grouping calculations, was introduced in [13]. In a related application space, messages are gathered by utilizing lexical, syntactic, structural, and area particular gimmicks [7]. Three grouping calculations (K-means, Bisecting K-means and EM) were utilized. The issue of grouping messages for legal investigation was additionally tended to in [15], where a Piece based variation of K-means was connected. They got results were investigated subjectively, and the creators inferred that they are fascinating and helpful from an examination point of view. All the more as of late [8], a FCM-based system for mining affiliation tenets from scientific information was depicted. The writing on Machine Criminology just reports the utilization of calculations that accept that the quantity of groups is known and altered from the earlier by the client. Went for unwinding this presumption, which is frequently improbable in viable applications, a typical approach in different areas includes evaluating the quantity of groups from information. Basically, one affects diverse information segments (with distinctive quantities of groups) and afterward evaluates them with a relative legitimacy record to gauge the best esteem for the quantity of bunches [1], [4], and [11]. This work makes utilization of such routines, consequently conceivably encouraging the work of the master inspector who in practice would scarcely know the quantity of bunches from the earlier.

Clustering algorithms are studied for several years, along with the literature on the subject is huge. Therefore, we thought we would choose some (six) representative algorithms in an effort to show the potential of the proposed approach, namely: the partitioned K-means [1] and K-medoids [10], the hierarchical Single, Complete or Average Link [14], along with the cluster ensemble algorithm referred to as CSPA [2]. These algorithms were run with some other mixtures of their parameters. Thus, like a contribution of our own work, we compare their relative performances to the studied application domain; using five real-world investigation cases conducted with the Brazilian Federal Police Department. So as to make the comparative research into the algorithms more realistic, two relative validity indexes (Silhouette [10] and its particular simplified version [6]) are familiar with estimate the sheer numbers of

clusters automatically from data. It really is well-known that the sheer numbers of clusters is a significant parameter of several algorithms and it also can be quite a priori unknown. In terms of we understand, however, the automated estimation of the sheer numbers of clusters hasn't been investigated within the computer Forensics literature.

4. 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 helpful for researchers and practitioners of forensic computing. More specifically, inside our experiments the hierarchical algorithms referred to as Average Link and Complete Link presented the best results. Despite their usually high computational costs, we demonstrate that they're particularly suitable for the studied application domain as the dendrograms that they offer summarized views of the documents being inspected, thus being helpful tools for forensic analyzers 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 [14]. The partitioned K-means and K-medoids algorithms also achieved accomplishment when properly initialized. Taking into consideration the approaches for estimating the number of clusters, the relative validity criterion referred to as silhouette has proven to be more accurate than its (more computationally efficient) simplified version. Additionally, some of our results suggest that using the file names combined with the document content information may be helpful for cluster ensemble algorithms. Above all, we observed that clustering algorithms indeed tend to induce clusters formed by either relevant or irrelevant documents, thus adding to boost the expert analyzer's job. Furthermore, our evaluation of the proposed approach in five real-world applications shows so it has the potential to speed up the computer inspection process.

Directed 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 analyzer to spot the semantic content of every cluster more quickly, eventually even before analyzing their contents. Finally, the analysis of algorithms that induce overlapping partitions (e.g., Fuzzy C-Means and Expectation-Maximization for Gaussian Mixture Models) may be worth of investigation.

References

[1] A. K. Jain and R. C. Dubes, "Algorithms for Clustering Data", Englewood Cliffs, NJ: Prentice-Hall, 1988.
[2] A. Strehl and J. Ghosh, "Cluster ensembles: A knowledge reuse framework for combining multiple partitions," J. Mach. Learning Res., 2002.
[3] 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.
[4] B. S. Everitt, S. Landau, and M. Leese, "Cluster Analysis", London, U.K.: Arnold, 2001.
[5] C. M. Bishop, "Pattern Recognition and Machine Learning", New York: Springer-Verlag, 2006.
[6] E. R. Hruschka, R. J. G. B. Campello, L. N. de Castro, "Evolving clusters in gene-expression data", Inf. Sci., 2006.
[7] F. Iqbal, H. Binsalleeh, B. C. M. Fung, and M. Debbabi, "Mining writeprints from anonymous e-mails for forensic investigation," Digital Investigation, Elsevier, 2010.
[8] K. Stoffel, P. Cotofrei, and D. Han, "Fuzzy methods for forensic data analysis," in Proc. IEEE Int. Conf. Soft Computing and Pattern Recognition, 2010.
[9] L. F. Nassif and E. R. Hruschka, "Document clustering for forensic computing: An approach for improving computer inspection," in Proc. Tenth Int. Conf. Machine Learning and Applications (ICMLA), 2011,
[10] L. Kaufman and P. Rousseeuw, "Finding Groups in Gata: An Introduction to Cluster Analysis",
[11] Hoboken, NJ: Wiley-Interscience, 1990.
[12] L. Vendramin, R. J. G. B. Campello, and E. R. Hruschka, "Relative clustering validity criteria: A comparative overview," Statist. Anal.Data Mining, 2010.
[13] N. L. Beebe and J. G. Clark, "Digital forensic text string searching: Improving information retrieval effectiveness by thematically clustering search results," Digital Investigation, Elsevier, 2007.
[14] R. Hadjidj, M. Debbabi, H. Lounis, F. Iqbal, A. Szporer, and D. Benredjem, "Towards an integrated e-mail forensic analysis framework," Digital Investigation, Elsevier, 2009.
[15] R. Xu and D. C.Wunsch, II, "Clustering", Hoboken, NJ: Wiley/IEEE Press, 2009.
[16] S. Decherchi, S. Tacconi, J. Redi, A. Leoncini, F. Sangiacomo, and R. Zunino, "Text clustering for digital forensics analysis," Computat.Intell. Security Inf. Syst., 2009.
[17] S. Haykin, "Neural Networks: A Comprehensive Foundation", Englewood Cliffs, 1998.