Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 10, October 2014, pg.903 – 909

RESEARCH ARTICLE

Improving Computer Inspection by Using Forensic Cluster Analysis to Develop the Document

K.ISRAIL[1], C.C.KALYAN SRINIVAS[2]

[1]M.Tech Student, Department of Computer Science & Engineering, KMMITS, Tirupathi, A.P, India
Email: israel.1232@gmail.com

[2]Assistant Professor, Department of Computer Science & Engineering, KMMITS, Tirupathi, A.P, India
Email: kalyan.chenta@gmail.com

Abstract: In present forensic analysis, many bulks of files are usually examined. Data mining is a process of searching through large amounts of data for patterns recognition. It is a relatively new concept which is directly related to computer science. It can be used with a number of older computer techniques such as pattern recognition and statistics. The goal of data mining is to extract important information from data that was not previously known. This paper make use of data mining concept for collecting patient details in hospital management .We use a new algorithm called RENOVATE algorithm to cluster the patient record for citation the disease of the patient. This algorithm cluster the data based on everyday update and produce the better result for the doctor to understand the state of affairs of the patient. The renovate algorithm provide the result by clustering the record on the core of labels. And also gives how the clustering process carried to place them in labeled order.

Index Terms: Clustering, forensic computing, text mining

I. INTRODUCTION

It is estimated that the volume of data in the digital world increased from 161 hexa bytes in 2006 to 988 hexa bytes in2010 [1], about 18 times the amount of information present in all the books ever written and it continues to grow exponentially. This large amount of data has a direct impact in Computer Forensics, which can be broadly defined as the discipline that combines elements of law and computer science to collect and analyze data from computer systems in a way that is admissible as evidence in a court of law. In our particular application domain, it usually involves examining hundreds of thousands offiles 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. Clustering algorithms are typically used for exploratory data analysis, where there is little or no prior knowledge about the data [2], [3]. 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 [2], [3]. 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 de sired, it still could be done. In a more practical and realistic scenario, domain experts (e.g., forensic examiners) are scarce and have limited time available for performing examinations. Clustering algorithms have been studied for decades, and the literature on the subject is huge. Therefore, we decided to choose set of (six) representative algorithms in order to show the potential of the proposed approach, namely: the partitional K-means [3] and K-medoids [4], the hierarchical

TABLE I
SUMMARY OF ALGORITHMS AND THEIR PARAMETERS

Acronmy	Algorithm	Attributes	Distance	Initialization	K-estimate
Kms	K-means	Cont.(all)	Cosine	Random	Simp.Sil.
Kms100	K-means	100>TV	Cosine	Random	Simp.Sil.
Kms100*	K-means	100>TV	Cosine	[18]	Simp.Sil.
KmsT100*	K-means	100>TV	Cosine	Random	Silhouette
KmsS	K-means	Cont.(all)	Cosine	Random	Rec.Sil.
Kms100S	K-means	100>TV	Cosine	Random	Rec.Sil.
Kmd100	K-mediods	100>TV	Cosine	Random	Silhouette
Kmd100*	K-mediods	100>TV	Cosine	[18]	Silhouette
KmdLev	K-mediods	Name	Lev.	Random	Silhouette
KmdLevS	K-mediods	Name	Lev.	Random	Rec.sil.
AL100	Average Link	100>TV	Cosine	-	Silhouette
CL100	Complete Link	100>TV	Cosine	-	Silhouette
SL100	Single Link	100>TV	Cosine	-	Silhouette
NC	CSPA	Name,cont.(all)	CSPA	Random	Simp.Sil
NC100	CSPA	Name,100>TV	CSPA	Random	Simp.Sil
E100	CSPA	Cont100.random	CSPA	Random	Simp.Sil

Single/Complete/Average Link [5] , and the cluster ensemble algorithm known as CSPA [6]. These algorithms were run with different combinations of their parameters, resulting in sixteen different algorithmic instantiations, as shown in Table I. Thus, as a contribution of our work, we compare their relative performances on the studied application domain—using fivereal-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 (Silhouette [4] and its simplified version [7]) have been used to estimate the number of clusters automatically from data. It is well-known that the number of clusters is a critical parameter of many algorithms and it is usually a priori unknown. As far as we know, however, the automatic estimation of the number of clusters has not been investigated in the Computer Forensics literature. Actually, we could not even locate one work that is reasonably close in its application domain and that reports the use of algorithms capable of estimating the number of clusters. Perhaps even more surprising is the lack of studies on hierarchical clustering algorithms, which date back to the sixties. Our study considers such classical algorithms, as well as recent developments in clustering, such as the use of consensus partitions [6]. The present paper extends our previous work , where nine different instantiations of algorithms were analyzed.

As previously mentioned, in our current work we employ sixteen instantiations of algorithms. In addition, we provide more in sightful quantitative and qualitative analyses of their experimental results in our application domain. The remainder of this paper is

organized as follows. Section II presents related work. Section III briefly addresses the adopted clustering algorithms and preprocessing steps. Section IV reports our experimental results, and Section V addresses some limitations of our study. Finally, Section VI concludes the paper.

II. RELATED WORK

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 [21]. Algorithms like SOM [22], in their turn, generally have inductive biases similar to K-means, but are usually less computationally efficient.

In [8], 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 [9] 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 [10]. In a related application domain, e-mails are grouped by using lexical, syntactic, structural, and domain-specific features [11]. Three clustering algorithms (K-means, Bisecting K-means and EM) were used. The problem of clustering-mails for forensic analysis was also addressed in [12], 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 described. 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 [2], [3], [14]. This work makes use of such methods, thus potentially facilitating the work of the expert examiner—who in practice would hardly know the number of clusters a priori.

III. CLUSTERING ALGORITHMS AND REPROCESSING

A. Pre-Processing Steps:

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 Snowball 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 [15]. 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) [16] that can increase both the effectiveness and efficiency of clustering algorithms. TV selects a number of attributes (in our case 100words) that have the greatest variances over the documents. In order to compute distances between documents, two measures have been used, namely: cosine-based distance [15] and Levenshtein- based distance [17]. The later has been used to calculate distances between file (document) names only.

B. Estimating the Number of Clusters from Data:

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 [2]–[5]). 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 (e.g., see [14] and references there in).For the moment, let us assume that a set of data partition swith different

numbers of clusters is available, from which we want to choose the best one according to some relative validity criterion. Note that, by choosing such a data partition, we are performing model selection and, as an intrinsic part of this process, we are also estimating the number of clusters.

A widely used relative validity index is the so-called silhouette[4], which has also been adopted as a component of the algorithms employed in our work. Therefore, it is helpful to define it even before we address the clustering algorithms used in our study. Let us consider an objecti belonging to cluster **A**. The average dissimilarity of to all other objects of **A** is denoted byα(i)Now let us take into account cluster **C**. The average dissimilarity of to all objects of **C** will be called d(i, **C**).Aftercomputing d(i, **C**).for all clusters **C**≠**A**, the smallest one is selected, i.e. $b(i) = \mathrm{mind}(i, \mathbf{C}),\ \mathbf{C} \neq \mathbf{A}$This value represents the dissimilarity of toi its neighbor cluster, and the silhouettefor a give object,$s(i)$, is given by:

$$s(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}} \qquad (1)$$

C. Clustering Algorithms:

The clustering algorithms adopted in our study—the partitional K-means [2] and K-medoids [4], the hierarchical Single/Complete/Average Link [5], and the cluster ensemble based algorithm known as CSPA [6]—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 [4] 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 [17].Considering the partitional algorithms, it is widely known that both K-means and K-medoids are sensitive to initialization and usually converge to solutions that represent local minima. Trying to minimize these problems, we used a nonrandom initialization in which distant objects from each other are chosen as starting prototypes [18]. Unlike the partitional algorithms such as K-means/medoids, hierarchical algorithms such as Single Complete/Average Link provide a hierarchical set of nested partitions [3], usually represented in the form of a dendrogram, from which the best number of clusters can be estimated. In particular, one can assess the quality of every partition represented by the dendrogram, subsequently choosing the one that provides the best results [14].

D. Dealing with 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. Table I summarizes the clustering algorithms used in our work and their main characteristics.

IV. EXPERIMENTAL EVALUATION

A. Datasets:

Sets of documents that appear in computer forensic analysis applications are quite diversified. In particular, any kind of content that is digitally compliant can be subject to investigation. Such documents have been originally created in different file formats, and some of them have been corrupted or are actually incomplete in the sense that they have been (partially) recovered from deleted data. We used five datasets obtained from real-world investigation cases conducted by the Brazilian Federal Police Department. Each dataset was obtained from a different hard drive, being selected all the non duplicate documents with extensions “doc”, “docx”, and “odt”. Subsequently, those documents were converted into plain text format and preprocessed as described in Section III-A. The obtained data partitions were evaluated by taking into account that we have a reference partition (ground truth) for every dataset. Such reference partitions have been provided by an expert examiner from the Brazilian Federal Police Department, who previously

inspected every document from our collections. The datasets contain varying amounts of documents (**N**) , groups(**K**) , attributes(**D**) , singletons(**S**) , and number of documents per group (#), as reported in Table II.

TABLE II
DATASET CHARACTERISTICS1

Dataset	N	K	D	S	#Largest cluster
A	37	23	1744	12	3
B	111	49	7894	28	12
C	68	40	2699	24	8
D	74	38	5094	26	17
E	131	51	4861	31	44

B. Evaluation Measure:

From a scientific perspective, the use of reference partitions for evaluating data clustering algorithms is considered a principled approach. In controlled experimental settings, reference partitions are usually obtained from data generated synthetically according to some probability distributions. From a practical standpoint, reference partitions are usually obtained in a different way, but they are still employed to choose a particular clustering algorithm that is more appropriate for a given application, or to calibrate its parameters. Which measures the agreement between a partition **R** , obtained from running a clustering algorithm, and the reference partition **R** given by the expert examiner. More specifically ARI ϵ [0, 1], and the greater its value the better the agreement between **P** and **R**.

C. Results and Discussions:

Table III summarizes the obtained ARI results for the algorithms listed in Table I. In general, AL100 (Average Link algorithm using the 100 terms with the greatest variances, cosine-based similarity, and silhouette criterion) provided the best results with respect to both the average and the standard deviation, thus suggesting great accuracy and stability. Note also that an ARI value close to 1.00 indicates that the respective partition is very consistent with the reference partition—this is precisely the case here. In this table, we only report the best obtained results for the algorithms that search for a consensus partition between file name and content (NC100 and NC)—i.e., partitions whose weights for name/content resulted in the greatest ARI value. The ARI values for CL100 are similar to those found byAL100. Single Link (SL100), by its turn, presented worse results than its hierarchical counterparts—especially for datasets A and B. This result can be explained by the presence of outliers, whose chain effect is known to impact Single Link performance [2].

The results achieved by Kmd100* and Kms100* were also very good and competitive to the best hierarchical algorithms (AL100 and CL100).

TABLE III
ADJUSTED RAND INDEX (ARI) RESULTS

Alg./Dataset	A	B	C	D	E	Mean	σ
AL100	0.94	0.83	0.89	0.99	0.90	0.91	0.06
CL100	0.94	0.76	0.89	0.98	0.90	0.89	0.08
KmsT100*	0.81	0.76	0.89	0.97	0.94	0.88	0.09
Kmd100*	0.81	0.76	0.89	0.96	0.93	0.87	0.08
SL100	0.54	0.63	0.90	0.98	0.88	0.79	0.19
NC100	0.66	0.64	0.78	0.74	0.72	0.71	0.06
Kms	0.61	0.60	0.69	0.79	0.84	0.71	0.11
NC	0.61	0.60	0.69	0.79	0.84	0.71	0.11
Kms100*	0.53	0.63	0.63	0.68	0.93	0.68	0.15
Kmd100	0.81	0.58	0.72	0.25	0.79	0.63	0.23
Kms100	0.64	0.64	0.78	0.29	0.72	0.62	0.19
KmsS	0.47	0.11	0.75	0.80	0.82	0.59	0.30
Kms100S	0.60	0.54	0.74	0.20	0.69	0.55	0.21
E100	0.61	0.10	0.29	0.76	0.08	0.37	0.31
KmdLevS	0.62	0.23	0.37	0.55	0.05	0.36	0.23
KmdLev	0.46	0.16	0.32	0.74	0.08	0.35	0.26

TABLE IV
EXAMPLE OF THE INFORMATION FOUND IN THE CLUSTERS

Cluster	Information
C_1	3 blank documents
C_2	4 financial transactions
C_3	2 maternity payments
C_4	2 grocery lists
C_5	1 foreign exchange transaction warning 1 list of documents for registration information
C_6	2 documents from foreign exchange operations
C_7	1 registration form from a brokerage company 1 contract template from the broker
C_8	1 investment club status 1 agreement for joining the club
C_9	2 models for handling cash greater than R$ 100k
C_{10}	8 receipts of foreign exchange insurance transactions
C_{11}	2 warnings about foreign brokerage business hours
C_{12}	3 label designs of a brokerage company
C_{13}	1 notice about working hours 1 check receipt
C_{14}	2 daily reports from buying/selling exchanges
C_{15}	2 sample documents from office application

Partitions, which are the inputs for the computation of the co-association matrix. Thus, a misleading consensus clustering is obtained. Therefore, the choice of random sets of attributes to generate partitions for consensus clustering algorithms seems to be an inappropriate approach for such text data. Considering the algorithms that recursively apply the Silhouette for removing singletons (KmsS and Kms100S), Table III shows that their results are relatively worse when compared to the related versions that do not remove singletons (Kms andKms100). However, KmdLevS, which is based on the similarities between file names, presented similar results to thosefound by its related version that does not remove singletons(KmdLev). In principle, one could expect that the removalof outliers, identified from carefully analyzing the singletons,could yield to better clustering results.

Compared to Kms and KmsS, the worse results obtained from featureselection by Kms100 and Kms100S, especially in the datasetD, are likely due to k-means convergence to local optima frombad initialization. By considering all the results obtained fromfeature selection, we believe that it should be further studiedmainly because of the potentially advantageous computationalefficiency gains.Finally, from a practical viewpoint, a variety of relevant findingsemerged from our study. It is worth stressing that, for allthe investigated datasets, the best data partitions are formed byclusters containing either relevant or irrelevant documents.
Forexample, in dataset C, the algorithm AL100 obtained a datapartition formed by some singletons and by other 15 clusters($C_1, C_2, \ldots\ldots, C_{15}$) whose information are listed in Table IV.

V. LIMITATIONS

It is well-known that the success of any clustering algorithm is data dependent, but for the assessed datasets some of our adaptations of existing algorithms have shown to be good enough. Scalability may be an issue, however.

More precisely, and aimed at circumventing computational difficulties, partitional clustering algorithms can be used to compute a hierarchical clustering solution by using repeated cluster bi-sectioning approaches. For instance, bi-secting k-means has relatively low computational requirements, i.e., it is $O(N.\log N)$, versus the overall time complexity of $O(N^2.\log N)$ for agglomerative methods. Since the inductivebiases of bisecting k-means and the hierarchical algorithmsused in our work are similar, we believe that, if the number of documents is prohibitively high for running agglomerative algorithms, then bisecting k-means and related approaches can be used.

Considering the computational cost of estimating the number of clusters, the silhouette proposed in [4] depends on the computation of all distances between objects, leading to an estimated computational cost of $O(N^2.D)$, where is N the number of objects in the dataset and 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 [7] 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

$O(N^2.D)$to O(K,N,D) , where , the number of clusters, is usually significantly less than . 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. Finally, as a cautionary note, we would like to mention that, in practice, it is not always of paramount importance to have scalable methods. In our particular application scenario, there are no hard time constraints to get data partitions (like those present when analyzing streaming data with online algorithms).Instead, domain experts can usually spend months analyzing their data before reaching a conclusion.

VI. CONCLUSION

In this paper we have used the label based clustering, to find the exact disease of the patient and this clustering is done as soon as the update is made in the database it will provide us the current status of the patient and the treatment they are suppose to undergo. When the clustering process is done using labels it will produce the exact result which help us to make fast and correct decision about treatment to be given to the patients.

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 through2010," 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. EnglewoodCliffs, NJ: Prentice-Hall, 1988.

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

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

[6] A. Strehl and J. Ghosh, "Cluster ensembles: A knowledge reuse frame work 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. DigitalForensics, 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, "Miningwriteprints from anonymous e-mails for forensic investigation," DigitalInvestigation, Elsevier, vol. 7, no. 1–2, pp. 56–64, 2010.

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

[13] K. Stoffel, P. Cotofrei, and D. Han, "Fuzzy methods for forensic data analysis," in Proc. IEEE Int. Conf. Soft Computing and Pattern Recognition,2010, pp. 23–28.

[14] L. Vendramin, R. J. G. B. Campello, and E. R. Hruschka, "Relative clustering validity criteria: A comparative overview," Statist. Anal.Data Mining, vol. 3, pp. 209–235, 2010.

[15] G. Salton and C. Buckley, "Term weighting approaches in automatic text retrieval," Inf. Process. Manage., vol. 24, no. 5, pp. 513–523, 1988.54 IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 8, NO. 1, JANUARY 2013

[16] L. Liu, J. Kang, J. Yu, and Z. Wang, "A comparative study on un supervised feature selection methods for text clustering," in Proc. IEEEInt. Conf. Natural Language Processing and Knowledge Engineering,2005, pp. 597–601.

[17] V. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," Soviet Physics Doklady, vol. 10, pp. 707–710,1966.

