



## Cluster for Forensic Investigation: An Advance for Civilizing Processor Check

Chandrakanth P\*  
Department of CSE  
YITS, India

K. Nagendra Rao  
Department of CSE  
CVSCE, India

---

**Abstract---** *The Fast algorithm works in two steps. In the first step, features are separated into clusters by using grid-theoretic clustering methods. In the second step, the most delegate feature that is powerfully related to objective classes is selected from each cluster to form a separation of features. Features in dissimilar clusters are comparatively independent, the clustering-based approach of FAST has a high prospect of producing a separation of useful and independent features. To ensure the good organization of FAST, we assume the capable minimum-spanning tree clustering method. The efficiency and effectiveness of the FAST algorithm are evaluated through an observed lessons. wide-ranging experiments are carried out to evaluate FAST and numerous representative feature selection algorithms, explicitly, FCBF, ReliefF, CFS, Consist, and FOCUS-SF, with admiration to four types of well-known classifiers, namely, the probability-based Naive Bayes, the tree-based C4.5, the instance-based IB1, and the rule-based RIPPER fore and after feature selection. The results, on 35 openly obtainable real-world high dimensional image, microarray, and text data, make obvious that FAST not only produces slighter subsets of features but also improves the performances of the four types of classifiers. We present an approach that applies document clustering algorithms to the forensic examination of computers seized in police investigations. We demonstrate the proposed approach by transportation out wide-ranging experimentation with six well-known clustering algorithms (Kmeans, K-medoids, Single Link, Complete Link, Average Link, and CSPA) functional to five real-world datasets obtained from computers seized in real-world investigations. Experiments have been perform with dissimilar combinations of parameters, consequential in 16 special instantiations of algorithms. In addition, two comparative authority indexes were used to mechanically estimate the number of clusters. Related studies in the literature are considerably more limited than our study. Our experiments show that the standard Link and absolute Link algorithms make available the best results for our submission domain. If correctly initialized, partitional algorithms (K-means and K-medoids) can also surrender to very good outcome. Finally, we also present and converse several sensible results that can be useful for researchers and practitioners of forensic computing.*

**Keywords---** *FAST, Clustering Algorithm, KMeans, K-Meloids*

---

### I. INTRODUCTION

In cluster analysis, graph-theoretic methods have been well studied and used in many applications. Their results have, sometimes, the best concurrence with human presentation. The wide-ranging graph theoretic clustering is simple: subtract a neighborhood graph of instances, then delete any edge in the table that is much longer/shorter (according to some criterion) than its neighbors. The consequence is a forest and each tree in the forest represents a cluster. In our study, we apply graph theoretic clustering methods to features. In particular, we adopt the minimum spanning tree (MST) based clustering algorithms, because they do not presuppose that data points are grouped around centers or separated by a standard arithmetical curve and have been extensively used in perform. Based on the MST method, we recommend a Fast clustering-based feature Selection algorithm (FAST). The FAST algorithm works in two steps. In the first step, features are separated into clusters by using graphtheoretic clustering methods. In the second step, the most representative feature that is powerfully connected to object classes is selected from each cluster to form the final separation of features. Features in different clusters are comparatively self-determining, the clustering-based approach of FAST has a high probability of producing a subset of useful and independent features. The proposed feature subset selection algorithm FAST was experienced upon 35 openly obtainable image, microarray, and text data sets. The investigational results show that, compared with other five different types of feature subset selection algorithms, the proposed algorithm not only reduces the number of features, but also improves the performances of the four well-known different types of classifiers. The rest of the article is prearranged as follows: In Section 2, we illustrate the related works. In Section 3, we present the new feature subset selection algorithm FAST. In Section 4, we report widespread investigational results to support the proposed FAST algorithm. Finally, in Section 5, we recapitulate the present study and draw some conclusions. In a more practical and realistic scenario, domain experts (e.g., forensic examiners) are scarce and have limited time available for the theater examinations. Thus, it is evenhanded to assume that, after discovery 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 examination. Such an approach, based on document clustering, can indeed improve the analysis of seized

computers, as it will be discussed in more detail later. Clustering algorithms have been studied for decades, and the literature on the subject is huge. Therefore, we determined to choose a set of (six) representative algorithms in order to show the impending of the proposed approach, namely: the partitional K-means and K-medoids, the hierarchical Single/Complete/Average Link, and the cluster ensemble algorithm known as CSPA. These algorithms were run with special combinations of their parameters, resulting in sixteen different algorithmic instantiations, as shown in Table I.

TABLE I  
SUMMARY OF ALGORITHMS AND THEIR PARAMETERS

Acronym	Algorithm	Attributes	Distance	Initialization	K-estimate
Kms	<i>K-means</i>	Cont. (all)	Cosine	Random	Simp. Sil.
Kms100	<i>K-means</i>	100 > TV	Cosine	Random	Simp. Sil.
Kms100*	<i>K-means</i>	100 > TV	Cosine	[18]	Simp. Sil.
KmsT100*	<i>K-means</i>	100 > TV	Cosine	[18]	Silhouette
KmsS	<i>K-means</i>	Cont. (all)	Cosine	Random	Rec. Sil.
Kms100S	<i>K-means</i>	100 > TV	Cosine	Random	Rec. Sil.
Kmd100	<i>K-medoids</i>	100 > TV	Cosine	Random	Silhouette
Kmd100*	<i>K-medoids</i>	100 > TV	Cosine	[18]	Silhouette
KmdLev	<i>K-medoids</i>	Name	Lev.	Random	Silhouette
KmdLevS	<i>K-medoids</i>	Name	Lev.	Random	Rec. Sil.
AL100	<i>AverageLink</i>	100 > TV	Cosine	-	Silhouette
CL100	<i>CompleteLink</i>	100 > TV	Cosine	-	Silhouette
SL100	<i>SingleLink</i>	100 > TV	Cosine	-	Silhouette
NC	CSPA	Name, Cont. (all)	CSPA	Random	Simp. Sil.
NC100	CSPA	Name, 100 > TV	CSPA	Random	Simp. Sil.
E100	CSPA	Cont. 100 random	CSPA	Random	Simp. Sil.

100 > TV: 100 attributes (words) that have the greatest variance over the documents  
 Cont. 100 random: 100 randomly chosen attributes from document content  
 Cont. (all): all features from document content  
 Lev.: Levenshtein distance  
 Simp. Sil.: Simplified Silhouette  
 Rec. Sil.: "Recursive" Silhouette  
 \*: Initialization on distant objects  
 Name: file name

Thus, as a payment of our work, we compare their relative performances on the studied application domain—using five real-world examination cases conducted by the Brazilian Federal Police subdivision. In order to make the comparative analysis of the algorithms more realistic, two relative validity indexes (Silhouette and its simplified version) have been used to approximate the number of clusters repeatedly from data. It is well-known that the number of clusters is a critical parameter of many algorithms and it is habitually a priori unknown. As far as we know, conversely, the automatic evaluation of the number of clusters has not been investigated in the Computer Forensics literature. Actually, we could not even locate one work that is practically close in its application domain and that reports the use of algorithms capable of estimating the number of clusters. Perhaps even more shocking 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 compromise partitions. 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 insightful 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 investigational results, and Section V concludes the paper.

## II. RELATED WORK

There are only a few studies that treat the use of clustering algorithms in the Computer Forensics field. Fundamentally, most of the studies describe the use of classic algorithms for clustering data—e.g., Expectation-Maximization (EM) for unsubstantiated 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 fastidious cases of EM. Algorithms like SOM, in their turn, generally have inductive biases similar to K-means, but are usually less computationally efficient. In SOM-based algorithms were used for clustering files with the aim of making the administrative process performed by the examiners more efficient. The files were clustered by taking into account their creation dates/times and their extensions. This kind of algorithm has also been used in order to cluster [3] the results from keyword searches. The underlying postulation is that the clustered results can increase the information retrieval efficiency, because it would not be essential to review all the documents found by the user anymore. An incorporated location for mining e-mails for forensic analysis, using organization and clustering algorithms, was accessible in. In a related application domain, e-mails are grouped by using lexical, syntactic, structural, and domain-specific features. Three clustering algorithms (Kmeans, Bisecting K-means and EM) were used. The problem of clustering e-mails for forensic analysis was also addressed in, where a Kernel-based variant of K-means was applied. The obtained results were analyzed individually, and the authors completed that they are motivating and useful from an investigation perspective. More recently, a FCM-based method [1] 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 peace with this assumption, which is often improbable 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, This work makes use of such methods, thus potentially facilitating the work of the expert examiner—who in perform would hardly know the quantity of clusters a priori.

### III. CLUSTERING ALGORITHMS AND PREPROCESSING

#### A. Pre-Processing Steps

Before running clustering algorithms on text datasets, we performed some preprocessing steps. In particular, stop words (prepositions, pronouns, articles, and immaterial 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. In this model, each document is represented by a vector containing the frequencies of occurrences of words, which are defined as surrounded 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 procedures have been used, namely: cosine-based distance and Leven shtein-based di tance. The later has been used to calculate distances between file (document) names only.

#### B. Clustering Algorithms

The clustering algorithms adopted in our study—the partitional K-means and Kmedoids 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. 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 initiali tion in which distant objects from each other are chosen as starting prototypes. Unlike the partitional algorithms such as K-means/medoids, hierarchical algorithms [4] such as Single/ Complete/Average Link provide a hierarchical set of nested partition, usually represented in the form of a dendrogram, from which the best number of clusters can be expected. In particular, one can assess the quality of every separation represented by the dendrogram, subsequently choosing the one that provides the best results. The CSPA algorithm fundamentally finds a agreement clustering, from a cluster collection formed by a set of different data partitions. More precisely, after applying clustering algorithms to the data, a similarity (coassociation) matrix is computed. Each element of this matrix represents pair-wise similarities between objects. The similarity between two objects is simply the fraction of the clustering solutions in which those two objects lie in the same cluster. Later, this similarity measure is used by a c stering algorithm that can process a proximity matrix—e.g., K-medoids—to produce the final consensus clustering. The sets of data partitions (clusterings) were generated in two different ways: (a) by operation Kmeans 100 times with different subsets of attributes (in this case CSPA processes 100 data partitions); and (b) by using only two data partitions, namely: one obtained by Kmedoids from the dissimilarities between the file names, and another partition achieved with K-means from the vector space model. In this case, each partition can have different weights, which have been varied between 0 and 1 (in increments of 0.1 and keeping their sum equals to 1). For the hierarchical algorithms (Single/Complete/Average Link), we simply run them and then assess every partition from the consequential dendrogram [7] by means of the silhouette. Then, the best partition (elected according to the relative validity index) is taken as the result of the clustering process. For each partitional algorithm (K-means/medoids), we plement it repeatedly for an mounting number of clusters. For each value of K, a number of partitions achieved from different initializations are assessed in order to choose the best value of K and its equivalent data partition, using the Silhouette and its simplified version, which showed good results in and is more computationally efficient. In our experiments, we assessed all possible values of K in the interval [2,N] where is the number of objects to be clustered.

### 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. In the datasets assessed in our study, for instance, there are textual documents written in different languages (Portuguese and English). Such documents have been formerly created in different file formats, and some of them have been corrupted or are essentially incomplete in the sense that they have been (partially) recovered from deleted data. We used five datasets obtained from real-world examination 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”. consequently, 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 (gr nd 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 untrustworthy amounts of documents (N), groups (K), attributes(D), singletons (S), and number of documents per group (#), as reported in Table II.

TABLE II  
DATASET CHARACTERISTICS<sup>1</sup>

Dataset	<i>N</i>	<i>K</i>	<i>D</i>	<i>S</i>	# Largest cluster
A	37	23	1744	12	3
B	111	49	7894	28	12
C	68	40	2699	24	8
D	74	38	5095	26	17
E	131	51	4861	31	44

**B. Evaluation Measure**

From a scientific perception, 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 unnaturally 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 exacting clustering algorithm that is more appropriate for a given application, or to standardize its parameters. In our case, reference partitions were constructed by a domain expert and reflects the expectations that (s)he has about the clusters that should be found in the datasets. In this sense, the estimate method that we used to assess the obtained data partitions is based on the Adjusted Rand Index, which measures the agreement between a partition P , obtained from running a clustering algorithm, and the reference partition R given by the expert examiner. More specifically , and the reater 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 orientation 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 ARIvalue. The ARI values for CL100 are similar to those found By AL100. 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 ffect is known to impact Single Link presentation. The results achieved by and were also very good and ready for action to the best hierarchical algorithms (AL100 and CL100). We note that, as expected, a Fig. 1.

TABLE III  
ADJUSTED RAND INDEX (ARI) RESULTS

Alg./Dataset	A	B	C	D	E	Mean	$\sigma$
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

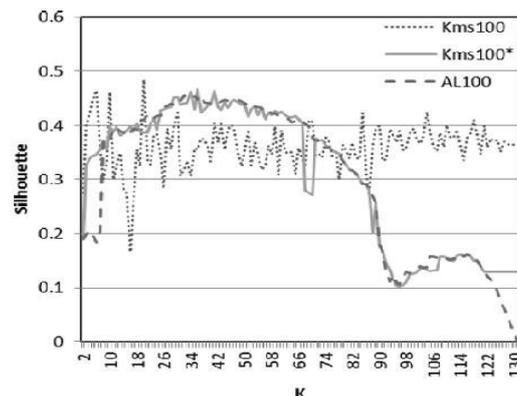


Fig 1. Silhouettes for Kms100, AL100, and Kms100 (dataset E).

Silhouettes for Kms100, AL100, and Kms100 (dataset E). good initialization method such as the one described in provides the best results. Particularly, the initialization on distant objects can minimize the K-means/medoids problems with respect to local minima. To illustrate this aspect, Fig. 1 shows the Silhouette values as a function of K for three algorithms . One can observe that Kms100 (with random initialization of prototypes), presents more local maxima for the Silhouette (recall that these were obtained from local minima of K-means), yielding to less stable results. Conversely , (initialization on distant objects ) has fewer local maxima, being more stable. This trend has also been observed in the other datasets. astoundingly, has curves similar to those of AL100, especially for higher values  $f$  . This fact can be explained, in part, because both algorithms tend to separate outliers. It can also be observed that got slightly better results than its variant with random initialization of prototypes (Kms100).

## V. CONCLUSIONS AND FUTURE WORK

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 purposely, in our experiments the hierarchical algorithms known as Average Link and Complete Link accessible the best results. Despite their usually high computational costs, we have shown that they are predominantly 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 experimental in other application domains, dendrograms provide very useful descriptions and mental picture capabilities of data clustering structures [5]. The partitional K-means and K-medoids algorithms also achieved good results when properly initialized. making an allowance for the approaches for estimating the number of clusters, the family member validity criterion known as silhouette has shown to be more accurate than its (more computationally efficient) simplified version. In addition, some of our results propose that using the file names along with the document content in sequence may be useful for cluster assembly algorithms. Most importantly, we observed that clustering algorithms indeed tend to induce clusters formed by either relevant or unrelated documents, thus causative 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 examination process. Aimed at further leveraging the use of data clustering algorithms in similar applications, a talented venue for future work involves investigating automatic approaches for cluster classification. The obligation of labels to clusters may enable the expert examiner to identify the semantic content of each cluster more quickly—eventually even before investigative 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 examination.

## 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] 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] 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.