RAIRO-Oper. Res. 50 (2016) 951–963
DOI: 10.1051/ro/2015057

RAIRO Operations Research
www.rairo-ro.org

CLUSTERING OF OPTIMIZED DATA FOR EMAIL FORENSICS

Dhai Eddine Salhi[1], Abdelkamel Tari[1] and M-Tahar Kechadi[2]

Abstract. Forensics is a study of evidence to help the police solving crimes. If we apply (Forensics) in Computer Sciences domain, crimes are mainly network attacks found more in emails; which become nowadays the most popular way of communication accessible *via* Internet. We receive in our Inboxes emails gangs without being aware of them. Therefore, it is necessary to build an automatic checking system to filter good emails from bad ones. In this paper, we propose a new emails processing approach using Singular Value Decomposition method (SVD) to optimize emails data before applying Data Mining techniques (Clustering) to extract bad emails located in the mail servers where the user's inboxes are hosted. Our study is based on filtering Emails (bads and goods) by the clustering of optimized data compared with unoptimized one.

Mathematics Subject Classification. 05C12, 05C50, 05B10, 91C20, 15A18, 34A05.

Received October 29, 2013. Accepted November 12, 2015.

1. Introduction

While the web grows more and more, large amounts of data are collected and we reap more than we can handle. We usually use the data optimization as a solution to reduce the volume of processing data, but at the same time we need to keep the information. In our case, we take the emails which are the fastest communication service on Internet thanks to its underlying technology in the network (sending and receiving), where we can have or receive criminalized emails by breakthroughs network. Therefore a tool for detecting these emails gangs become a necessity.

In the literature review a lot of research are carried out on bad emails filtering based on Data Mining techniques, but existing works use the data retrieved without modification (pretreatment). Our study is to optimize the data retrieved in pretreatment (in our case these are the body of emails) and then to apply the existing classification algorithms. The data optimization is based on the SVD method (Singular Value Decomposition) to reduce processing time and gain memory space.

In our paper, we first present in second section a related work on email classification, email thread detection and email dataset, in section three we give some background on emails, Singular Value Decomposition and Clustering, in section four we present our proposed approach – we work on the body of the email – and discuss the results. Finally we end with a conclusion and perspectives.

Keywords. Email, feronsics, spam, SVD, LSI, optimisation, data mining, clustering.

[1] University Abderrahmane Mira of bejaia, LIMED Laboratory, 06000 Bejaia, Algeria. `dh.salhi@gmail.com; tarikamel59@gmail.com`

[2] University college Dublin, Parallel Comptaional Research Group Laboratory, Dublin4, Dublin, Ireland. `Tahar.kechadi@ucd.ie`

Article published by EDP Sciences

2. Related work

Classification has been widely used for the detection of spam but the main existing systems use text classification and machine learning techniques. For instance, TF-IDF, Naïve Bays, rule-based system and support vector machine [18]. SwiftFile and MailCat use the AIM text classifier, which is a TF-IDF [25] classifier to score high classification accuracy on the task of classifying Emails into folders.

The email classification has the potential to eliminate many of problems of current mail filters. Research shows that the text classification and machine learning algorithms can achieve an accuracy of 80% or more [26], even when the classification tasks are across a wide range of classes.

The mail systems widely used to make conversations and discussion groups and research show that the threading varies between 25% and 30% of the ordinary emails [25].

Before the emails prevalence, the online mail systems as the Usenet network and other report systems provide more support for threading [25]. The thread organization in the mail systems can use the same methods as long as the threading systems include some information for the source email in the reply email, the message-Id is copied into the field In-Reply-To of the child email, in the same way the subject is copied also but prefixed by Re: it's important to add the source email and the sending date into the body of child emails [3].

This kind of studies based on the follow of children in the email list to detect the good and bad ones.

The common email databases from real data are not easy to obtain because of the privacy concerns. Before the ENRON Email dataset, the researchers used personal Email data to evaluate the performance of emails classifications, and this is a big problem [7].

With the large number of users, messages, threads and folders, the ENRON Email dataset is a suitable corpus for evaluation of email classification methods. The research on ENRON dataset helped to reveal many emails features, for example 200 399 messages, 30 091 threads and 61.63% of emails in the corpus are in threads [2].

3. Background

Before presenting our approach (in Sect. 4), we will give in this section some background on Emails, Singular Value Decomposition and Clustering.

3.1. E-mails

The e-mail was invented by Ray Tomlinson in 1972 [16] and its usage is relatively simple, this is why it becomes the main service used of Internet. In the manner of conventional postal service, you only know the address of the sender to send a message [16]. It has two main advantages to e-paper which are: a speed of transmission and the cost reduce (overall cost of Internet connection).

3.1.1. The structure of an e-mail

In 1982, a standard way for E-mails provided by Internet has been defined. It is based on a convention adopted by Ray Tomlinson [12], but they are updated to reflect the modern state of Internet. SMTP (Simple Mail Transfer Protocol) is the underlying technology behind the transfer of email from a web host to another. In the early days of Internet, electronic mail systems use special applications Gateway to transfer email from a proprietary system to another. Hence the E-mail is composed of two parts shown below in the figure [14], where the first one is the header and the second one is the message body.

3.1.1.1. *The header of the message*

The E-mail is composed of several lines of text when transmitted between network nodes, where each line is terminated by the Carriage return character (ASC II code 13). The header contains a limited number of lines that are structured and well defined by RFC 822 format. An example of a common header for composing messages [5]:

```
From:  <dh.salhi@exemple.com>
To:  <societyX@exemple.com>
```

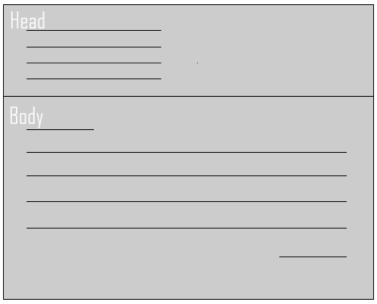


FIGURE 1. The structure of an E-mail.

```
Cc:  <Assistant@exemple.com>
Date:  Fri, 7 Jan 2011 14:20:24 -0500
Subjet:  How electronic mail works
```

From: This is the email address of the source, it is generally defined by the mail client according to its preference.

To: This field corresponds to the email address of the reciever.

Cc: We can send an email to many people by writing their respective email addresses separated by commas.

Date: Creation date of the email.

Subject: This is the way that the reciever will see when they want to read mail.

3.1.1.2. *The body of an e-mail*

The message body is a text written by human users, the Internet standard coded e-mails in ASC II format [14], it is common for end users. Another possibility is to attach files to e-mail that are in a binary format and of course not written by human users. Because of these attachments, it is possible for client applications E-mails to create massages having different body ASC II code [16].

3.1.2. MIME standard [5]

MIME (Multipurpose Internet Mail Extensions) is a standard that was proposed by Bell Laboratories in 1991 to extend the limited possibilities of Emails and in particular to allow the insertion of documents (images, sounds, text, ...) in an email. He is originally defined by RFC 1341 and 1342 from June 1992. MIME messaging brings the following features: MIME standard has the ability to handle multiple objects (attachments) in a single message and has unlimited message length. It uses a rich text (formatting of the messages, fonts, images, *etc* ...) and binary attachments (excutables, images, audio or video files, *etc* ...), possibly with several parts.

3.2. Singular value decomposition

This algorithm is based on the compaction of identical rows and columns in a matrix (or proportional) by merging them into one row or column and then performing of the singular values decomposition (SVD) of the original matrix. The last step in the expansion can be omitted to preserve storage space [6].

3.2.1. The mechanism of decomposition

We present definitions of singular value decomposition given by GOLUB and VAN LOAN (1989) [17].

Definition 3.1 ([17])**.** A matrix Q^{m*m} is orthogonalif $Q^TQ = I$
where Q^T is the transposed matrix of Q.

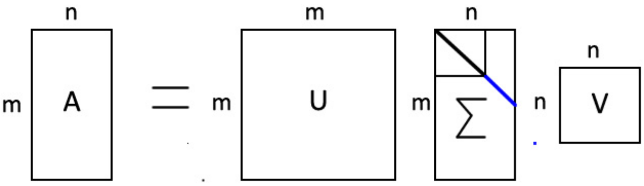


FIGURE 2. Singular Value Decomposition.

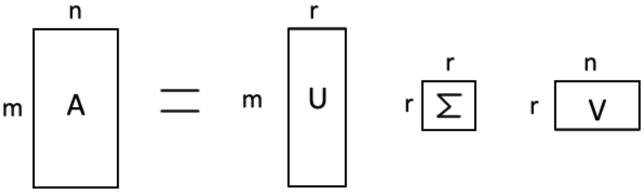


FIGURE 3. Singular Value Decomposition reduced.

Definition 3.2 ([17]). A matrix $Q \in R^{n*m}$ is said orthotoned column if $Q^T Q = I$.

Theorem 3.3 ([19]). *It exist $A \in R^{n*m}$ so that two orthogonals matrix*
$U \in R^{m*m}$ and $V \in R^{n*n}$ where $m \rangle n$
where: $U^* AV = \Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r) \in R^{n*m}$
So: $A = U \Sigma V^*$.

Where:

$U \equiv$ singular matrix $(m * m)$.
$V \equiv$ singular matrix $(n * n)$.
$\Sigma \equiv$ diagonal matrix $(n * m)$.
$V^* \equiv$ Adjoint $(V) \equiv \text{Comatrix}\,(V)^T$.

and here is a graphic demonstration of the SVD showed in Figure 2.

- The black plug contains singular positive values (R+).
- The blue bung contains zero singular values.

Σ contains mostly zeros, suggesting that there must be a smaller version of SVD, or most of the zeros would be eliminated [15], this is the case and the reduced version of SVD is represented as follows in Figure 3:

- $(m - r)$ columns are removed to the right of U.
- $(r - n)$ columns are removed to the right of V.
- It just takes the square part Superior (black) of Σ This reduced form of SVD is equivalent to as "full" which is obtained from the reduced form.
- We can show that the reduced SVD is unique if and only if all singular purchesers strictly positive are distinct.

Definition 3.4 ([19]). Σ is said a singlar value of A if
it exists $u \in R^m$, $v \in R^n$ that
$Av = \Sigma u$
$A^* u = \Sigma v$

Where:
u: right singular vector.
v: left singular vector.

Table 1. A diagonal matrix "Σ".

$$\begin{pmatrix}
u_1 & 0 & . & 0 & 0 & 0 & . & 0 & 0 \\
0 & u_2 & . & 0 & 0 & 0 & . & 0 & 0 \\
. & . & . & . & . & . & . & . & . \\
0 & 0 & . & u_m & 0 & 0 & . & 0 & 0 \\
0 & 0 & . & 0 & v_1 & 0 & . & 0 & 0 \\
0 & 0 & . & 0 & 0 & v_2 & . & 0 & 0 \\
. & . & . & . & . & . & . & . & . \\
0 & 0 & . & 0 & 0 & 0 & . & v_n & 0 \\
0 & 0 & . & 0 & 0 & 0 & . & 0 & 0
\end{pmatrix}$$

Note:
If A is Hermitian (its eigenvalues are ones) so: $A = VIV^*$.

3.2.2. The relationship between SVD and eigenvalues

From the previous remark, we see that if there is a hermitian matrix, we will have the same equation like SVD except that $U = V$, then we will compute A^*A and AA^* as follow:

$$A^*A = V\Sigma^*U^*U\Sigma V^* = V(\Sigma^*\Sigma)V^*$$
$$AA^* = U\Sigma V^*V\Sigma^*U^* = U(\Sigma\Sigma^*)U^*.$$

where A^*A and AA^* has respectively $(m*m)$ and $(m*m)$ dimensions. The square modules of each nonzero singular value of A is equal to the modulus of the eigenvalue A^*A et de AA^*.
$|VS|^2 = |VP|$.

In sammury

- The columns of U are the eigenvectors of the matrix AA^*.
- The columns of V are the eigenvectors of the matrix A^*A.
- The matrix Σ is a diagonal matrix built by the eigenvalues of AA^* then A^*A showed in Table 1.

3.3. CLUSTERING

Clustering is an operation that consists in grouping objects into groups (clusters) having two properties: they are not predefined by the analyst but discovered during the operation and the clusters includes objects with similar characteristics and separating items with different characteristics (internal homogeneity, external heterogeneity). Unlike ranking algorithms, the clusters are not known in advance and even the number of classes to which each object belongs. The number of classes is not always fixed in advance. This means that there are no variables to explain.

3.3.1. Complexity of the segmentation [11]

To find the complexity of the problem, recall that the number of partitions of n objects is the number of Bell:

$$B_n = \frac{1}{e}\sum \frac{k^n}{k!}.$$

3.3.2. Distances computation

In order to perform the clustering of the data structured in a matrix as follows:

TABLE 2. Data Matrix.

	Att 1	Att 2	…	Att j	…	Att m
row 1	0	3	…	4	…	1
…	…	…	…	…	…	…
row i	1	5	…	7	…	1
…	…	…	…	…	…	…
row n	2	7	…	4	…	5

We need to compute distances between the different lines of data matrix to obtain another matrix shown in Table 3 [21]:

TABLE 3. Distance matrix.

row 1	0					
row 2	d(2,1)	0				
…				0		
row $n-1$	$d(n-1,1)$	$d(n-1,2)$	$d(n-1,3)$	…	0	
row n	$d(n,1)$	$d(n,2)$	$d(n,3)$	…	$d(n,n-1)$	0

3.3.3. Main measures of distance [1]

It exists in the literature several measures of distances, the main ones are:

- Euclidean distance: $d_E(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$.
- Manhattan distance: $d_M(x,y) = \sum_{i=1}^{n}|x_i - y_i|$.
- Minkowski distance: $d_{MK}(x,y) = \sqrt[q]{\sum_{i=1}^{n}(x_i - y_i)^q}$.

3.3.4. Basic concepts of clustering

(1) Partitioning method: Construct a partition of a database D of n objects into a set of k clusters [22].
(2) Given a k, find a partition of k clusters that optimizes the chosen partitioning criterion [22].

4. PROPOSED APPROACH

Compared to the existent works presented before (Sect. 2. Related Work) the clustering has a very important role to filter good emails from bad ones. To apply the clustering algorithms, we use a dataset of emails organized in a matrix of two dimensions $A_{(n,m)}$. The aim of our study is not to create new clustering algorithm but to optimize the number of dimensions of emails matrix before applying the existent algorithms and compare the results found with the matrix unoptimized.

In this section, we will present our proposed approach starting with the data collection to the detection of bad emails (Fig. 4).

4.1. Collect emails

To apply our proposed approach we need to extract emails located on mail servers or user's inboxes, therefore we collect a big number of emails to test our approach.

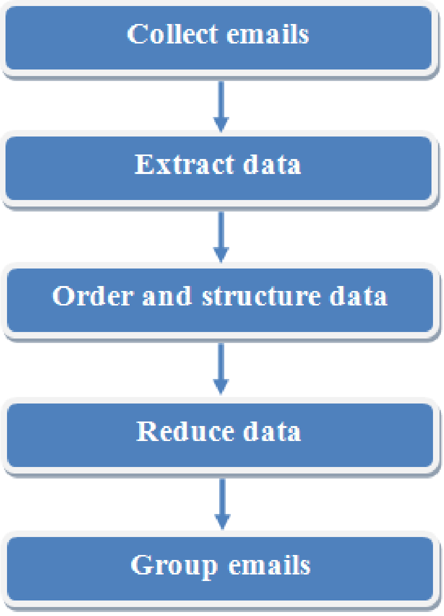


FIGURE 4. Proposed approach.

4.2. Extract data

In this subsection, we discuss the step of extracting data from our emails to build the data matrix (Emails matrix), where we present a new mechanism to treat the body of the email based on the language processings.

Emails are a large collection of data, written in different, the language of emails is considered as very important way to analyse it. So, we need to process the email body.

4.3. Order and structure data

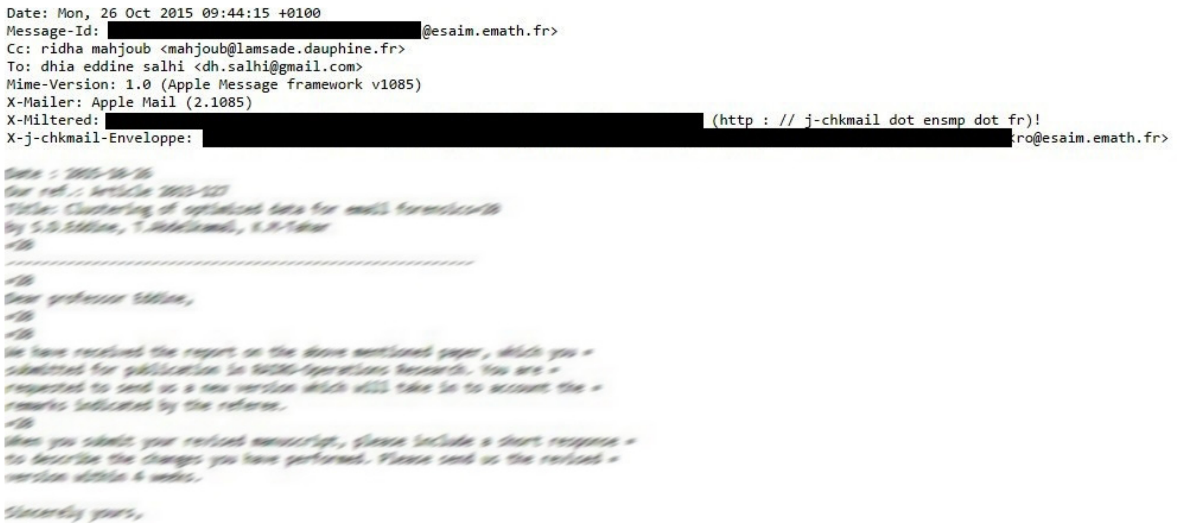
From the precedent subsection (the extraction of words making the body of emails) we base in this step to organize these words in a emails matrix with two dimensions, or the lines are the identifiers of the emails and the columns are the attributes (words cited before), contents of the matrix is the number of occurrences of each attribute in such email. Let us take the following example: we have three emails or each one has its body:

Email 01: "I like hotmail and gmail"
Email 02: "I like hotmail and I like gmail"
Email 03: "I hate gmail"

the matrix associated with these emails is (Tab. 4):

TABLE 4. The associated matrix of emails.

	I	**like**	**hate**	**hotmail**	**and**	**gmail**
Email 01	1	1	0	1	1	1
Email 02	2	2	0	1	1	1
Email 03	1	0	1	0	0	1



```
Date: Mon, 26 Oct 2015 09:44:15 +0100
Message-Id: [redacted]@esaim.emath.fr>
Cc: ridha mahjoub <mahjoub@lamsade.dauphine.fr>
To: dhia eddine salhi <dh.salhi@gmail.com>
Mime-Version: 1.0 (Apple Message framework v1085)
X-Mailer: Apple Mail (2.1085)
X-Miltered: [redacted] (http : // j-chkmail dot ensmp dot fr)!
X-j-chkmail-Enveloppe: [redacted]ro@esaim.emath.fr>

[redacted]
```

Figure 5. Sample of extraction of email source.

In the case: M(2,1) = 2 ⇒ the number of occurrences of “I” in the second email. From here we can build our total matrix of emails.

4.4. Reduce data

In this part, we use SVD method (illustrated before) to reduce the volume and the number of dimensions of data, before applying the algorithms of classification. It is based on the decomposition of the original matrix in three matrixes U, Σ and V, to reduce the number of the columns.

Now, we have a reduced matrix which is ready to use in classification.

4.5. Group emails

After having the optimized emails matrix by Singular Value Decomposition, the classification (Clustering) of emails is very important step. To classify these emails into groups (clusters), the first step is to make distance between emails. To calculate the distance several methods exists (Euclidian distance, Manhattan distance, Minkowski distance, *etc.*). In our case, we use discrete variables, the most direct path between two points (emails) is calculated with the Euclidian distance, therefore we choose it in our study.

Now, we have the distances between emails represented in distance matrix, we can apply a classification algorithm of data mining.

From this classification, we can define number of clusters, at the same time we get a number of emails not classified. These unclassified emails are what we call criminalized emails (gangs).

5. Results and discussion

In this section, we will implement our proposed approach starting with the collection of information (emails) and ending with the classification passing through data optimization. The language used in the implementation is the R, where we will program the SVD mechanism and the classification algorithms (Kmeans, and hclust AGNES).

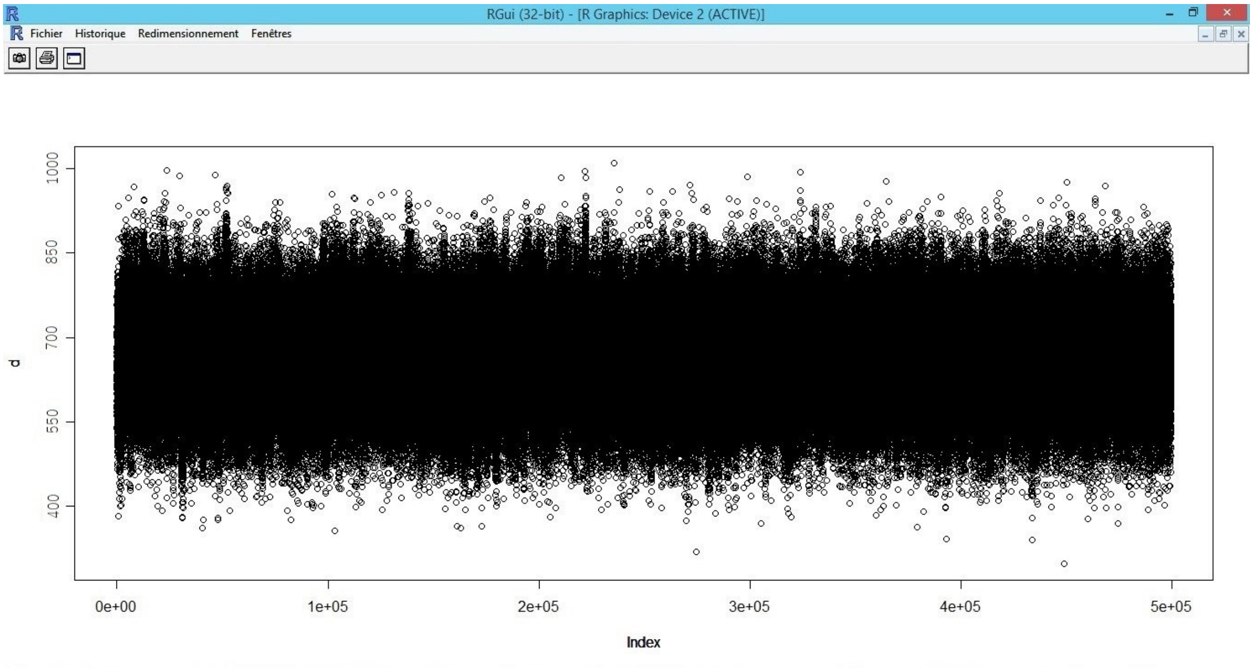


FIGURE 6. The projection points of the original matrix.

5.1. Matrix creation

For any treatment on emails (classification, filtering, *etc* ...), we need a well-organized representation of these emails in a well structured form. the Email is composed of several words in several languages. For this, the best representation of the email is in the form of a two-dimensional matrix where the rows represent the number of emails and the columns contain the identifier of the email (Message-ID) and the number of occurrences of each word that compose it.

The extraction of words made by recording sources emails (In our case we use an inbox first author of paper – Gmail) in a text file and process them in an application that we made, which calculates the number of occurrences of each word in the emails and makes an overall array that includes all the emails. For example, if a word exists in an email three times: we represent it by the number of occurences which is three, if does not exist we represent it by zero (explained before in Sect. 4.3).

After the creation of the matrix we can calculate now its size, 1000 rows and 4503 columns, the projection of this original matrix is represented in the next figure like a Cloud of points in two dimensions.

5.2. The reduction of the matrix

After the extraction of information (words) from the emails and build the data matrix as explained in the previous subsection we go to the second step which is the reduction of this matrix, remembering to keep the information.

The reduction (optimization) is made by an algorithm that is the Singular Value Decomposition (SVD), where we decompose the original matrix into three submatrices U, V and Σ to get the reduced matrix illustrated as follows:

$$[U, , V] = \mathrm{svd}(\mathrm{M}, 0);$$
$$\mathrm{MSVD} = U * \Sigma * V.$$

Figures 7–9 present this decomposition graphically:

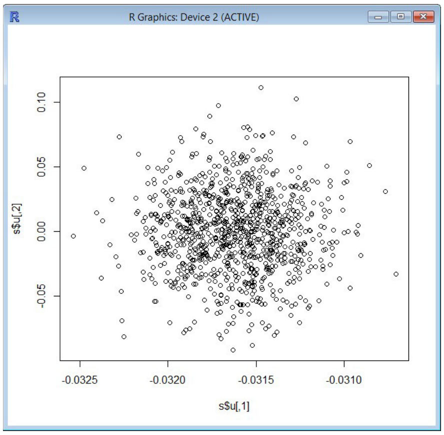


Figure 7. *U* table.

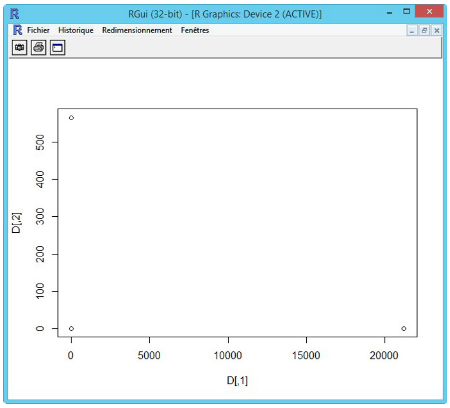


Figure 8. Σ table.

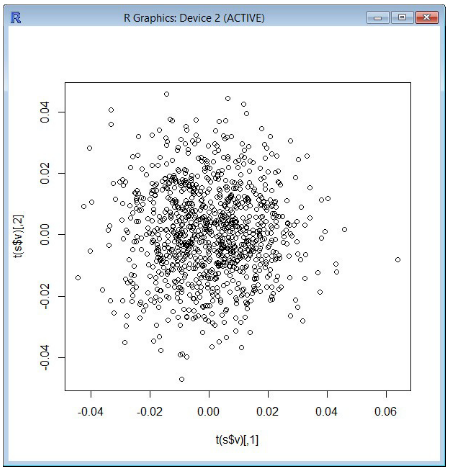


Figure 9. *V* table.

And here is the result of the reduced matrix projected in a Cloud points in two dimensions, shown as follow (Fig. 10):

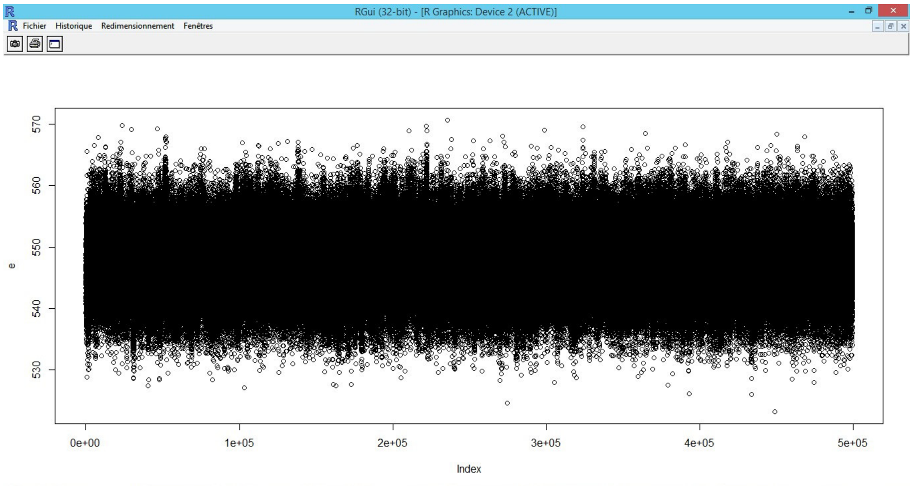


FIGURE 10. The projection points of the reduced matrix.

the following table (Tab. 5) shows a simple comparison between original matrix and reduced matrix, where we can observe the gain of rows and columns eliminated.

TABLE 5. Comparison table.

Matrix	Number of lines	Number of Column
Original	1000	4503
Reduced	750	3700

5.3. Clustering

Now, we implement the clustering to detect bad emails. First, we calculate the distances between emails to build distance matrix based on Euclidian method (explained before sin Sect. 3.3).

From the distance matrix we apply clustering algorithms: where we begin with the **Kmeans** algorithm, after we pass to apply **HCLUST** and **AGNES** algorithms for the hierarchic Clustering, and this is the result of those algorithms shown in Figures 11 and 12.

After having the classification results (clustering) of the three algorithms implemented before: **Kmeans**, **HCLUST** et **AGNES** shown in both previous figures. Now we can extract the number of clusters, the number of emails that make up each one and the number of unclassified emails.

– Cluster 01: 128 Emails.
– Cluster 02: 57 Emails.
– Cluster 03: 188 Emails.
– Cluster 04: 154 Emails.
– Cluster 05: 87 Emails.
– Cluster 06: 122 Emails.

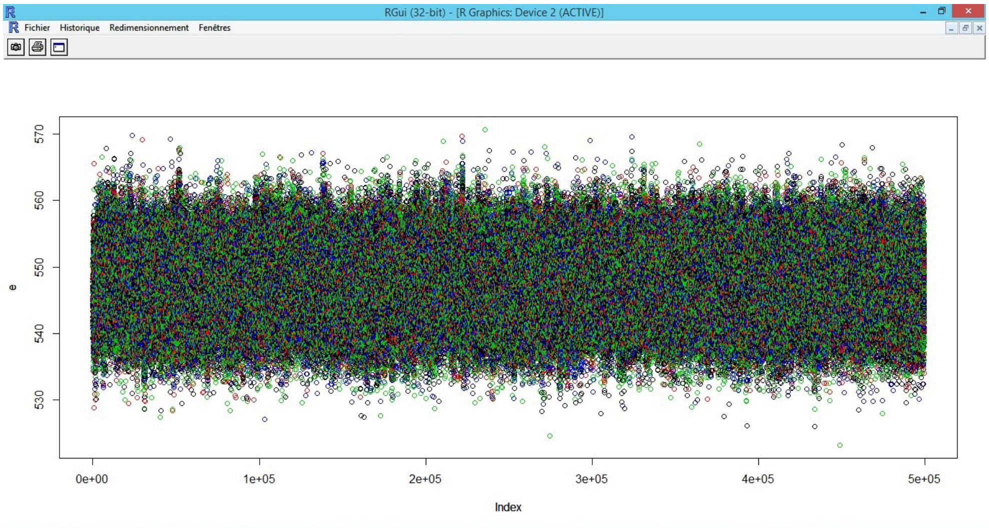


FIGURE 11. Results of clustering with Kmeans algorithm.

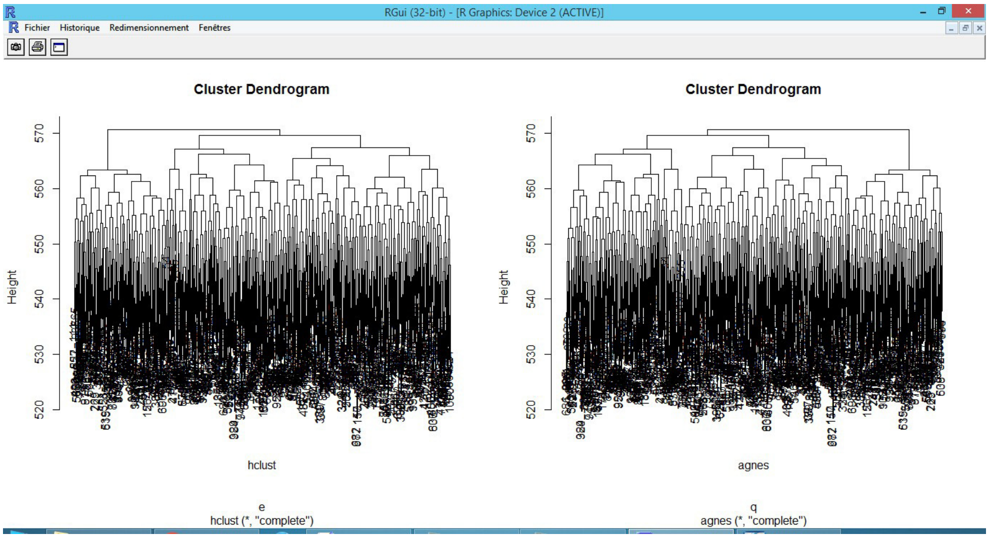


FIGURE 12. Results of clustering with hierarchic (**HCLUST** and **AGNES**) algorithms.

– Unclassified Emails: 14 Emails.

Our result is very close to reality as the inbox that we used for testing contains 15 emails like spam, and when we checked the identifiers of spam with the unclassified emails of our result, we find that they are the same except that an email classified in the cluster 5 but in our box is spam.

6. Conclusion

In this paper, we studied new technique for processing emails to detect the bad ones. Our study based on the processing of the second part of email (Body), we began by a preprocessing step where the data (Emails) are organized in a matrix and optimized by Singular Value Decomposition method to be classified by clusterinf methods. Our perspectives is to treat the other part of emails (Header) and develop new method to filter them.

References

[1] S. Bandyopadhyay *et al.*, Clustering distributed data streams in peer-to-peer environments. *Inf. Sci.* **176** (2006) 1952–1985.

[2] R. Bekkerman, Automatic categorization of email into folders: Benchmark experiments on Enron and SRI corpora (2004).

[3] P. Bowes, Increased use of electronic communications tools among North American and European workers, press release (2000).

[4] D. Clot, *Méthodologies de fouille de données pour la modélisation dans les processus d'aide à la décision complexe: application à l'analyse des paramètres de déformation du coeur.* Thèse de doctorat, Lyon 1 (2002).

[5] S. Curtis, Pro open source mail: Building an enterprise mail solution (2006).

[6] L. De Lathauwer, B. De Moor and J. Vandewalle, A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.* **21** (2000) 1253–1278.

[7] J. Diesner, T.L. Frantz and K.M. Carley, Communication networks from the Enron email corpus It's always about the people. Enron is no different. *Comput. Math. Organization Theory* **11** (2005) 201–228.

[8] U. Fayyad, G. Piatetsky–Shapiro and P. Smyth, From data mining to knowledge discovery in databases. *AI magazine* **17** (1996) 37.

[9] G.T. Fernando, Distributed systems: principles and paradigms. Edited by Andrew S. Tanenbaum, Maarten Van Steen Pearson Education, Inc., 2007 ISBN: 0-13-239227-5. *J. Comput. Sci. Technol.* **11** (2011) 115–116.

[10] J.Y. Halpern and R. Fagin, Modelling knowledge and action in distributed systems: Preliminary report. Springer Berlin Heidelberg (1988).

[11] J. Han, M. Kamber and J. Pei, Data mining: concepts and techniques: concepts and techniques. Elsevier (2011).

[12] P. Hazel, Exim: The Mail Transfer Agent. O'Reilly Media, Inc. (2001).

[13] D.T. Larose, Discovering knowledge in data: an introduction to data mining. John Wiley Sons (2014).

[14] A. Mcdonald *et al.*, Linux E-mail. Packt Publishing Ltd (2009).

[15] A. Mirzal, Clustering and Latent Semantic Indexing Aspects of the Singular Value Decomposition. Preprint arXiv:1011.4104 (2010).

[16] D. Mullet and I. Managing, O'Reilly Media, Inc. (2000).

[17] B. Rosario, Latent semantic indexing: An overview. *Techn. Rep. Infosys* **240** (2000).

[18] P.H. Sellers, The theory and computation of evolutionary distances: pattern recognition. *J. Algorithms* **1** (1980) 359–373.

[19] M. Sogrine, T. Kechadi and N. Kushmerick, Latent semantic indexing for text database selection. In: *Proc. of the SIGIR 2005 Workshop on Heterogeneous and Distributed Information Retrieval* (2005) 12–19.

[20] R. Sureswaran *et al.*, Active e-mail system SMTP protocol monitoring algorithm. In: *Broadband Network Multimedia Technology*, 2009. IC-BNMT'09. 2nd IEEE International Conference on. IEEE (2009) 257–260.

[21] E. Triantaphyllou, Data Mining and Knowledge Discovery via Logic-Based Methods: Theory, Algorithms, and Applications. Springer Science Business Media (2010).

[22] J. Tarhio and M. Tienari, Computer Science at the University of Helsinki 1991. University of Helsinki, Department of Computer Science (1991).

[23] S. Tufféry, Data mining et statistique décisionnelle: l'intelligence dans les bases de données. Editions Technip (2005).

[24] G.J. Williams and S.J. Simoff (eds.). Data mining: Theory, methodology, techniques, and applications. Springer (2006).

[25] S. Whittaker, Supporting collaborative task management in e-mail. *Human Comput. Interaction* **20** (2005) 49–88.

[26] S. Whittaker and C. SIdner, Email overload: exploring personal information management of email. In: *Proc. of the SIGCHI conference on Human factors in computing systems.* ACM (1996) 276–283.