The application of the CLOPE algorithm for clustering a set of
the network packets

A V Belov 1 T D Novikova 2
1. School of The Applied Mathematics of the National Research University Higher School of
Economics, Myasnitskaya ulitsa 20, 101000, Russian Federation
2. Department of Systems and Software Engineering, NUST MISIS, Moscow, Leninskiy prospekt 4
Moscow, 119049, Russian Federation

Abstract - For the modern software that uses network communication protocols the problem
of the ensuring reliability is acute. To solve such an important problem the stress testing is
used. This type of the testing involves the generation of a large number of the test data,
including sets of network packets. Reducing the stored data after the testing process is the main
task. This task can be solved by clustering the set of received packets. To solve this problem it
is proposed to use the clustering algorithm for categorical data of CLOPE. This algorithm
allows to cluster datasets without the information about the source clusters. It has the low
computational complexity and ease of implementation. The article describes the preparation
and results of sets of the network packets processing experiments. The article shows that the
CLOPE algorithm can be effectively used for the clustering network packets received during
stress testing. The results of the research extend the toolkit for the SW stress testing process.

1. Introduction
For testing software which used network messaging one of the most popular methods is the stress
testing. Tests are developed using the black box method. Test scenarios are selected according to the
functional profile and usually using a pseudo-random selection algorithm [1]. This algorithm can be
used for the testing non-functional attributes such as SW reliability and SW performance.

However the negative consequence of using the random testing is that the obtained test results, and the
program execution paths do not show the reasons why they are performed and do not allow us to
understand which parts of the input data are the cause of this execution.

Another important feature is a large amount of data sets generated during testing. This feature of
testing leads to the “path explosion problem” [2] when there are many semantically similar traces that
can not be resolved to one branch of the SW application code.

Thus, it becomes necessary to select groups depending on certain signs in the set of paths obtained
which will reduce the number of data stored in the test results and highlight those parts of the data,
changing which changes the behavior of the application. In the case of using selective testing in
applications that use network interaction, it becomes expedient to replace clustering with a set of
traces to clustering a set of packages, each of which would in fact be an unique identifier of the path.

2. Choosing an approach to the clustering
To improve the efficiency of the testing process, it is proposed to use the network packet clustering. It
is required to construct an algorithm α: X→Y where X – the test network packets and Y – a set of
clusters (initially nothing is known about Y, even the number of clusters can be unknown) [3]. Each
cluster consists of close packages and packages from the different clusters are the significantly
different.
The clustering algorithm will allow:

- Identify the structure of the unknown network protocol for researcher.
- Evaluate the test coverage of the software being investigated using coverage metrics based on the input data and metrics based on the control flow, since the input network packets may contain data used in the program environment.
- Map selected clusters of the packets that lead to failures with the types of errors.
- Select packet masks that lead to errors.

When an approach to clustering was choosing it was necessary to specify a measure of the proximity of objects. For string sequences the Hamming metric is most often used:

\[
d_{ij} = \sum_{k=1}^{p} |x_{ik} - x_{jk}|
\]

Where is \(d_{ij}\) – distance between two objects,

\(x_{ik}\) – value of k-coordinate i-object,

\(x_{jk}\) – value of k-coordinate j-object.

To solve the problem of the clustering a set of network packets it is necessary that the final form of clusters has to be an arbitrary form. Since the estimated number of clusters at the initial stage of the study is not known it is necessary to use an algorithm that does not use this input parameter.

To solve this problem is proposed the CLOPE Algorithm (Clustering with sLOPE), a non-hierarchical iterative cluster analysis method for the processing the large sets of the categorical data.

3. **CLOPE algorithm for the stress testing process**

The original CLOPE algorithm is a transaction data clustering algorithm (a transaction is some arbitrary set of the objects of finite length). The main idea of this method is to use a global optimization criterion based on the maximization of the cost function as applied to clustering problems [4].

During the execution of the algorithm, a relatively small amount of information about each cluster is required to be stored in RAM and the minimum number of passes is made on the data set. When one uses the CLOPE algorithm the number of clusters is selected automatically and depends on the repulsion coefficient - the parameter that determines the level of the similarity of transactions within the cluster. The repulsion factor is set by the user: The larger the repulsion coefficient the lower the level of transaction similarity and as a result more clusters will be created.

Transaction database consisting of a set of transactions. Each transaction is a set of objects. A set of clusters is a partition of a set such that each element of C is called a cluster, n is the number of transactions, A is the length of the transaction, k is the number of clusters. For the task of clustering network protocols, A is the maximum packet length.

In this case, the parameter r is set – the repulsion coefficient.

Cj – cluster can be characterized by the following parameters:

\(D(Cj)\) – the set of objects in the Cj cluster.

\(Occ(Cj)\) – the number of occurrences of the object I in the cluster Cj.
\[ S(C_j) = \sum_{i \in D(C_j)} \text{Occ}(i, C_j) = \sum_{t_i \in C_j} |t_i| - \text{is the total number of elements in the cluster } C_j. \]

\[ W(C_j) = |D(C_j)| - \text{cluster width.} \]

\[ H(C_j) = \frac{S(C_j)}{W(C_j)} - \text{is the height of the cluster.} \]

CLOPE algorithm includes 2 main phases (Fig.1):

![CLOPE flowchart.](image)

At the initialization phase the clustering cost is calculated for each packet in the set and then the refinement occurs where the packets are moved between the clusters to maximize the clustering cost function – profit.

\[
\text{Profit}(C) = \frac{\sum_{i=1}^{k} H(C_i) \times |C_i|}{\sum_{i=1}^{k} |C_i|} = \frac{\sum_{i=1}^{k} \frac{S(C_i)}{W(C_i)^r} \times |C_i|}{\sum_{i=1}^{k} |C_i|} \tag{2}
\]

After that refinement iterations take place in order to improve the existing distribution. As part of the refinement iterations the entire set of transactions is re-iterated. For each the cost of deletion from the current cluster is calculated:

\[
\text{Cost}^-(C_i, t_n) = \frac{(S(C_i) - \text{length}(t_n)) \times (|C_i| - 1)}{W(C_i/t_n)^r} - \frac{S(C_i) \times |C_i|}{W(C_i)^r} \tag{3}
\]

And also the cost of adding to the another cluster is calculated:

\[
\text{Cost}^+(C_i, t_n) = \frac{(S(C_i) + \text{length}(t_n)) \times (|C_i| + 1)}{W(C_i \cup t_n)^r} - \frac{S(C_i) \times |C_i|}{W(C_i)^r} \tag{4}
\]

Based on the calculated values it is decided to move the transaction to another cluster or this transaction remains in the current cluster. As with initialization it is done in such a way that the total cost for the deleting from the current cluster and adding to another cluster is the highest (if for all movements the total cost is a negative then the transaction remains in place).
4. Quality metrics

For the CLOPE algorithm the quality parameters are as follows [2]:

- The average similarity of elements

\[
M_{\text{similarity}} = \frac{1}{E} \sum_{i=0}^{E} \left( \frac{1}{O(C_i)} \sum_{j=1}^{O(C_i)} \frac{|\{L_m | L_m \in D_j, L_m \in C_i \}, m = 1, ..., W(D_j)|}{W(D_j)} \right) \times 100
\]  

(5)

Where \(O(C_i)\) is the number of packets in the cluster, \(L_m\) – cluster package, \(D_j\) – the first cluster, \(C_j\) – second cluster, \(E\) - number of clusters.

- The average distance between clusters, as the average value of the "Manhattan" distance

\[
M_{\text{manh}} = \frac{\sum_{i=1}^{E} \sum_{j=1}^{E} |\text{Occ}(L_n, C'_i) - \text{Occ}(L_n, C'_j)|}{(E-1)^2}, i \neq j, n = 1, ..., W(C'_i) = 1, ..., W(C'_j)
\]  

(6)

Where \(C'_i\) and \(C'_j\) are clusters. \(\text{Occ}(L_n, C_j)\) - the number of the occurrences of the object \(L_n\) in the cluster \(C_j\).

- The minimum distance between clusters

\[
M_{\text{min}} = \min \{|\text{Occ}(L_n, C'_i) - \text{Occ}(L_n, C'_j)|, i = 1, ..., E, j = 1, ..., E, i \neq j, n = 1, ..., W(C'_i) = 1, ..., W(C'_j)\}
\]  

(7)

- The maximum distance between clusters

\[
M_{\text{max}} = \max \{|\text{Occ}(L_n, C'_i) - \text{Occ}(L_n, C'_j)|, i = 1, ..., E, j = 1, ..., E, i \neq j, n = 1, ..., W(C'_i) = 1, ..., W(C'_j)\}
\]  

(8)

- The number of iterations of the algorithm

5. Computational experiments

Computational experiments were conducted on a test bench which is a network server that uses the HTTP protocol for the message transfer. Clustering was performed with the value of the repulsion coefficient \(r\): 2.7. This repulsion coefficient was used for a specific set of data generated using the Peach utility.

Table 1. Example of input data

| Cluster   | Packet                                      |
|-----------|---------------------------------------------|
| Cluster 1 | POST HTTP/1.1                               |
| Number of packets: | Host: 127.0.0.1\n                                   |
|           | Content-Type: text/html; charset=utf-8\n       |
| 121       | Content-Length: 30                          |
|           | Number:455                                  |
|           | Value1: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  |
|           | AaaaaaAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  |

The main scheme of the computational experiment is on the Figure 2.
![Flowchart](chart.png)

**Figure 2.** The computational experiment flowchart

The results of the computational experiment are presented in the Table 2.

**Table 2.** Test results on a test bench

| Parameters                             | Value  |
|----------------------------------------|--------|
| The average similarity of the elements | 62.33% |
| Average distance between clusters      | 74.35% |
If we draw conclusions from the obtained results we can see that in the clustering kit generated on the test bench the generated packets are similar to each other (the average similarity is more than 50%). The obtained similarity can be explained by the nature of the generated data since the transfer via the HTTP protocol implies the preservation of identical field names in the packet structure.

### 6. Conclusion

According to the results of successful testing of the clustering algorithm it was noted that the existing solution may improve the implementation of the mechanism for selecting the optimal parameters of the algorithm. Clusters have been identified that reflect the behavior of the software under investigation. This model allows clustering data only after performing the stress testing and isolating the types of software behavior of the interest to the researcher based on the data of the tested network packets. However, in the further investigations, it is possible to improve the result obtained by solving the problem of the selecting the parameters of the algorithm for each set of the packages individually.

Using the CLOPE algorithm for the clustering network packets will achieve the following goals:

- To identify the structure of the network protocol used in the software.
- To evaluate the software test coverage and to reduce the time for analyzing the test results.
- To reduce the volume of the stored data after testing replacing the set of the elements included in one cluster with one typical representative which will save memory for the database.

### 7. References

[1] Richard Hamlet (1994). "Random Testing". In John J. Marciniak. Encyclopedia of Software Engineering (1st ed.). John Wiley and Sons. ISBN 0471540021.

[2] Yan Shoshitaishvili, Ruoyu Wang, Christopher Salls, Nick Stephens, Mario Polino, Andrew Dutcher, John Grosen, Siji Feng, Christophe Hauser, Christopher Kruegel, and Giovanni Vigna. SoK: (State of) The Art of War: Offensive Techniques in Binary Analysis. In IEEE Symposium on Security and Privacy, 2016.

[3] Igor Mandel. Cluster analysis - M: Finance and Statistics– 1988. – 176 p.

[4] Katerina Novokhatska, Oleksii Kungurtsev Application of clustering algorithm CLOPE to the query grouping problem in the field of materialized view maintenance // J. Comput. Inf. Technol. 2016. Vol. 24, № 1. P. 79–89.