A New Method for Clustering Heterogeneous Data: Clustering by Compression

DORIN CARSTOIU, ALEXANDRA CERNIAN, VALENTIN SGARCIU, ADRIANA OLTEANU
Automatic Control and Computer Science Faculty
University Politehnica of Bucharest
313, Splaiul Independentei
ROMANIA
cid@aii.pub.ro, alexandra.cernian@aii.pub.ro, vsgarciu@aii.pub.ro , adriana@aii.pub.ro

Abstract: Nowadays, we have to deal with a large quantity of unstructured data, produced by a number of sources. For example, clustering web pages is essential to getting structured information in response to user queries. In this paper, we intend to test the results of a new clustering technique – clustering by compression – when applied to heterogeneous sets of data. The clustering by compression procedure is based on a parameter-free, universal, similarity distance, the normalized compression distance or NCD, computed from the lengths of compressed data files (singly and in pair-wise concatenation). Compression algorithms allow defining a similarity measure based on the degree of common information, whereas clustering methods allow clustering similar data without any previous knowledge.

Key-Words: clustering, heterogeneous data, clustering by compression, Normalized Compression Distance (NCD), FScore

1 Introduction

The expansive nature of the Internet produces a vast quantity of unstructured data, compared to our conception of a conventional data base [24]. This state of things is due to the fact that the data sources are very different: commercial websites, individuals, universities, research laboratories etc. Given the variety of data sources, we can expect the data itself to be very heterogeneous (text, images, films, programs etc.), which makes it difficult to organize and structure it in directories which could efficiently be used by professionals from different scientific domains or, simply, by whoever is interested in this sort of data.

The application of clustering on the World Wide Web is essential to get structured information from this information sea [25]. The most common application is to cluster web-pages into categories so as to facilitate the retrieval and filtering [1][2]. In these applications, a web-page is often represented by a bag-of-words model, constituting a dimensional feature vector. Based on this feature representation, most existing clustering techniques, such as partitioning-based clustering, hierarchical clustering and density-based clustering, could be applied to group similar web-pages. But the high dimensionality, as well as noise such as denotations, abbreviations and advertisement in web-pages, severely decreases the accuracy of clustering results.

In this paper, we intend to test the results of a new clustering technique – clustering by compression – when applied to heterogeneous sets of data. The clustering by compression procedure is based on a parameter-free, universal, similarity distance, the normalized compression distance or NCD, computed from the lengths of compressed data files (singly and in pair-wise concatenation). In order to validate the results, we calculate some quality indices. If the values we obtain prove a high quality of the clustering, in the near future we plan to include the clustering by compression technique into a framework for clustering web objects.

The rest of this paper is organized as follows. In Section 2, we present some related work regarding clustering heterogeneous data. In Section 3, we describe the clustering by compression procedure. We show how the normalized compression distance is deduced, the influence of the compression algorithms on the NCD and we present the main clustering methods of interest to our work. We show the experimental results of the proposed method in Section 4. Finally, we conclude in Section 5.

2 Related Work

Data clustering is a well-studied problem [3]. The applications of Web objects clustering (including web-pages, purchase items, queries, users, and etc.) utilize the link information at different level. Traditional clustering methods ignore the link information and clusters object solely based on content features [1][2]. Finding good similarity functions and selecting better features are the main focus of these works. There are some clustering algorithms which treat link information as additional features. That is, the heterogeneous data is first represented by homogeneous features before clustering. Su et al. [4] described a correlation-based document clustering method, which measures the similarity between web-pages based on the simultaneous visits to them.

In recent years, link information began to play a key role in some web applications. The agglomerative clustering approach described in [5] is another approach. This paper exploited the "click-through data" to form a bipartite graph of queries and documents. Afterwards, a graph-based agglomerative iterative clustering method is applied to merge vertices of the graph, until a termination condition is being reached [23]. But, unfortunately, it does not take into account the content features of both query and document, leading to ineffective clustering. In addition, this clustering method resulted in big clusters, which makes it difficult to specify a good termination criterion.

Collaborative filtering is another important application, where the clustering of people based on the items they have purchased allows more accurate recommendations of new items. Ungar et al. [6] presented a formal statistical model and compared different algorithms for estimating the model parameters.

A series of papers [7][8] about aspect model seems to be the first effort to assess the common problems of clustering heterogeneous data. Hofmann et al. [7] presented a statistical mixture models for unsupervised learning from dyadic data which contain pairs of two elements from two finite sets. This model is consequently applied in text mining [9], image segmentation [8] and collaborative filtering [10]. However, in order to apply their approach, one should first identify the latent class model of available data.

Other related works include distributional clustering [11] and categorical data clustering [12][13]. Distributional clustering builds statistical language models for natural language processing. Its main idea is to map each verb to a point in a high dimensional real-valued space, thus facilitating the similarity computation. Categorical data clustering deals with data whose attribute is non-numeric, e.g., a category name. Gibson et al. [12] proposed to assign and propagate weights on categorical values, and Guha et al. [13] proposed a concept of links to measure the similarity between data points. Both of them could be thought of as utilizing the similarity between category names in every column of tables to measure the similarity.

3 Clustering by Compression

Clustering is one of the most useful tasks in the data mining process for discovering groups and identifying interesting distributions and patterns in the underlying data. The clustering problem is about partitioning a given data set into groups (clusters), so that the data points in a cluster are more similar to each other than points in other clusters [14]. The relationship between objects is represented in a Proximity Matrix (PM), in which rows and columns correspond to objects. For example, consider a retail database records containing items purchased by customers. A clustering procedure could group the customers in such a way that customers with similar buying patterns are in the same cluster. Thus, the main concern in the clustering process is to reveal the organization of patterns into groups which allow us to discover similarities and differences, as well as to derive useful conclusions about them. This idea is applicable in many fields, such as life sciences, medical sciences and engineering. Clustering may be found under different names in different contexts, such as unsupervised learning (in pattern recognition), numerical taxonomy (in biology, ecology), typology (in social sciences) and partition (in graph theory) [15].

In the clustering process, there are no predefined classes and no examples to show what kind of relations would be valid among the data. Consequently, it is perceived as an unsupervised process [16]. On the other hand, classification is a procedure of assigning a data item to a predefined set of categories [17]. Clustering produces initial categories in which values of a data set are classified during the classification process.

In 2004, Rudi Cilibrasi and Paul Vitanyi proposed a new method for clustering based on compression [18]. The method does not use subject-

specific features or background knowledge, and works as follows:

• First, it determines a parameter-free, universal, similarity distance, the normalized compression distance or NCD, computed from the lengths of compressed data files (singly and in pairwise concatenation).

• Second, it applies a hierarchical clustering method.

The method is based on the fact that compression algorithms offer a good evaluation of the actual quantity of information comprised in the data to be clustered, without requiring any previous processing. This property of the compression algorithms can be deduced from Kolmogorov's mathematical complexity [19], as well as from the triangular inequality applied to two data samples and their concatenation. Both approaches lead to the definition of the normalized compression distance (NCD), which can be used as distance metric with the clustering algorithms.

If x and y are the two objects concerned, and C(x) and C(y) are the lengths of the compressed versions of x and y using compressor C, then the NCD is defined as follows

$$NCD(x,y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}} \qquad (1)$$

The normalized compression distance has a number of advantages over the classic distance metrics, the most important being its ability to cluster a large number of data samples, due to the high performance of the compression algorithms.

The NCD is not restricted to a specific application area, and works across application area boundaries. A theoretical precursor, the normalized information distance is provably optimal. However, the optimality comes at the price of using the non-computable notion of Kolmogorov complexity [20]. Nevertheless, experiments have proven that the NCD approximates optimality. To extract a hierarchy of clusters from the distance matrix, a dendrogram (ternary tree) is determined by using a clustering algorithm, under choice of different compressors. To substantiate the claims of universality and robustness, evidence of successful application has been reported in areas as diverse as genomics, virology, languages, literature, music, handwritten digits, astronomy, and combinations of objects from completely different domains, using statistical, dictionary, and block sorting compressors.

3.1 The Kolmogorov Complexity

The Kolmogorov information theory shows that a measure of information exists, at least theoretically, in the form of the Kolmogorov complexity, K(x). A data sample can be described by using a finite binary representation. This description is not unique and among the many descriptions that a data sample can have, the shortest one represents the measure of the complexity of that data sample.

Intuitively, the Kolmogorov complexity K (x|y) is defined like the length of the shortest binary description of X or, in other words, the length of the shortest machine program which prints X and then stops its execution. The programming language can be Java, LISP or any other. Moreover, the same program written in two different languages will have the same length up to an additive constant.

An important final remark is that the Kolmogorov complexity is not computable in the Turing sense of definition. This means that all applications using the Kolmogorov complexity must employ approximations.

3.2 Deduction of the Normalized Compression Distance

To deduce the NCD intuitively [18], we use the triangular inequality applied in the case of a compressor C and two samples of data x and y:

$$C(x) + C(y) - C(xy) \geq 0 \qquad (2)$$

This inequality is valid given the nature of the compression algorithms. The size of two files compressed together is smaller than the size of the files compressed separately, because of the following arguments:

The auxiliary information required by the compressor in order to decode the file is stored once if the files are compressed together and not twice if the two files are compressed separately.

If the two files share common information, then the size of the compressed version of their concatenation will be even smaller. It is easy to verify the validity of the triangular inequality by using a compression tool such as WinZip or WinRar. One must just choose two files and compress them separately, afterwards together and eventually check their compressed sizes.

In order to obtain a normal distance (which takes its value within the range of 0 to 1), it is necessary to carry out a series of calculations in (2). The first step is to check what happens when x = y (all the information contained in x is a part of y or the other

way around) or when x ≠ y (x and y do not share any piece of information).

$$lim_{x=y}[C(x) + C(y) - C(xy)] = max\{C(x),C(y)\} \quad (3)$$

$$lim_{x\neq y}[C(x) + C(y) - C(xy)] = 0 \quad (4)$$

If we withdraw (2) of 1 and divide the result by (3) in order to standardize the distance, the obtained formula covers the range from 0 to 1 and represents the normalized distance:

$$d(x,y) = 1 - [\, C(x) + C(y) - C(xy) \,] / max\{C(x),C(y)\}$$
$$\Rightarrow d(x,y) = [\, max\{C(x),C(y)\} - C(x) - C(y) + C(xy) \,] / max\{C(x),C(y)\}$$
$$\Rightarrow d(x,y) = [\, C(xy) - min\{C(x),C(y)\} \,] / max\{C(x),C(y)\} = d_{NCD}(x,y) \quad (5)$$

3.3 The influence of compression algorithms on the NCD

The quality of the NCD evaluation between the elements to be classified depends on the performances of the compression algorithms which are used to approximate the Kolmogorov complexity. The general rule to be considered is that the precision of the evaluation of the distance grows with the performances of the compression algorithm.

There are also other aspects which must be taken into account when analyzing the precision of the evaluation of the NCD. The currently available compression algorithms are still far from having ideal performances. Thus the influence of the compression algorithm on the NCD is determined by the characteristics and the performances of these algorithms when they are applied to various types of files of different sizes.

In order to obtain a good evaluation of the NID a real compressor must satisfy at least two of the conditions met by an ideal normal compressor:

1. Symmetry: C(xy) = C(yx);
2. Idempotence: C(xx) = C(x).

In fact, the tests prove that ZIP carries out very good evaluations of the NCD if the size of the submitted data is lower than 16KB, due to the fact that the dimension of the window of ZIP is 32KB. Hence, the compression ratio is very good. But, when the subjected files are large, ZIP generally tends not to observe the two conditions. In this case, other compressors, such as the arithmetic coder or BZIP [15] are appropriate to be used to evaluate the NCD.

Another very important aspect which must be taken into account is dealing with different manners of clustering, according to the nature of the elements to be classified.

Thus, from the point of view of the types of the files, it is necessary to distinguish two situations:
Case 1: the elements to be classified are files of different types (heterogeneous data)

Case 2: the elements to be classified are files of the same type (homogeneous data). Throughout this work, we are only concerned with the second case, namely clustering heterogeneous data.

The cause of the differences in the behavior of the algorithm is that certain algorithms (generally algorithms by statistical coding, such as Huffman coding) have similar compression rate, for example, for both image and text files, even if, generally, text is better compressed than images. Other algorithms adapt their performances to the density of the information of the file. If an algorithm which does not adapt its performance to the type of data, is employed with the NCD for data of various types, the clustering could return incorrect results.

It is also important to take into account the size of the files to be classified. The compressed version of a file is always made up of two parts:

- Useful information representing the part where the real content of the initial file is stored. In the best situation, this part would contain only the information of the original file.
- Auxiliary information - represents the part where additional information required to decompress the file is stored.

In order to evaluate more precisely the influence of the auxiliary data on the evaluation of the NCD, it is useful to analyze the ratio of auxiliary data and useful data given by:

$$R_{A/U} = |A| / |U| \quad (8)$$

where |A| is the size of the auxiliary data and |U| is the size of the useful data in the compressed version of a file. For small files, $R_{A/U}$ takes higher values as compared to the case of larger files, when these values are lower. In order to obtain the most precise evaluation of the NCD, it is required that the ratio $R_{A/U}$ be small (ideally 0).

3.4 Clustering methods

The goal of clustering methods is to group elements sharing the same information. The concept of similarity represents the essence of clustering. Data are represented by computers in binary format by

using two distinct symbols: 1 and 0. Therefore, two files are rarely identical, but could be similar to a certain degree, which means that the real problem of clustering remains to gather the elements which are most similar between them but less similar to all the others. This observation raises an important question if it is always possible to classify data. The answer is positive, but in certain cases the clustering might be not relevant.

The clustering methods can be divided into the following three categories:

– Distance methods
– Characters methods
– Quadruplets methods

In order to achieve the objective of this work, only the methods of distance and the methods of quadruplets (which also use criteria based on the notion of distance) were considered for analysis and implementation, for the reason that the normalized compression distance (NCD) can only be used with methods based on distance.

3.4.1 Distance methods

Distance methods are based on the initial calculation of a distance matrix [20], which contains information about the resemblance between the elements to be classified. The matrix is built by using a distance metric (for example the Levenstein distance) [7] and by calculating the similarity distance between all the elements, taken two by two. Afterwards, the clustering algorithm uses the information contained in the distance matrix to build a binary tree or to group the elements into distinct groups. The two best known type of distance methods are the hierarchical methods of clustering and the nonhierarchical methods of clustering.

In order to reach the goal of this work, only hierarchical methods were of interest. The disadvantage of the nonhierarchical methods consists in the fact that the number of classes must be known “a priori”, which is big inconvenient when talking about queries launched on search engines.

The hierarchical methods of clustering generate binary trees by unifying at each stage the two closest elements into a new partition.

The elements to be classified will be the leaves of the resulting tree, while the internal nodes represent a network, which models the relationship between the elements. The closest elements are always children of same parents within the tree. This type of methods does not produce a division in Q classes, but it creates a hierarchy of the partitions induced by the structure of the resulted binary trees. These trees are also called *dendograms*.

The hierarchical algorithm of clustering consists of the following stages:

Stage 1: Let us consider N elements to be classified. The initial distance matrix is built by calculating the distance between all N elements, two by two; the two closest elements are identified; a new element is formed by incorporating the two found elements. At this point there is a new partition in (n-1) classes.

Stage 2: A new reduced distance matrix is formed by calculating the distance between the (N-2) remaining elements after the aggregation of the two closest elements. The new distances can be calculated by using various formulas. Let us consider x, y, z three elements where x and y were the closest elements in the initial distance matrix and z is another distinct element. A new element formed by aggregating x and y is designated as h.

With these notations, the following formulas are considered for calculating the new distances between the elements:

- Minimum jump distance ("single linkage"): $d(h, z) = min\{d(x, z), d(y, z)\}$
- Maximum jump distance: $d(h, z) = max\{d(x, z), d(y, z)\}$
- Average distance: $d(h, z) = \{d(x, z) + d(y, z)\}/2$

The new two closest elements in the distance matrix are found and aggregated to form a new element. Thus, a new partition in (n-2) classes is created.

Stage 3: New distances are calculated and the process is reiterated until the distance matrix reaches the dimension one. The most popular algorithm of the hierarchical clustering is UPGMA ("Un-weighted Pair Group Method with Arithmetic Mean") [7]. It represents the reference method to which all new clustering methods are compared.

3.4.2 Quadruplet methods

This family of methods gained its name from the structures called topologies of quadruplets (or simply quadruplets). A topology of quadruplet is a binary tree with 4 internal leaves and 2 nodes. These trees present some interesting properties which can be exploited in the process of clustering. Given a set of 4 elements u, v, w, x there are always 3 possible distinct topologies of quadruplets which include

these four elements in various configurations: uv|xw, ux|vw and uw|xv (see figure 1).

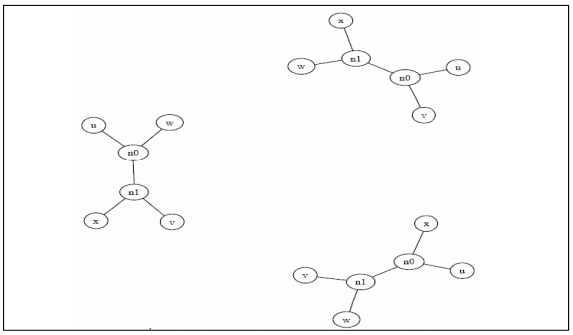

Fig. 1. The possible topologies of quadruplets

A set of quadruplet topologies can be aggregated into a tree of quadruplets, which preserves the properties of the quadruplet topology:

- n leaves and (n-2) internal nodes
- each internal node has exactly 3 neighbors
- each leaf has exactly one neighbor (his/her parent)
- the tree is not rooted.

The methods of quadruplets are usually implemented in two stages [9] called the stage of inference and the stage of recombination or aggregation. The most known quadruplet methods are "Addtree" and "Quartet Puzzling".

4 Experimental results

In order to test the performances of clustering by compression we have conducted a number of tests, comparing the clustering obtained using with the clustering obtained using other metrics. In what follows, we will present the results obtained for one of the tests, in order to prove the better performance of the NCD compared to traditional metrics.

The quality of a clustering solution will be determined by analyzing the hierarchical tree that is produced by a particular clustering algorithm using the *FScore measure*. Given a particular class L_r of size n_r and a particular cluster S_i of size n_i , suppose n_{ri} documents in the cluster S_i belong to L_r , then the FScore of this class and cluster is defined to be:

$$F(L_r, S_i) = \frac{2 * R(L_r, S_i) * P(L_r, S_i)}{R(L_r, S_i) + P(L_r, S_i)} \qquad (6)$$

where $R(L_r, S_i) = \frac{n_{ri}}{n_r}$ is called the recall value for the class L_r and the cluster S_i and $P(L_r, S_i) = \frac{n_{ri}}{n_i}$ is called the precision value for the class L_r and the cluster S_i.. Roughly, the precision answers the question: "How many of the documents in this cluster belong there?", whereas the recall answers the question: "Did all of the documents that belong in this cluster make it in?"

The FScore of the class *Lr* is the maximum FScore value obtained for that class:

$$F(L_r) = \max_{S_i} F(L_r, S_i) \qquad (7)$$

The FScore of the entire clustering solution is defined as the sum of the individual FScore for each class, weighted according to the class size:

$$FScore = \sum_{r=1}^{c} \frac{n_r}{n} F(L_r) \qquad (8)$$

where c is the number of classes. A perfect clustering solution will be the one in which every class has a corresponding cluster containing exactly the same documents in the resulting clustering. In this case, the FScore will be one. The higher the FScore value, the better the clustering solution is.

The data set we have used for our test comprises the following files:

1.conference1_description.pdf
2.conference2_description.pdf
3.conference3_description.pdf
4.OrganizingWeb.doc - organizing Web results byclustering by compression
5.OrganizingWeb.pdf – file 4 in PDF format
6.Clustering.doc – clustering by compression
7.clustering.eml – the content of file 6 sent by email
8.SOSFriends.ppt – Power Point presentation describing a software application for the financial management of an organization
9.SOSFriends.pdf – file 8 in PDF format
10.SOSFriends.pps – file 8 in PPS format
11.albnegru.jpg – file containing a picture in JPG format
12.albnegru.doc – Microsoft Word file containing picture 11
13.albastru.jpg – file containing a picture in JPG format
14.albastru.doc – Microsoft Word file containing picture 13
15.verde.jpg – file containing a picture in JPG format
16.verde.doc – Microsoft Word file containing picture 15

17.XML.txt – text document containing a short description of XML
18.xml.eml – the content of file 17 sent by email
19.xml_syntax.txt – text document describing theXML syntax

In order to obtain the classification tree, we have used the BZIP2 compression algorithm and the UPGMA clustering method. The reason for this choice resides in our previous experiences with the clustering by compression procedure, when the best clustering results were obtained with this combination of algorithms.

The pre-defined structure, the clustering we expect to receive, is the following:

Class 1:
- conference1_description.pdf
- conference2_description.pdf
- conference3_description.pdf
- OrganizingWeb.doc
- OrganizingWeb.pdf
- Clustering.doc
- clustering.eml

Class 2:
- XML.txt
- xml.eml
- xml_syntax.txt

Class 3:
- albnegru.jpg
- albnegru.doc
- albastru.jpg
- albastru.doc
- verde.jpg
- verde.doc

Class 4:
- SOSFriends.ppt
- SOSFriends.pdf
- SOSFriends.pps

When calculating the FScore, each of these groups will be the classes. The groups returned by a clustering algorithm in the following subsections will be the clusters.

4.1 NCD – the normalized compression distance

The result obtained using the newly defined metric, NCD, is the following:

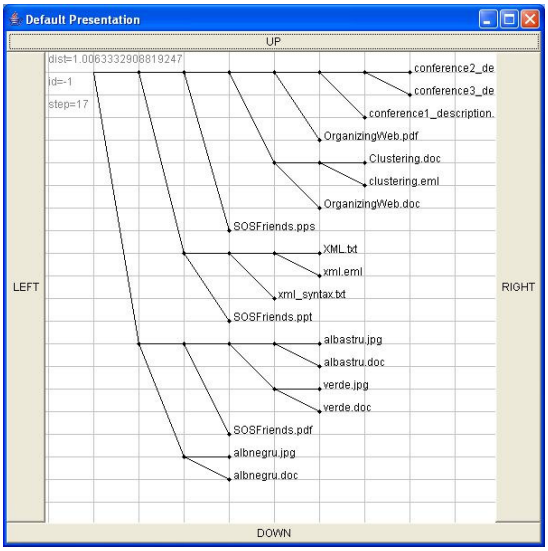

Fig. 2. NCD clustering tree

The structure we have obtained after applying the
clustering algorithm is the following:

Cluster 1:
- conference1_description.pdf
- conference2_description.pdf
- conference3_description.pdf
- OrganizingWeb.doc
- OrganizingWeb.pdf
- Clustering.doc
- clustering.eml

Cluster 2:
- XML.txt
- xml.eml
- xml_syntax.txt
- SOSFriends.ppt
- SOSFriends.pps

Cluster 3:
- albnegru.jpg
- albnegru.doc
- albastru.jpg
- albastru.doc
- verde.jpg
- verde.doc
- SOSFriends.pdf

The FScore for this clustering solution is 0.85, which is rather high. If we analyze the solution from a human expert point of view, we notice that the first cluster is entirely correct. Also, the documents belonging to the second and third classes were correctly identified. The only problem is that the fourth class does not appear, although two of the

documents from the fourth class were clustered together.

4.2 NKLD – the normalized Kullback-Leibler distance

Mutual information is a measure of the shared information between two variables. The greater the mutual information, the more similar two variables are. The opposite of the mutual information is the relative entropy, also known as the Kullback-Leibler's distance or divergence [22].

Let a discrete distribution have probability function p_k, and let a second discrete distribution have probability function q_k. Then the relative entropy of p with respect to q, also called the Kullback-Leibler distance, is defined by:

$$d = \sum_k p_k \log_2 (\frac{p_k}{q_k}) \quad (9)$$

Although $d(p, q) \neq d(q, p)$, so relative entropy is therefore not a true metric, it satisfies many important mathematical properties.

Relative entropy is a very important concept in quantum information theory, as well as statistical mechanics.

The structure we have obtained after applying the clustering algorithm is the following:

Cluster 1:
- conference1_description.pdf
- conference3_description.pdf
- OrganizingWeb.doc
- OrganizingWeb.pdf
- Clustering.doc
- clustering.eml
- SOSFriends.pps

Cluster 2:
- XML.txt
- xml.eml
- xml_syntax.txt
- SOSFriends.ppt
- conference2_description.pdf

Cluster 3:
- albnegru.doc
- albastru.doc
- verde.doc

Cluster 4:
- SOSFriends.pdf
- albnegru.jpg
- albastru.jpg
- verde.jpg

The FScore for this clustering solution is 0.41, so the quality of this clustering solution is significantly lower than the quality of the clustering solution obtained using the NCD. Using the NKLD, none of the clusters perfectly matches the predefined classes for the set of documents. Moreover, the last two clusters seem to be based on the type of document (file extension), rather than on content.

4.3 NLD – the normalized Levenshtein distance

In information theory, the Levenshtein distance or edit distance between two strings is given by the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution [21].

It can be considered a generalization of the Hamming distance, which is used for strings of the same length and only considers substitution edits.

The structure we have obtained after applying the clustering algorithm is the following:

Cluster 1:
- conference1_description.pdf
- conference3_description.pdf
- conference2_description.pdf
- SOSFriends.pdf
- SOSFriends.pps

Cluster 2:
- XML.txt
- xml.eml
- xml_syntax.txt
- SOSFriends.ppt
- OrganizingWeb.pdf
- clustering.eml

Cluster 3:
- albnegru.doc
- albastru.doc
- verde.doc
- Clustering.doc
- OrganizingWeb.doc

Cluster 4:
- albnegru.jpg
- albastru.jpg
- verde.jpg

The FScore for this clustering solution is 0.49, so the quality of this clustering solution is significantly lower than the quality of the clustering solution obtained using the NCD. Using the NLEVD, none

of the clusters perfectly matches the predefined classes for the set of documents. Moreover, the clusters seem to be based on the type of document (file extension), rather than on content.

To summarize, the statistics for the obtained results look like this:

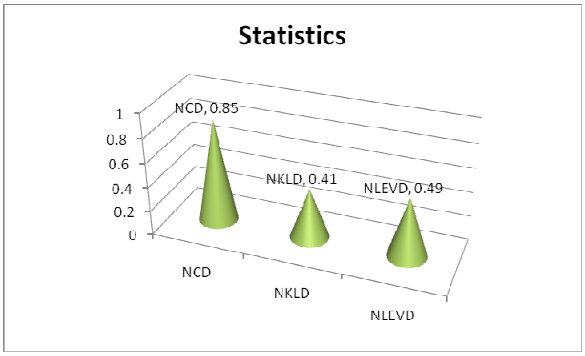

Fig. 3. Statistics

5 Conclusion and future work

In this paper, we have tested the results of a new clustering technique – clustering by compression – when applied to heterogeneous data. The clustering by compression procedure is based on a parameter-free, universal, similarity distance, the normalized compression distance or NCD, computed from the lengths of compressed data files (singly and in pair-wise concatenation).

In order to validate the results, we have compared the clustering solutions obtained with the NCD with clustering solutions obtained with other traditional metrics (the normalized Kullback-Leibler distance and the normalized Levenstein distance). In this paper, we have demonstrated the good results of clustering heterogeneous data using clustering by compression. We have made our demonstration using one set of well predefined data. In order to obtain the classification tree, we have used the BZIP2 compression algorithm and the UPGMA clustering method. Finally, we calculated some quality indices, namely the FScore for the clustering solutions resulted. This quality measure takes values between 0 and 1. The closer it is to 1, the better the quality of the clustering structures. As shown in section 4, for our heterogeneous data set, we have obtained a value of 0.85 when using clustering by compression. Therefore, this indicates a high degree of similitude between the clustering structure obtained as a result of the clustering by compression technique and the predefined structure designed according to our intuition.

The results of these tests encourage us to proceed to including the clustering by compression procedure into a framework for clustering heterogeneous data, which could be further tested for clustering heterogeneous web data or for integration into a document management system.

Moreover, we intend to use the MD5 hashing technique [26] as a clustering algorithm and test the performances of clustering by compression in this case.

References:

[1] D. Boley, M. Gini, R. Gross, E.H. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore, "Partitioning-based clustering for web document categorization," *Decision Support Systems 27:329-341*, 1999.

[2] O. Zamir and O. Etzioni, "Web document clustering: A feasibility demonstration," *Proceedings of SIGIR '98*, pp. 46--53, 1998.

[3] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. Morgan Kaufman, 2001.

[4] Z. Su, Q. Yang, H. J. Zhang, X. Xu and Y. H. Hu, "Correlation-based Document Clustering using Web Logs," In *Proceedings of the 34th Hawaii International Conference On System Sciences(HICSS-34)*, January 3-6, 2001.

[5] D. Beeferman and A. Berger, "Agglomerative clustering of a search engine query log," In *Proceedings of the Sixth ACM SIGKDD*, pp. 407--416, 2000.

[6] L. H. Ungar and D. P. Foster, "Clustering methods for collaborative filtering," In *Proceedings of Workshop on Recommendation Systems at the 15th National Conference on Artificial Intelligence*, 1998.

[7] T. Hofmann, J. Puzicha, and M. Jordan, "Learning from dyadic data," In *Advances in Neural Information Processing Systems 11 (NIPS)*, Cambridge, MA, MIT Press, 1999.

[8] T. Hofmann, J.Puzicha, and J.M. Buhmann. "Unsupervised texture segmentation in a deterministic annealing framework," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):803--18, 1998.

[9] T. Hofmann, "The cluster-abstraction model: Unsupervised learning of topic hierarchies from text data," In *Proceedings of 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, pp. 682-687, 1999.

[10] A. Popescul, L. H. Ungar, D. M. Pennock, and S. Lawrence, "Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments," in *Proceedings of the*

Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI-2001), Seatle, WA, August 2001.

[11] F. C. Pereira, N. Tishby, and L. Lee, "Distributional clustering of English words," In *Proceedings of the 30th Annual Meeting of the ACL*, pp. 183--190, 1993.

[12] D. Gibson, J. Kleinberg, and P. Raghavan, "Clustering categorical data: An approach based on dynamical systems," In *Proceedings of the 24th International Conference on Very Large Databases (VLDB)*, pp. 311–323, New York City, New York, August 24-27, 1998.

[13] S. Guha, R. Rastogi, and K. Shim, "ROCK: A robust clustering algorithm for categorical attributes," In *Proceedings of the IEEE International Conference on Data Engineering*, Sydney, March 1999.

[14] S. Guha, R. Rastogi, and K. Shim K,CURE: An Efficient Clustering Algorithm for Large Databases. In *Proceedings of the ACM SIGMOD Conference*, 1998.

[15] S. Theodoridis and K. Koutroubas, *Pattern Recognition*. Academic Press, 1999.

[16] M.J.A. Berry and G. Linoff, *Data Mining Techniques For Marketing, Sales and Customer Support*. John Wiley & Sons, Inc., USA, 1996.

[17] M.U. Fayyad, G. Piatesky-Shapiro, P. Smuth , R.Uthurusamy, *Advances in Knowledge Discovery and Data Mining*. AAAI Press, 1996.

[18] Rudi Cilibrasi, Paul M.B. Vitanyi, Clustering by compression. *IEEE Transactions on Information Theory*, Vol. 51, No. 4, pp 1523–1545, 2005.

[19] Peter Grunwald and Paul Vitanyi. *Shannon Information and Kolmogorov Complexity*, 2004.

[20] C. Bennett et al., Information Distance, *IEEE Transactions on Information Theory*, Vol. 44, No. 4, pp. 1407-1423

[21] Levenshtein Distance : http://knowledgerush.com/kr/encyclopedia/Levenshtein_distance/

[22] Kullback - Leibler Distance : http://www.cis.hut.fi/aapo/papers/NCS99web/node26.html

[23] Gang Zhao, Xiao-Hui Kuang, Yong Guo, Associativity based Clustering Algorithm in Mobile Ad Hoc Networks, *Proceedings of the 11th WSEAS International Conference on COMPUTERS*, 2007

[24] Shankar Kambhampat, Asrita Chetan Avasarala, Murty Eranki , Architecting for the Next Generation Business Applications, *Proceedings of the 10th WSEAS International Conference on COMPUTERS*, 2006

[25] Zaigham Mahmood, Architectural Representations for Describing Enterprise Information and Data, *Proceedings of the 10th WSEAS International Conference on COMPUTERS*, 2006

[26] MD5 Hashing Algorithm: http://en.wikipedia.org/wiki/MD5