CMST 21(4) 241-249 (2015) DOI:10.12921/cmst.2015.21.04.008

Numerical Representation of Symbolic Data

B. Kozarzewski

University of Information Technology and Management
ul. H. Sucharskiego 2, 35-225 Rzeszów, Poland
E-mail: bkozarzewski@wsiz.rzeszow.pl

Received: 09 November 2015; revised: 17 December 2015; accepted: 21 December 2015; published online: 29 December 2015

Abstract: A method of direct numerical representation of symbolic data is proposed. The method starts with parsing a sequence into an ordered set (spectrum) of distinct, non-overlapping short strings of symbols (words). Next, the words spectrum is mapped onto a vector of binary components in a high dimensional, linear space. The numerical representation allows for some arithmetical operations on symbolic data. Among them is a meaningful average spectrum of two sequences. As a test, the new numerical representation is used to build centroid vectors for the *k*-means clustering algorithm. It significantly enhanced the clustering quality. The advantage over the conventional approach is a high score of correct clustering several real character sequences like novel, DNA and protein.

Key words: symbolic sequence, numerical representation, distance measure, clustering

I. INTRODUCTION

I. 1. Related work

Sequence comparison allows for unveiling similarity relationships between various sorts of data. In some applications, such as genome or protein analysis, data objects are sequences consisting of a large number of symbolic (character) entities and are not necessarily of the same length. A comparison of any objects requires a certain measure of distance or similarity. There are several methods to define such a measure, including the

Hamming distance. It is a number of character positions in which two sequences differ. The Hamming distance is not flexible enough. Sequences have to be of equal length, and there is generally no fixed correspondence between their character positions.

Alignment based measures[1]. An alignment of two sequences is a scheme in which two sequences are placed one above the other, and the spaces are inserted so that similar characters from the sequences line up and the sequences have the same size. Proximity measure is the smallest number of changes (deletions, insertions, substitutions, shifts) that are necessary to match all characters in both sequences.

Graphical representations [2]. It is an approach in which sequences are compared on the basis of a set of structural sequence invariants. The idea is to represent a sequence by a vector typically devised from a graphical representation of the DNA. In this way, the similarity between two sequences can be measured by e.g. cosine of the angle between the corresponding vectors.

Word or k-mer frequencies [3], [4]. The method compares mathematical invariants of sequences rather than the sequences themselves. It starts with the mapping of sequences onto vectors defined by the counts of each k-mers. The sequence is summarized by n-tuples consisting of all k-mer counts, where k typically ranges from 2 to 8. The distance between two sequences is determined by e.g. the Euclidean distance between the corresponding vectors.

Number of common words. In the papers [5] by the present author a novel word representation for a symbolic sequence, based on specific parsing of the sequence into a set of words is proposed. For the sake of the readers' comfort it seems worth recalling the algorithm.

Let us suppose there is a sequence C of symbols $c_1,\ c_2,\ \ldots,\ c_n$. Let us assume S_t is a set of words ob-

tained so far and the first symbol of the new word w is c_i. The next word is formed as a result of a specific procedure of appending symbol c_i by the following symbols in three steps.

Step1. String $Q = c_i$ is neither periodic nor chaotic because there is only one symbol in it. Hence it has to be appended by the next symbol. Appending is continued until some symbol c_{i+j+l} repeats one of the symbols, say k -th, in the string $Q = c_i, \ldots, c_{i+j}$.

Step 2. Let $P = c_k$ and $R = c_{i+j+1}$, so far they are equal. Both strings are appended $P = c_k c_{k+1}$, $R = c_{i+j+1} c_{i+j+2}$ and so on, until they become different. Then string Q found in Step 1 is appended by string R, and the new string is $Q = QR$.

Step 3. Set S_t of words is searched for the presence of string Q. If string Q is found, it is appended by the following (next to the last symbol of Q) symbol of C becoming $Q = Qc_{i+j+k+1}$. Appending is continued until some string $Qc_{i+j+k+l}$ does not replicate any word from S_t. The string $w = Qc_{i+j+k+l}$ becomes the new word of the spectrum representing sequence C. It may happen that one or two last symbols of C cannot be processed by the above replication. They make a new word or can be discarded.

The code of the parsing algorithm is available on request. The result of the sequence decomposition is a set of ordered, distinct and non-overlapping words which will be called the word spectrum S of the sequence C. The similarity measure between two symbolic sequences C_1 and C_2 represented by their spectra S_1 and S_2, respectively

$$simi\,(C_1, C_2) = \frac{2length\,(intersection\,(S_1, S_2))}{length\,(S_1) + length\,(S_2)}$$

has been introduced in [7]. The value of similarity varies between 0 when the spectra are disjoint sets and 1 when sequence C_1 and C_2 are mutual copies.

The similarity matrix represented by an n-by-n table stores a collection of similarities that are available for all pairs of n sequences. Elements of i-th row of the matrix are similarities between the i-th data vector and other data vectors make a numerical vector which can be considered as the indirect numerical representation of the original i-th sequence data. Now the arithmetic operations like the mean vector of several sequences or distances between them become possible. However, the indirect numeric representation which is defined there is meaningful within a given set of sequences: it is context dependent. For example, the distance between two definite sequences being a member of certain group usually changes when the sequences become members of another group.

There is often a need for clustering a set of symbolic (character) sequences.The aim of clustering is to assign a set of objects into groups so that the objects in the same group are more similar to each other than to those in other groups. Clustering symbolic sequences is more challenging than clustering numeric data because there is no natural measure of distance between the sequences. Instead, another proximity measure that quantifies how similar are two sequences has to be used. There are many clustering methods and algorithms, includinf the hierarchical agglomerative k-means algorithm that is often used (see, for example [6] and quotes therein). To work efficiently, the algorithm needs a numerical data matrix, a number of clusters and an initialising set of cluster centres – centroids as input. Actually, cluster initialisation remains the biggest drawback of the clustering algorithms. In Ref. [7] the sets of most similar pairs, fours or eights sequences as the initialising set were discussed. The average vector of each set was considered as the starting centroid location. The performed tests did not always lead to satisfactory results, mainly due to the lack of a proper distance measure.

I. 2. Present paper contribution

First of all, a new direct numerical representation of any symbolic sequence is introduced. The sequence via its word spectrum is related to vector in many dimensional binary vector spaces. The vector depends solely on the sequence itself. Introduced are some arithmetical operations on binary vectors, including the distance between two vectors, defined as the normalized number of nonzero components of their logical sum and the average or mean value of several word spectra.

To answer how useful the new numerical representation can be, it is employed to construct centroid vectors for the k-means clustering algorithm. Usually the clustering is performed in one step, but in the present paper it takes several steps. The quality measure of a single cluster is the sum of silhouette values per length of the cluster. The silhouette of the sequence says how similar the sequence is to sequences in its own cluster compared to the sequences in other clusters. The silhouette value ranges from -1 to $+1$, the higher silhouette value, the better the cluster is. More details concerning the plot function *silhouette* are available in the Matlab® documentation.

The starting centroids are mean vectors of part (e.g. half) of the least distant pairs. From output clusters a set of best clusters is selected and the mean vector of each of them becomes the centroid input to the k-means algorithm in the next step. In the vicinity of the expected number of clusters, steps may differ by one. After each step of the clustering process is completed, the number of clusters and the sum of silhouette values of all clusters is collected. The recommended clusters for making centroids to the next step are that of the highest sum of the silhouette values.

The algorithm is applied to 600 fragments of each of 5000 characters long from six novels, 400 mitochondrial DNA and mitochondrial COX1 gene coding protein sequences, and 1234 haplogroups of human mitochondrial DNA sequences. It is demonstrated that the quality of clustering with the use of the direct numerical representation is significantly higher than with the use of indirect representation.

II. RESULTS

II. 1. Numerical representation of word spectrum

It is assumed that a sequence was decomposed into a set of words according to the algorithm proposed in Ref. [5]. When a set C of sequences is analysed, a union set of words of all sequences of the set may be very useful. The union set includes all the words present in the spectra of all the sequences but with no repetitions. Let a set C consist of n sequences c_1, c_2,..., c_n, their word spectra are S_1, S_2,...,S_n, respectively. The union set U of the set of all spectra can be found in the following 3 steps. In the beginning, $U = S_1$, next the set difference of two vectors S_2 and U ($diff(S_2, U)$) of words is found. Former set U appended by $diff(S_2, U)$ becomes a new union and so on, until there is no spectrum left. The union U obtained in this way is the ordered set (list) of words. Let p mean the length of union set (number of words in U). Let every word w_i from U be represented by p dimensional unit vector $\boldsymbol{e}_i$ of i-th component 1 and all other zeros. The union set can be mapped onto linear, p dimensional space Σ spanned by the set of all vectors $\boldsymbol{e}_i$. The set of unit vectors creates the Cartesian coordinate system in the space. Any arbitrary subset of words from U is represented by one vector $\boldsymbol{x}$ being a linear combination of unit vectors $\boldsymbol{e}_i$ with coefficients 0's or 1's. It means that all the spectra related by permutation of their words are mapped onto single vector $\boldsymbol{x}$. So do all the sequences whose spectra are permutation of the same set of words. One can call $\boldsymbol{x}$ a logical or binary vector. For example, *length*(S_1) components of vector $\boldsymbol{x_1}$ representing spectrum S_1 are 1's and all others are 0's. Note that for any binary vector $\boldsymbol{x}$ the squared Euclidean length $\left|\boldsymbol{x}^2\right| = |\boldsymbol{x}|$ equals the number of nonzero components of $\boldsymbol{x}$.

Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be vectors of Σ. The following functions can be defined on them:

- bit wise sum:

$$\boldsymbol{x}+\boldsymbol{y}=((x_1+y_1)\mathrm{mod}\ 2,\ldots,(x_N+y_N)\mathrm{mod}\ 2) \quad (1)$$

- bit wise product:

$$\boldsymbol{x}\cdot\boldsymbol{y} = (x_1y_1,\ldots,x_Ny_N), \quad (2)$$

- Hamming distance:

$$|\boldsymbol{x}-\boldsymbol{y}|, \quad (3)$$

- mean vector:

$$\mathrm{mean}(\boldsymbol{x},\boldsymbol{y}), \quad (4)$$

binary vectors are added with subsequent shift right one bits of the resulting vector, e.g. $mean(01_2, 11_2) = 100_2 \rightarrow 10_2 = 2_{10}$, the measure is simply the arithmetic mean of two numbers rounded to the nearest integer less than or equal to mean,

- similarity:

$$simi(\boldsymbol{x},\boldsymbol{y}) = \frac{2\,|\boldsymbol{x}\cdot\boldsymbol{y}|}{|\boldsymbol{x}|+|\boldsymbol{y}|}, \quad (5)$$

- distance:

$$dist\,(\boldsymbol{x},\boldsymbol{y}) = 1 - simi(\boldsymbol{x},\boldsymbol{y}), \quad (6)$$

which is the dual of the similarity measure:

Similarity defined in [5] coincides with one given by Eq. (5). In (5) $\boldsymbol{x}$ and $\boldsymbol{y}$ correspond to a numerical representation of the spectra. The distance (6) appears to be a normalized Hamming distance between binary vectors (not between characters), $dist\,(\boldsymbol{x},\boldsymbol{y}) = |\boldsymbol{x}-\boldsymbol{y}|\,/\,(|\boldsymbol{x}|+|\boldsymbol{y}|)$**.** It satisfies triangle inequality $dist\,(\boldsymbol{x},\boldsymbol{y}) \leq dist\,(\boldsymbol{x},\boldsymbol{z}) + dist\,(\boldsymbol{y},\boldsymbol{z})$.

In the subsequent discussion a centroid of the cluster will be defined as the mean of several binary vectors. Therefore, the equation (4) has to be generalized, which is straightforward for a set of 2^n vectors otherwise is not trivial. However, in the present application $\boldsymbol{x}$ are sparse binary vectors (more than 90% of their components are 0's). In that case the bit wise sum is a good approximation for the mean vector.

The order of words in a union set built according to the algorithm described above depends on the order of sequences in C (and the order of corresponding spectra). Any change in the order of words means their permutation. But any permutation of the words in a set union corresponds to the unitary transform of the coordinate system in the space Σ. Vector norm, distance and similarity between vectors are invariants of unitary transforms. Therefore, the order of words in a union set does not matter as far as distance or similarity is considered.

II. 2. Clustering algorithm

The clustering method proposed in the present paper is a combination of the hierarchical and partitioning method. It starts with a large set of the least distance pairs of the n data, then consecutively uses the standard k-means algorithm. The software from Matlab® is extended to the procedure of adapting the number of clusters, set of centroids and validation function at each step of the clustering process.

The preliminary steps are as follows:

i. Every sequence of the set to be clustered is partitioned into words, their list represents a spectrum of the sequence.
ii. The union of spectra of all sequences is found and space Σ of dimension p is constructed.
iii. All spectra are transformed into binary vectors in Σ and assembled as matrix X of n rows and p columns.

The proper clustering algorithm includes the following steps:

i. The list of all pairs of sequences, sorted in an ascending distance is collected. Next, starting from the top, the pruning of the list is done to leave only the pairs of unique indices. The subset of nc least distant pairs is selected.

ii. The mean vector of each pair is found and assumed to be the centroid of the starting cluster. All centroids are assembled as matrix M of nc rows and p columns.

iii. The Matlab® *kmeans* function with X, nc, M, and string 'cosine' as inputs for data set, number of clusters, starting centroids and distance measure, respectively, is used. The 'cosine' distance measure is selected from the valid measures, as it is very close to the distance measure defined by Eq.(3).

$$[C, idx] = kmeans(X, nc, \ 'start', M, 'distance', 'cosine', \ 'emptyaction', 'singleton').$$

The function returns, vector idx containing the cluster indices of each data point and the nc cluster centroid locations in the nc-by-p matrix C. The Matlab® *silhouette* function

$$s = silhouette(X, idx, \ 'cosine'),$$

with X, idx and 'cosine' as inputs, returns the silhouette value of every data, from which the within-cluster sum of the silhouette value is obtained.
All the results are assembled into classes. The class k is a structure including nc vectors of the cluster centroid location, cluster length, within-cluster sum of silhouette values per cluster length, the set of data indices which are members of the cluster. The classes are sorted by the silhouette value per cluster length in a decreasing order.

iv The next step in the clustering procedure consists of selecting a set of the best classes (i.e. of the highest silhouette value per cluster size). The corresponding centroids are inputs to Matlab® *kmeans* function. The output is the next generation set of classes.

v At each step the number of clusters – nc and two sums of the positive and negative silhouette values of all clusters – ssn and ssp are recorded. The correct number of clusters is expected at the nc value corresponding to the lowest ratio value of $|ssn/ssp|$.

However, several local minima of $|ssn/ssp|$ at the same nc may happen, and the algorithm does not necessarily arrive the best of them. It is possible that reassigning some points to a new cluster would provide a better solution. In order to find it, the *kmeans* function with optional 'replicates' parameter is proposed by Matlab®

$$[idx, nc, sumdist] = \ kmeans(X, \ nc, \ 'dist', \ 'cosine', \ 'display', \ 'final', \ 'replicates', \ 5).$$

The final solution that *kmeans* function returns is the one with the lowest total sum of distances, and from silhouette plot

$$[silh, h] \ = silhouette(X, idx, 'cosine'),$$

the lowest $|ssn/ssp|$ ratio follows. Besides, a silhouette plot for this solution indicates whether the clusters are or are not well separated.

III. EXAMPLES OF APPLICATION

The details and efficiency of the proposed algorithm are demonstrated on four examples. To verify the correctness of the clustering algorithm results real symbolic sequences of the known assignment were used. The experimental datasets used in this research are from the Project Gutenberg (http://www.gutenberg.org) (novels) and the GenBank database (http://www.ncbi.nlm.nih.gov) (DNA and protein sequences). The numerical experiments were conducted on a PC with eight core processor (2.9 GHz) and 8 GB memory. The used software was from the Matlab® system.

III. 1. Novels

The data sequences come from six novels: "Pride and Prejudice" and "Emma" by Jane Austen, "Wuthering Heights" by Emily Brontë, "Lord Jim" by Joseph Conrad, "David Copperfield" by Charles Dickens and "Ulysses" by James Joyce. All were downloaded from the Project Gutenberg database. In the following acronyms A1, A2, B, C, D, and J are used for them.

In the beginning all sort of titles and introductions were removed from the texts. All the uppercase characters were converted to lower case ones to make the alphabet shorter. Next, 100 segments, each of 5000 character long, were selected from each novel. The segments were selected consecutively starting from the beginning of the text and indexed from 1 to 100 and 101 and 200 for Jane Austen books, 201 to 300 for Emily Brontë book and so on, and stored as the data sets. Each segment was partitioned into words, their list represents the spectrum of the segment. Then the union of all words was found, it counts $p = 225518$ items. The first several ones are: 'itis', 'atruth', 'universally', 'acknowledged', 'thata', 'singlemani', and so on.

All the spectra were transformed into vectors and assembled as matrix X of 600 rows and p columns. Then a list of all the pairs sorted in an ascending distance was built. Next, starting from the top of the list the pruning of the list was done to leave only 300 pairs of unique indices. The first step of the clustering algorithm consists of selecting a subset of nc least distant pairs, in my experiment nc =100 were chosen. The mean vector of each pair was found and assumed the centroid of the starting cluster set. From nc vectors of cluster centroid location 90 were selected and used as input arguments to the *kmeans* function of Matlab®. From 90 output centroid locations 80 were selected and used as input arguments to *kmeans* and so on, until the number of clusters dropped down to $nc = 20$. All the time sum of silhouette values was at a low level, about 1.1. Then the step of nc decreasing was set

to 1. At each step the number of clusters nc and the absolute ratio of sums of positive and negative silhouette values of all clusters $|ssn/ssp|$ was recorded.

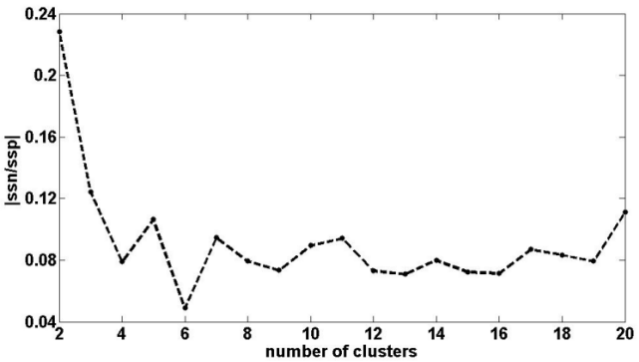

Fig. 1. The most interesting part of $|ssn/ssp|$ dependence on nc

It follows from the plot that there is a global minimum at $nc = 6$ and it is the one with the lowest total sum of distances over all the replicates.

Tab. 1. Distribution of data among 6 clusters

cluster	1	2	3	4	5	6
A1	1	99				
A2	97	3				
B			100			
C					100	
D				100		
J			1	1	3	95

From Table 1 it follows that nine data are incorrectly assigned, so the accuracy of clustering is 98.5 %. Also the silhouette plot shows that clusters are not perfectly separated.

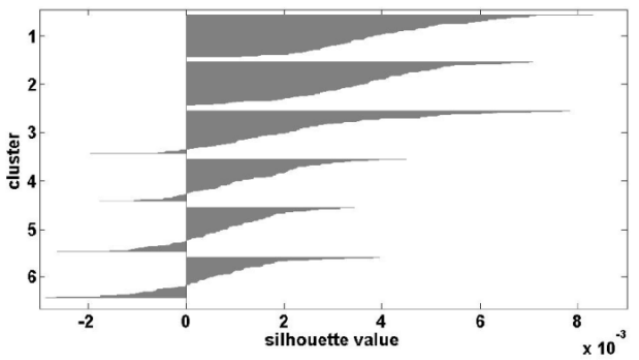

Fig. 2. The silhouette plot for six clusters

There are several local minima of $|ssn/ssp|$. One of them is at $nc = 4$. The resulting distribution of data among four clusters presented in Table 2

Tab. 2. Distribution of data among four clusters

cluster	1	2	3	4
A1	100			
A2	100			
B	1	96	3	
C		96	2	2
D			100	
J		1	1	98

The corresponding clusters form mainly from joining data A1 and A2 into one cluster and data B and C into another single cluster.

III. 2. Mitochondrial DNA sequences of 400 species

The used data set includes mitochondrial genomes of four Vertebrata: birds, fish, mammals and reptiles, (in the following acronyms B, F, M, and R are used respectively) each of 100 species, all downloaded from the GenBank database. Each sequence (of approximately 16000 nucleotides long) was partitioned into words, their list represents a spectrum of the sequence. The union of all words was found, its length is $p = 36941$. The first several words of the union are: 'GTCCA', 'TGTA', 'GCTTA', 'CAAC', 'AAAG', 'CATGAC', and so on. All the spectra were transformed into vectors and assembled as matrix X of 400 rows and p columns. As before, 200 least distant pairs of unique indices were selected and 60 least distant of them were used to build the starting centroids. Next, from *kmeans* output k centroid locations, k – 10 was selected and used as input arguments to *kmeans* until the number of clusters dropped down to $nc = 20$. Then the step of nc decreasing was set to one. The set of pairs, nc and corresponding ratio of $|ssn/ssp|$ was used to make the plot.

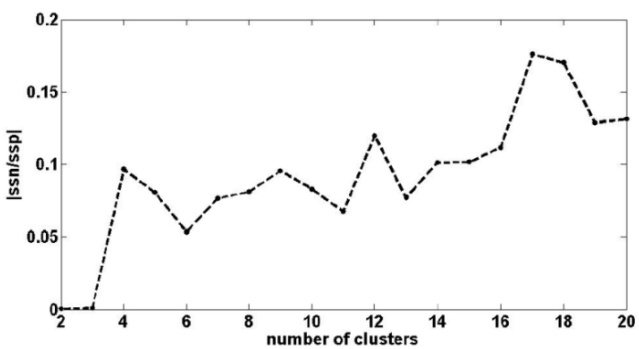

Fig. 3. The ratio $|ssn/ssp|$ versus number of clusters

The $|ssn/ssp| = 0$ at $nc = 2$, it suggests two clusters as the best solution. The *kmeans* function with optional 'replicates' parameter rearranges several data resulting in the distribution shown in Table 3.

Tab. 3. Distribution of data among two clusters

cluster	1	2
B		100
F	16	84
M	94	6
R	45	55

The silhouette plot (Fig. 4) shows that clusters indeed are well separated.

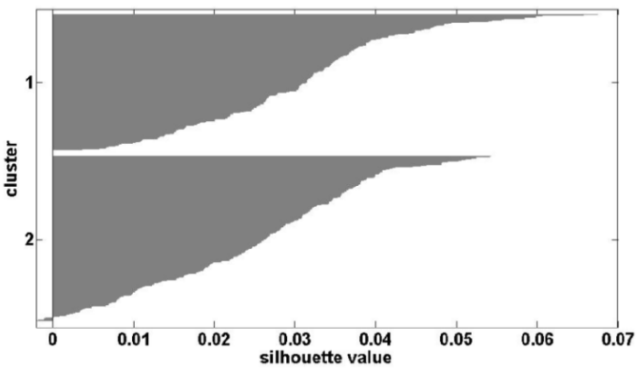

Fig. 4. The silhouette plot for two clusters

There are also several local minima of $|ssn/ssp|$. One of them is at $nc = 6$. The resulting distribution of data among six clusters after some data rearrangements is presented in Table 4.

Tab. 4. Distribution of data among six clusters

cluster	1	2	3	4	5	6
B					100	
F		6	6		2	86
M		94			6	
R	24		54	21	1	

This time the silhouette plot (not shown) indicates that clusters are not separated as well as for $nc = 2$. The number of incorrectly assigned data is 21 so the accuracy of clustering is 95%. The reptile sequences are distributed among four clusters, so are the fish sequences.

III. 3. Mitochondrial COX1 gene encoded protein sequences of 400 species

The data set comprises 400 protein sequences of birds, fish, mammals and reptiles, each of 100 species, all downloaded from the GenBank database. Each sequence was partitioned into words. Their list represents a spectrum of the sequence. The union of all words is $p = 2735$ long. The plot of function $|ssn/ssp|$ versus nc is given in Fig. 5.

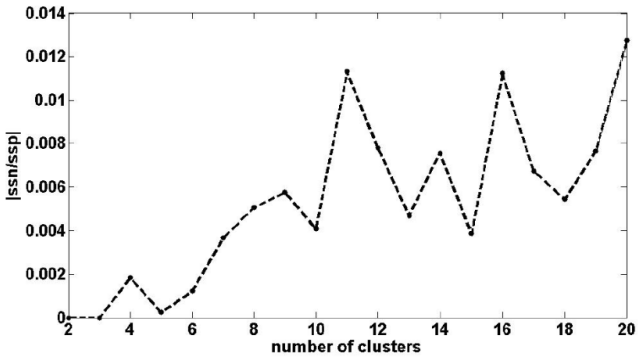

Fig. 5. The ratio $|ssn/ssp|$ versus number of clusters

Like before the $|ssn/ssp| = 0$ at $nc = 2$, it suggests two clusters as the best solution. The *kmeans* function with optional 'replicates' parameter rearranges several data reduces the total sum of distances. The final distribution of data is given in Table 5.

Tab. 5. Distribution of data among two clusters

cluster	1	2
B	100	
F		100
M		100
R	45	55

The silhouette plot for this solution indicates very good clusters separation as follows from Fig. 6.

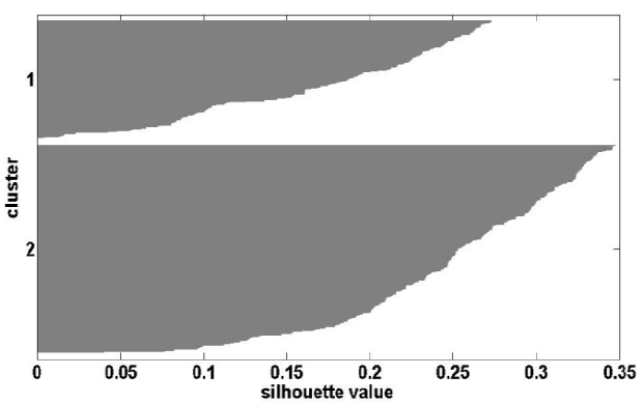

Fig. 6. The silhouette plot for two clusters

There is also a relatively deep local minimum of $|ssn/ssp|$ at $nc = 5$. The resulting distribution of data among five clusters after some data reassignments is presented in Table 6.

Tab. 6. Distribution of data among five clusters

cluster	1	2	3	4	5
B					100
F				100	
M			100		
R	16	29	54	1	

However, the silhouette plot indicates that clusters separation becomes slightly worse (in particular cluster #3) than when the data were distributed among two clusters. As in the case of nucleotide sequences, protein sequences of reptile species are distributed among four clusters.

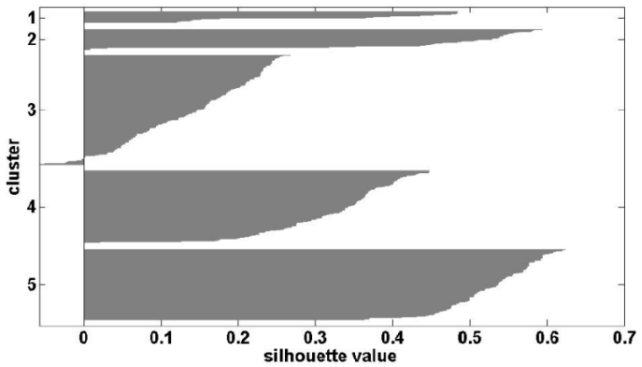

Fig. 7. The silhouette plot for five clusters

The distribution of data among ten clusters is also interesting as all birds, fish and mammals species protein sequences are assigned to their own clusters. The reptile sequences are distributed among other seven clusters (Tab. 7) as follows: all members of cluster #1 are species of chameleons native to the sub-Saharan Africa, Arabian Peninsula, Madagascar and Sri Lanka, cluster #2 – snakes and vipers, cluster #4 – crocodiles and caimans, cluster #5 – turtles, cluster #8 – large lizards, cluster #9 – lizards, cluster #10 – worm lizards, geckos, blind snakes.

Tab. 7. Distribution of reptile data among seven clusters

cluster	1	2	4	5	8	9	10
R	9	27	8	23	16	7	10

It follows from Fig. 8 that clusters one to eight are well separated. Cluster #9 is not well separated and cluster #10 is a mix of rather distant species.

III. 4. Mitochondrial nucleotides of H. sapiens haplogroups

The set DNA sequences including $n = 1234$ mitochondrial genomes of Homo sapiens were downloaded from the GenBank database. After partitioning each sequence into words, the union of length $p = 14875$ of words was obtained and binary vectors in p dimensional space Σ corresponding

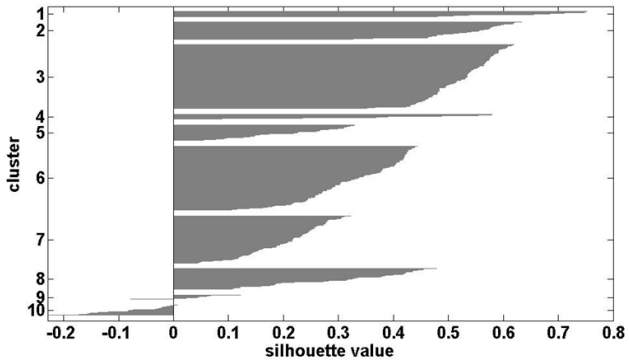

Fig. 8. The silhouette plot for ten clusters

to all the sequences were created and indexed from 1 to n. The starting 617 centroids were built on least distant pairs of unique indices.

The *kmeans* function returned *nc* = 559 clusters (some created clusters were empty and the corresponding sets became void). The $nc - 1$ clusters were used to build a new set of centroids and so on. At each step the values of $|ssn/ssp|$and *nc* were determined. The corresponding plot in a logarithmic scale is presented in Fig. 9.

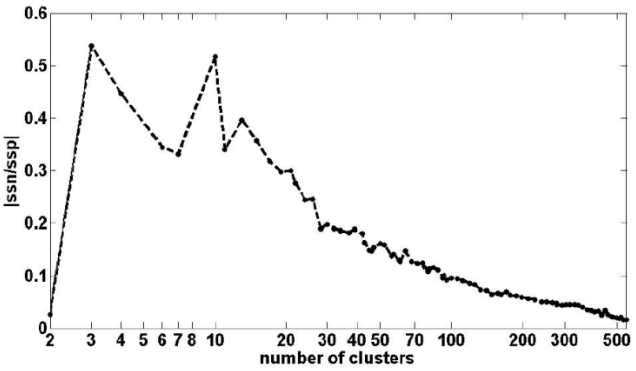

Fig. 9. The ratio $|ssn/ssp|$ versus number of clusters

It follows from the plot that there is a deep minimum at $nc = 2$. It corresponds to relatively well separated clusters. Cluster #1 consists of 27 sequences of haplogroups A out of 30 present in the data. However, the silhouette measure of a significant number of sequences assigned to the second cluster is negative, indicating that they are in the wrong cluster.

There are two less prominent minima at *nc* equal 7 and 11. Their silhouette plots show that the silhouettes of many sequences are negative, indicating that the clusters are not well separated.

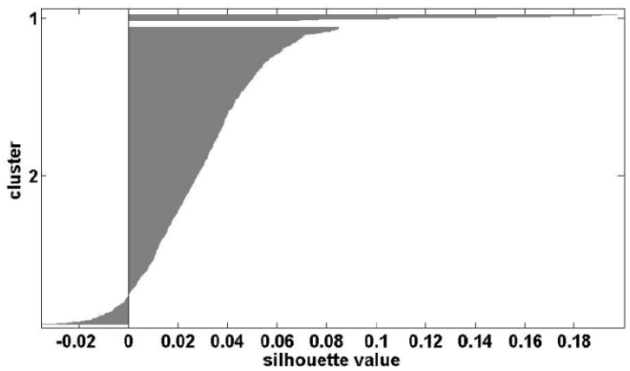

Fig. 10. The silhouette plot for two clusters

We have checked that there are no well separated sets of clusters below *nc* = 500. However, the substantial amount of sequences is clustered reasonably in several sets of clusters. An example is the set of 46 clusters which comes from a local minimum of $|ssn/ssp|$. About 400 sequences are assigned to clusters of the same haplotype name letter. Here are some examples. Clusters # 1, 2, 16, 33 consist exclusively of haplogroup A sequences. Cluster #23 consists of 51 haplogroup T sequences. Cluster #28 consists of 98 haplogroup H, HV and V sequences.

IV. CONCLUSIONS

It was demonstrated that a symbolic data sequence can be represented by a numerical vector of binary components in many dimensional linear spaces. The dimension of the space equals the number of words in a union of spectra resulting from parsing the sequences of the set. The similarity and distance measures between any two sequences were defined. Four examples of clustering long real character sequences were considered. With one exception a good clustering was achieved. The ability of numerical representation to enhance the clustering power of the standard k-means algorithm was demonstrated. The example of clustering quality with the use of indirect and direct numerical representation of 600 sequences from six novels is presented in Table 8. The use of the former representation leads to 80% of correct assignments, while the present paper representation leads to 98% of correct assignments.

Tab. 8. Quality of clustering with the use of indirect and present paper numerical representation

	indirect			direct		
cluster	size	correct	false	size	correct	false
A1	80	80	0	102	99	3
A2	92	72	20	98	97	1
B	104	76	28	101	100	1
C	100	76	24	103	100	3
D	103	79	24	101	100	1
J	121	100	21	95	95	0
total	600	483	117	600	591	9

The attempt to cluster above 1000 human mitochondrial DNA sequences failed to find a reasonable number of well separated clusters.

It follows from our examples that sets of a different number of clusters are often equally meaningful. In these cases, the experts in the application area have to interpret the clustering results and decide how detailed the clustering process should be and which set of cluster is more reasonable.

More details concerning the used data and results obtained are available upon request.

References

[1] C. Notredame, *Recent progress in multiple sequence alignment: a survey*, Pharmacogenomics **3**(1), 131-144 (2002).

[2] M. Randic, M. Vrako, *On the similarity of DNA primary sequences*, Journal of Chemical Information and Computer Sciences **40**, 599-606 (2000).

[3] S. Vinga and J. Almeida, *Alignment-free sequence comparison – a review*, Bioinformatics, **19**(4), 513-523 (2003).

[4] A. Kelil, S. Wang, Q. Jiang, R. Brzezinski, *A general measure of similarity for categorical sequences*, Knowl. Inf. Syst. **24**,197-220 (2010), DOI 10.1007/s10115-009-0237-8.

[5] B. Kozarzewski, *A method for nucleotide sequences analysis*, CMST **18**(1), 5-10 (2012).

[6] T. Kanungo, N.S. Netanyahu, A.Y. Wu, *An Efficient k-Means Clustering Algorithm: Analysis and Implementation*, IEEE Trans. Pattern Analysis and Machine Intelligence **24** (7), 881-892 (2002).

[7] B. Kozarzewski, *A New Method for Symbolic Sequences Analysis*, CMST **20**(3), 93-100 (2014), DOI:10.12921/cmst.2014.20.03.93-100.

Bohdan Kozarzewski received PhD degree in physics (1965) at the Jagiellonian University in Cracow. Habilitated in 1975 in solid state theory. Since 1989 Professor at the Institute of Physics Technical University Cracow, sine 2006 at the University of Information Technology and Management, Rzeszow. Present research activity in computer modeling of nonlinear dynamical systems and time series analysis. Author and co-author of about 50 scientific publications.

DOI:10.12921/cmst.2015.21.04.008