An investigation of K-means clustering to high and multi-dimensional biological data

Barileé B. Baridam
Department of Computer Science, University of Pretoria, South Africa
Email: bbaridam@cs.up.ac.za

M. M. Ali
School of Computational and Applied Mathematics,
Witwatersrand University, Wits 2050, Johannesburg, South Africa
Email: montaz.ali@wits.ac.za

Abstract

The K-means clustering algorithm has been intensely researched owing to its simplicity of implementation and usefulness in the clustering task. However, there have also been criticisms on its performance, in particular, for demanding the value of K before the actual clustering task. It is evident from previous researches that providing the number of clusters *a priori* does not in any way assist in the production of good quality clusters. Our investigations in this paper also confirm this finding.

The objective of this paper is to investigate further, the usefulness of the K-means clustering in the clustering of high and multi-dimensional data by applying it to biological sequence data. The squared Euclidean distance and the cosine measure are used as the similarity measures. We use the silhouette validity index first to show that K-means algorithm is not suitable for clustering high and multi-dimensional biological data irrespective of the distance or similarity measure employed. A preprocessor scheme is then added to the K-means algorithm. The scheme is used to automatically initialize a suitable value of K prior to the execution of the K-mean algorithm. Central to the preprocessor is the average silhouette value of the clusters. Our investigation suggests that the use of the silhouette value in the preprocessor improves the quality of clusters significantly for the biological datasets considered.

Furthermore, we suggest a scheme which maps the high dimensional data into low dimensions. We have then shown that the K-means algorithm with preprocessor produces good quality, compact and well-separated clusters of the biological data mapped in low dimensions. For the purpose of clustering we conduct a character-to-numeric conversions to transform the nucleic/amino acids symbols to numeric values.

Keywords: Clustering, Dimensionality, Categorical data, Silhouette validity index.

1 Introduction

Clustering is a statistical concept that has to do with the problem of identifying interesting distribution patterns and similarities between objects in a data set [1, 2]. It is an optimization problem that seeks to classify objects based on their proximity to one another. In this sense, objects that are most similar are grouped together forming groups of similar objects referred to as clusters. Clustering tasks involve generating clusters that are compact and well-separated from one another. It follows then that clustering task has to do with minimizing the intra-cluster distance or the within-cluster dispersion and maximizing the inter-cluster distance or the between-cluster dispersion.

There are two broad categories of clustering algorithms, namely hierarchical and partition-based clustering. K-means [4] is a well known partition-based clustering technique. It has been widely used since it was first introduced in 1967. It, as a general rule, demands the value of K, the number of clusters expected, to be provided before the actual clustering. This is common to partition-based clustering algorithms [5]. Besides the provision of the value of K a priori, it is actually expected that the clusters centers are also to be identified, and then the algorithm performs the partitioning tasks iterative until a solution is achieved. On the contrary, hierarchical clustering algorithms group objects into clusters without any knowledge of how many clusters there should be in the clustering task. This paper deals with the partition-based K-means clustering.

The task of determining K *a priori* actually results into the problem of determining which cluster each object belongs. Clearly, the initial K has impact on the performance of the algorithm. A wrong choice of K results in the algorithm converging to a local minimum instead of an expected global minimum solution. Running the algorithm several times with different initializations tend to overcome this problem. However, this process results in high computational time. A number of algorithms have been suggested to determine a suitable value of initial K, see for example ISODATA [3], SYNERACT [7], DYNOC [8] and MLBG [9]. However, all these algorithms contain sensitive parameters, and this means that trying to solve one problem creates another of similar nature.

Under the above circumstances, we incorporate a preprocessor prior to the execution of K-means. The silhouette validity index [20] plays an important role in determining the initial K in the preprocessor. We have shown that this optimizes K-means' performance in clustering the high dimensional data sets.

The remaining part of this paper is divided into sections as follows: Section 2 focuses on the preliminaries and related literature; Section 3 describes the new approach suggested for high dimensional data. Section 4 briefly presents the silhouette validity index. Section 5 shows the experimental results and Section 6 presents the concluding remarks.

2 The clustering task: data sets, similarity measures and algorithms

2.1 The data set

Associated with a given data set

$$\mathcal{S} = \left\{x^1, x^2, \cdots, x^N\right\}, \tag{1}$$

that needs clustering, are the attributes $(x_1^i, x_2^i, \cdots, x_D^i)$ of $x^i \in \mathcal{S}$ and N is the maximum number of items in the data set. For a numerical data set $\mathcal{S}$, each j-th attribute x_j^i in x^i is real and hence $x^i \in \mathbb{R}^D$. However, for a mixed data set features of x^i are generally two: numerical and categorical. Therefore, the attributes of x^i can be written as $(x_1^i, x_2^i, \cdots, x_p^i, y_1^i, y_2^i, \cdots, y_q^i)$, $p + q = D$, where $y_1^i, y_2^i, \cdots, y_q^i$ are categorical values.

Biological data being considered in this paper are that of nucleic acids - Deoxyribonucleic acid (DNA) and ribonucleic acid (RNA). Thus the categorical biological data set $\mathcal{S}$

$$x^i = (AAAAUUUUGGGCCAAAGGCCCUUUAAGCCCGG) \quad \text{for RNA} \tag{2}$$

and

$$x^i = (AAAATTTTGGGCCAAAGGCCCTTTAAGCCCGG) \quad \text{for DNA.} \tag{3}$$

DNA is a double helix structure with two strands of re-occurring nucleotides held together by base pairing. The top of the helix consists of a Guanine-Cytosine (GC) pair, referred to as purines, while the bottom consists of an Adeline-Thymine (AT) pair, referred to as pyrimidines. The GC base pair forms three hydrogen bonds, whereas the AT base pair forms two hydrogen bonds. DNA is a polymer with the nucleotides forming the monomer units. In its double stranded form, DNA is the genetic material of most organisms. The two strands form a double helix with the strands running in opposite directions as determined by the sugar-phosphate backbone of the molecule. DNA is represented in chains of symbols - AGCT (Adenine, Guanine, Cytosine, Thymine). For the purpose of this research they are represented as in (3). RNA is a bi-molecule made up of a chain of nucleotides as DNA, except that RNA introduces uracil (U) in place of Thymine (T). RNA and DNA are functionally and structurally different. A RNA strand folds onto itself. The folds form hydrogen bonds between G and C, A and U, and G and U, and their respective mirror images. The hydrogen bonds bind the base pairs to form DNA. There are enough literature for those interested in studying more of the structure of nucleic acids [12, 13, 14, 15].

The clustering process involved in this paper uses the above type of data. However, for the clustering purpose the biological sequence data have been converted into numerical data. Nucleic acids are represented in 3 dimension. During the sequencing process (that is, the process of converting nucleic acids into readable sequences) the 3 dimensional structure is rendered in a chain of nucleotides. The sequencing process renders the sequences in high- and multi-dimensions.

2.2 The similarity measure

An important component of a clustering algorithm is the distance measure between data points, say x^i and x^j. For continuous numerical data sets the squared Euclidean distance

$$d_{ij} = d(x^i, x^j) = \sum_{k=1}^{D} (x^i_k - x^j_k)^2 \tag{4}$$

is often used. The other well known similarity measure is the cosine similarity measure:

$$d_{ij} = d(x^i, x^j) = \frac{x^i \cdot x^j}{\|x^i\| \|x^j\|}, \tag{5}$$

where $\|x^i\|$ is the length of the vector x^i, and $x^i \cdot x^j$ is the dot product between vectors x^i and x^j. We have implemented both measures for comparison purposes.

2.3 The K-means algorithm

The minimization problem involved in the K-means algorithm for numerical data set can be formally written as follows [10]:

$$\min \sum_{m=1}^{K} \sum_{i=1}^{N} r_{im}\, d(x^i, C^m),\ r_{im} \in \{0, 1\}, \quad \text{subject to}$$

$$\sum_{m=1}^{K} r_{im} = 1,\ \forall i, \quad \text{and} \quad \sum_{i=1}^{N} r_{im} > 0,\ \forall m,$$

where C^m is the centroid of the m-th cluster and $d(x^i, C^m)$ is defined by equation (4) or (5). If x^i is assigned to cluster m then $r_{im} = 1$. The clustering process partitions a data set into K clusters S^i ($i = 1, 2, \cdots, K$) such that

(i) $S^i \neq \emptyset, i = 1, \cdots, K$;
(ii) $\bigcup_{i=1}^{K} S^i = \mathcal{S}$
(iii) $S^i \cap S^j = \emptyset, \forall\, i, j = 1 \cdots, K$ and $i \neq j$.

The basic steps of the K-means algorithm for numerical data set are as follows.

Algorithm 1: K-means clustering

```
Step 1. Assign $K$ initial centroids $C^1, C^2, \cdots, C^K$, one for each cluster $S^m$.
Step 2. For each data element $x^i \in \mathcal{S}$ find the nearest $C^m$ according to some similarity
        measure, e.g. the measures (4) or (5), and assign $x^i$ to the cluster $S^m$.
Step 3. For each cluster $S^m$ calculate a new centroid $C^m$.
Step 4. If some stopping condition $\lambda$ is reached stop Algorithm 1 else goto Step 2 with the
        new centroids $C^1, C^2, \cdots, C^K$.
```

2.4 The modified K-means algorithms

A number of modified K-means algorithms have been proposed in the literature. The purpose of these modified versions is to handle the problem related to initial K value.

Turi [6] proposed a K-means algorithm by dynamically changing the value of K as the iterations progress. Central to this algorithm are the merging and splitting of clusters. However, the algorithm requires the user to specify the values of several parameters (e.g. the merging and splitting thresholds). These parameters have a profound effect on the performance of making the result subjective.

Huang [7] proposed a K-means algorithms, referred to as SYNERACT. SYNERACT combines K-means algorithm with hierarchical divisive approaches to overcome K-means' setbacks. SYNERACT employs a hyper-plane to split a cluster into two smaller clusters and then compute their centroids, performs an iterative clustering to assign objects into clusters, and constructs a binary tree to store clusters generated from the splitting process. This method does not demand the initial provision of K and the initial location of centroids before the clustering task. However, the user is expected to specify the values of two parameters needed for the splitting process.

The dynamic optimal cluster-seek (DYNOC) algorithm was introduced by Tou [8]. DYNOC is a dynamic clustering algorithm. It achieves a maximization of the ratio of the minimum inter-cluster distance to the maximum intra-cluster distance through an iterative procedure with the capability of splitting and merging clusters. There are however user-specified parameters that suggest whether splitting or merging are necessary.

Rosenberger and Chehdi [9] attempted an improvement on K-means by introducing an iterative procedure known as the modified Linde-Buzo-Gray (MLBG) algorithm. MLBG automatically finds the number of clusters in a data set by using intermediate results. A cluster maximizing an intra-cluster distance measure is chosen for splitting iteratively. In the process, two cluster centroids are generated from the initial cluster. The first cluster centroid, C^1, is initialized to the centroid of the original (initial) cluster. The second cluster centroid, C^2, is chosen to be the object in the original cluster which is the most distant from C^1. At this point, K-means is on the new $K + 1$ centroids. The acceptance of the new set of centroids depends on if an evaluation criterion based on a dispersion measure is satisfied. This process is repeated until there are no valid partitions in the data set. The main problem with this method is that it requires the specification of the values of four parameters which have a fundamental effect on the resultant number of clusters [11].

The K-means algorithm is found to be the best applied to numeric data [16], and the modifications, discussed above, dealing with numerical data set are very encouraging. However, the application of the K-means algorithm to mixed data set are extremely limited. An attempt is made by Gupta et. al. [17] to apply the K-means algorithm by adopting two different similarity measures. An integrated cost function is suggested which has two components. A cost owing to numeric attributes is minimized by usual way i.e. assigning elements to clusters, while the other cost, owing to categorical attributes, is minimized by selecting the categorical elements of centroid. However, the method of Gupta et. al. [17] has neither been justified by mathematical means nor has it been validated by sufficient numerical testing. In addition, the method is not parameter-free. Finally, although Andreopoulos, *et al.* introduced a bi-level clustering of mixed categorical and numerical biomedical (gene expression) data [24], the clustering of categorical biological data set (nucleic or amino acids) is not so much addressed in the literature. This paper is concerned with the investigation of the performance of K-means in the clustering of high/multi-dimensional data of which biological sequence data is one. Biological data sets investigated in this paper contains high and multi-dimensional data sequence x^i, and to the best of our knowledge, there is no K-means algorithm developed for clustering of these data sets. Also, conventional clustering methods cannot be applied to the clustering of biological data owing to the structural nature of the data [25]. Hence, we have decided to study this clustering problem.

3 Application of K-means to biological sequence data

In this paper, we investigate the ability of K-means in the clustering of high and multi-dimensional data sets - a situation where the input data are of several dimensions. In addition, the biological sequence data sets we consider are naturally not numeric. The original objective of the K-means algorithm [4] and the subsequent findings [16] suggest that the numerical presentations of the categorical biological data set is needed for successful applications of K-means. For this, we use conversions from symbols to numeric by representing each sequence in the data set in a D-dimensional space through the application of a comma delimited conversion format. In particular, the nucleic acid symbols are represented numerically as follows: $A = 1, C = 2, G = 3$ and U or $T = 4$. For the clustering of a biological sequence data set, we adopt the following two separate approaches.

- Firstly, the sequences in the data set were truncated to a uniform dimension before the clustering, leaving the sequences in their high-dimensional state.
- Secondly, the dimension of each of the sequences was reduced to a uniform low dimension (D_r) before clustering.

We define the dimension reduction by introducing the following concepts and definitions. Let N represent the number of nucleotides in a sequence; l, the sequence length; n_i, the i-th individual nucleotides (symbols) in a sequence already represented in numeric format. It is important to note that it is conventional to state that $N = l$ in cases where the delimiters are not counted to constitute part of the length e.g. as presented by equations (2) and (3). We calculate the coordinates of a sequence $x^i \in \mathcal{S}$ as follows:

$$Q_i = \left(\sum_{i=1}^{\bar{d}_1} n_i, \sum_{i=\bar{d}_1+1}^{\bar{d}_2} n_i, \cdots, \sum_{i=\bar{d}_j+1}^{D} n_i \right), \tag{6}$$

where $\bar{d}_p = \sum_{i=1}^{p} d_i$ with $\bar{d}_1 = d_1, p = 1, 2, \cdots, j$. We use $d_p = d_q, p \neq q$ for all $p, q = 1, 2, \cdots, j$, whenever possible. When this is not possible an integer in $\{d_1, d_2, \cdots, d_j\}$ is selected at random and its value is adjusted so that

$$\left(\sum_{k=1}^{j} d_k\right) + \left(D - \bar{d}_j\right) = D.$$

where d_i is any converted nucleotide in x^i.

A two dimensional representation of Equation (6) is given by:

$$Q_i = \left(\sum_{i=1}^{d_1} n_i, \sum_{i=d_1+1}^{N} n_i\right). \qquad (7)$$

If the coordinates of Q_i become large then they can be represented in ratios of least common multiples, but this was not required for the data sets we considered for numerical testing. We have implemented both the above procedures for the clustering of the nucleic acid sequences.

4 The silhouette index

The silhouette validity index for each data element is simply a measure of how similar that data element is to elements in its own cluster compared to elements in other clusters [18, 19]. It ranges from -1 to +1. The silhouette validation index is particularly useful when seeking to know the number of clusters that will produce compact and clearly separated clusters [21, 22, 20]. The silhouette index [20, 23] of the element x^i of a cluster S^j is defined as

$$q_i = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad -1 \leq q_i \leq 1, \qquad (8)$$

where $a(i)$ is the average similarity between x^i and the rest of the objects in cluster S^j and $b(i)$ is the minimum average similarity between object x^i and the rest of the objects in all the clusters, defined as

$$\min_{S^m \neq S^j} d(x^i, S^m) \;\; (m = 1, 2, \cdots, K; m \neq j).$$

Every object x^i with a silhouette index close to 1 indicates it belongs to the cluster being assigned. A value of zero indicates that the object could also be assigned to another closest cluster. A value close to -1 indicates that the object is in a wrong cluster or somewhere in between the clusters. The highest value indicates the best clustering, meaning that the number of clusters selected for the clustering is the best [20].

5 Experimental results and performance analysis

5.1 Application to high dimensional data

We begin with the application of the K-means algorithm on high dimensional data sets. Six datasets were used, namely *emblFasta Rickettsia typhi str.* RNA sequences with Accession Number AE017197 from Wilmington Complete Genome of 1111500 nucleotides, Homo sapiens' *melanatonic melanoma* DNA sequences, mRNA *bos taurus* sequences from Genetic Sequence Databank with Accession Number BE484664 obtained from the work of Sonstegard, *et al* [26], and DNA dental sequences from Department of Micro-biology, University of Pretoria, South Africa. Each data set contains data elements (sequences) of equal length, due to the truncation mentioned earlier. The K-means algorithm was applied more than once on a data set to see the effect of K in the clustering process. Results of this investigation is presented in Table 1. In Table 1, the following symbols are used: i (data set), N (size of data set), D (dimension), K (number of clusters), I_E (number of iterations required when using squared Euclidean distance), I_C (number of iterations required when using cosine similarity measure), Td_E (distance using squared Euclidean), Td_C (distance using cosine measure), Sh_E (silhouette mean under Euclidean distance) and Sh_C (silhouette mean under cosine measure). The data in columns under 'Total distance' are the total intra cluster distance from the centroid of formed clusters[1]. That is if there are three clusters and d_{ij} is the distance the i-th element (of the j-th cluster S^j with n_j elements) and its centroid C^j, then the total is calculated over the three centroids of clusters of the data set, generated during the iteration process. This means that the total sum of distance is the value realized at the last iteration when the algorithm reaches a minimum, and the total is calculated over the set

$$\left\{\sum_{i=1}^{n_1} d_{1i}, \sum_{i=1}^{n_2} d_{2i}, \sum_{i=1}^{n_3} d_{3i}\right\}.$$

The data in columns under 'Silhouette mean' are the average of the silhouette values. For example, the average silhouette index values for the m-th cluster is given by

$$Q(m) = \frac{1}{n(m)} \sum_{i=1}^{n(m)} q_i(m),$$

where $q_i(m)$ is the silhouette value for the i-th member of the m-th cluster, and $n(m)$ is the total number of elements in the m-th cluster. The values presented in the last two columns in Table 1 are therefore the values $\frac{1}{K}\sum_{m=1}^{K} Q(m)$, where K is the total number of clusters.

Table 1: Effects of K in the K-means algorithm applied to the high dimensional data

				Iterations		Total distance		Silhouette mean	
i	N	D	K	I_E	I_C	Td_E	Td_C	Sh_E	Sh_C
1	117	128	5	7	11	16264.3	8.71803	0.0335	0.0338
	117	128	10	11	9	14791.7	7.99373	0.0533	0.0396
2	117	198	5	20	10	25844.5	9.03496	0.0180	0.0288
	117	198	10	11	7	23776.3	8.32794	0.0309	0.0318
3	100	50	4	12	10	4654.08	6.19611	0.0622	0.0561
	100	50	6	7	11	4968.8	7.8109	0.0663	0.0656
4	50	50	5	8	5	2366.79	3.24409	0.0916	0.0742
	50	50	4	7	11	2512.22	3.3818	0.0707	0.0720
5	50	20	5	6	5	865.011	2.93337	0.1260	0.1270
	50	20	4	4	11	937.727	3.13017	0.0995	0.1281
6	20	50	4	4	3	829.083	1.16313	0.1193	0.1134
	20	50	3	3	4	919.786	1.30774	0.0962	0.1077

To see the effect of K we study the the 4th and 5th major columns (Total sum of distances and Silhouette mean) in Table 1. The total sum of distances should be as low as possible–a better clustering should give a lower value of the total sum of distances. However, these values seem quite high. We therefore study the silhouette means in Table 1.

[1] The total sum of distances decreases at each iteration as K means reassigns points between clusters and recomputes cluster centroids.

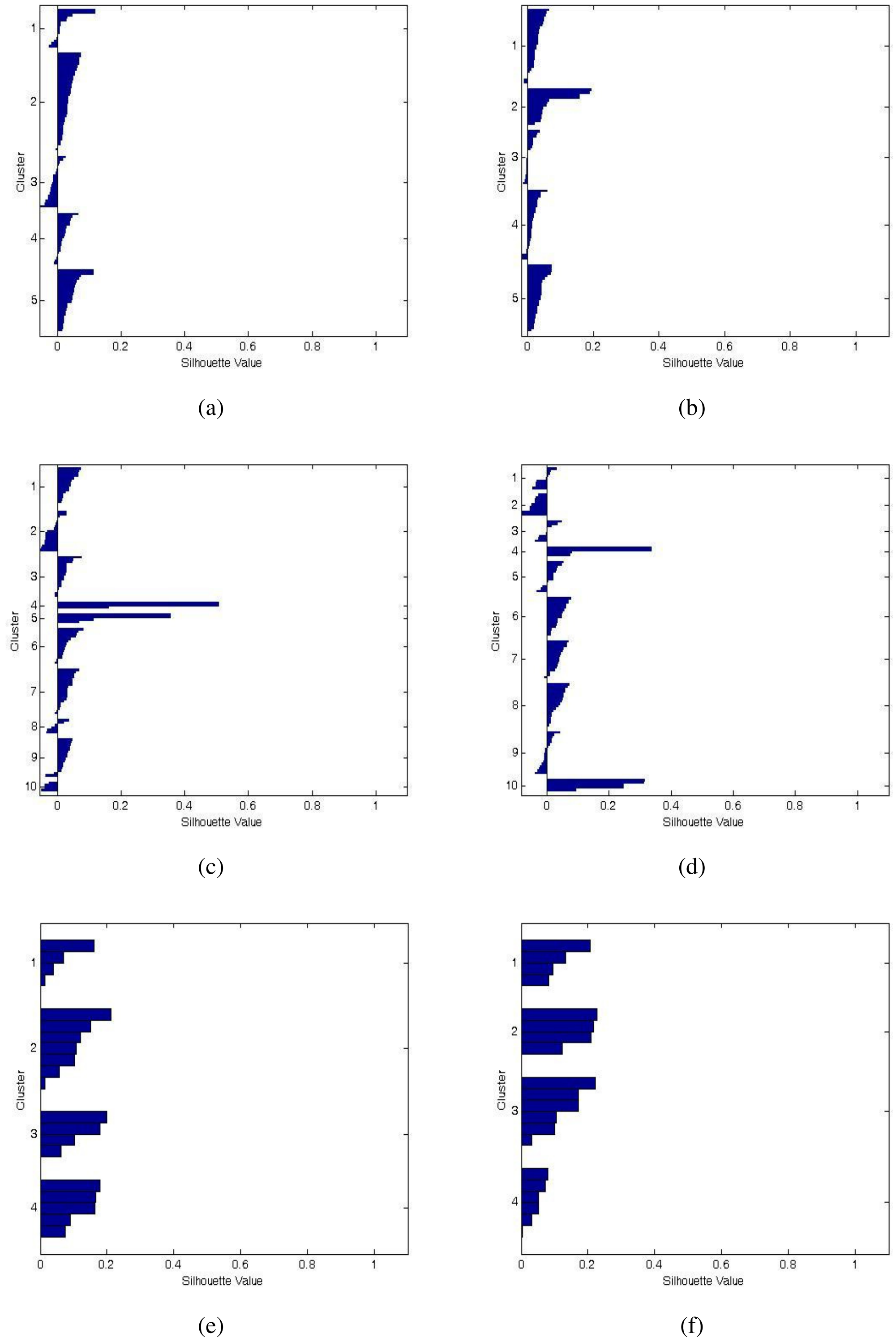


Figure 1: The Silhouette plots for clusters of high dimensional data with various values of K, with the squared Euclidean distance measure [(Figs. 1(a), (c) - (128D) and (f) - (50D)) and the Cosine similarity measure (Figs. 1(b) and (d) - (128D) and (e) - (50D))].

These values determine how far apart the clusters are. With a high value, the cluster quality is near optimal. Again these values seems non-optimal. To visualize the cluster-wise silhouette index values, we present in Figure 1 the silhouette plot of generated clusters. For this we have used data sets 1 and 3 in Table 1. Figures 1(a) and 1(c) are, respectively, for K=5 and 10 using squared Euclidean measure, data set 1. Figures 1(b) and 1(d) are, respectively, for K=5 and 10 using cosine measure, data set 1. Figures 1(e) and 1(f) are for the data set 3 using cosine and Euclidean measures respectively, K=4. Figure 1 clearly shows that the many silhouette values are negative and the overall results are unsatisfactory. Observe from Table 1 and Figure 1 that as the dimension decreases, the result of the clustering becomes better. It is clear from the figures that the value of K greatly determines the cluster quality.

5.2 The preprocessor of the K-means algorithm

To deal with the initialization problem of K, we suggest an automatic initialization scheme. The silhouette mean under Euclidean distance measure, Sh_E, presented in Table 1 plays the most important role in the scheme. For an initial value of K, provided by the user, the Algorithm 1 (the K-means algorithm) is run for a small number of iterations (e.g. typically 3) three times, respectively using $K-2$, K and $K+2$. Three corresponding Sh_E values corresponding to $K-2$, K and $K+2$ are found (hereafter denoted as $Sh_E(K-2)$, $Sh_E(K)$ and $Sh_E(K+2)$, respectively). The initial value, K_o, of K is then assigned using the following procedure:

1. If $Sh_E(K-2) < Sh_E(K)$ and $Sh_E(K) > Sh_E(K+2)$ then the Algorithm 1 is run again twice (each time for 3 iterations) using $K+1$ and $K-1$ and the corresponding $Sh_E(K-1)$ and $Sh_E(K+1)$ are found. The maximum value of three $\{Sh_E(K-1), Sh_E(K)$ and $Sh_E(K+1)\}$ then determines K_o. For example if $Sh_E(K-1)$ is the maximum then we assign $K_o = K-1$.

2. If $Sh_E(K+2) > Sh_E(K)$ and $Sh_E(K+2) > Sh_E(K-2)$ then the Algorithm 1 is run again using $K+1$, $K+3$ and $K+4$. The K value corresponding to the maximum in $\{Sh_E(K+1), Sh_E(K+2), Sh_E(K+3), Sh_E(K+4)\}$ is then assigned to K_o.

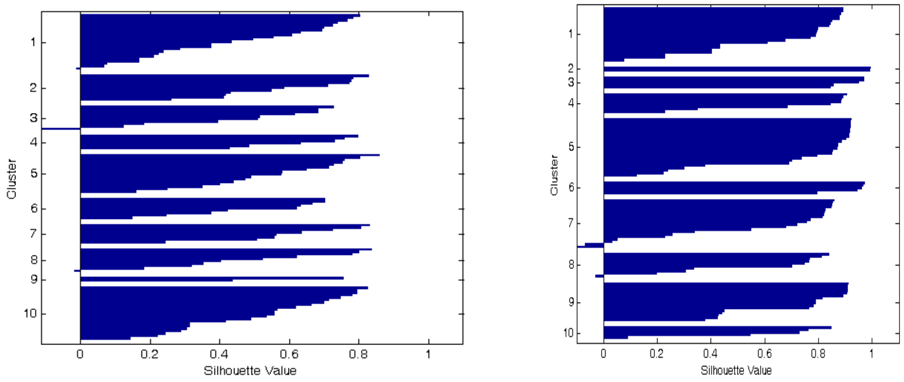
3. If $Sh_E(K+2) < Sh_E(K-2)$ and $Sh_E(K) < Sh_E(K-2)$ then the value corresponding to the maximum in $\{Sh_E(K-1), Sh_E(K-2), Sh_E(K-3), Sh_E(K-4)\}$ is then assigned to K_o.

The initial value[2], K_o, of K found using the above procedure is then used to find K_o clusters using K-means algorithm, i.e. the Algorithm 1. To test the effectiveness of the above procedure we use two data sets from Table 1, namely the first and the third data sets. We have used the initial K in the preprocessor as given in Table 1. Results obtained are presented in Table 2. Table 2 clearly shows that the results have been improved for both

Table 2: Effects of preprocessor in the K-means algorithm

					Iterations		Total distance		Silhouette mean	
i	N	D	K	K_o	I_E	I_C	Td_E	Td_C	Sh_E	Sh_C
1	117	128	5	8	10	7	14576.1	7.8258	0.0437	0.0415
3	100	50	4	5	6	5	3052.75	3.05384	0.1163	0.1143

[2] The above process consisting of steps 1-3 can be repeated anew (with a new K) if the Sh_E increases monotonically. However, this was not needed for our implementation.



(a) (b)

Figure 2: The silhouette plots for clusters derived for reduced dimension e.g for D_r=2 and without preprocessor: $K = 10$, N= 117, [(a) Euclidean], [(b) Cosine].

data sets, although the problem dimension is very high.

5.3 Application to reduced dimensional data without preprocessor

A further test was done on K-means algorithm without the preprocessor scheme using the data sets presented in Table 1, but with reduced dimensions (D_r). We first test data sets of dimension two obtained by equation (7). We use the data sets presented in Table 1 and present the results obtained in Table 3. To see the effect of reduced dimensionality we do not incorporate the preprocessor in this experiment. We also use the same K values as in Table 1 as this will allow us to compare Tables 1 and 3 directly. The results in Table 3 show significant improvement in all data sets with high silhouette means than those in Table 1. We present two figures, both for the data set 1 (K=10), corresponding to two different measures. Figures 2(a) and (b) correspond to the corresponding Figures 1(a) and (c). This comparison also establishes positive effect of dimension reduction.

Table 3: Effects of reducing D in the K-means clustering, D_r=2

				Iterations		Total distance		Silhouette mean	
i	N	Previous D	K	I_E	I_C	Td_E	Td_C	Sh_E	Sh_C
1	117	128	5	8	12	7021.1	0.0110784	0.5377	0.7014
	117	128	10	16	6	3240.82	0.00461887	0.5184	0.6808
2	117	198	5	12	9	10019.5	0.00600523	0.5879	0.7000
	117	198	10	9	12	5678.06	0.00242334	0.5118	0.6507
3	100	50	4	15	5	2461.83	0.0221821	0.5103	0.7254
	100	50	6	9	13	1697.71	0.0106729	0.4941	0.7253
4	50	50	4	9	4	1018.37	0.0074717	0.5014	0.7700
	50	50	5	6	4	815.016	0.00648176	0.5038	0.7492
5	50	20	5	4	4	291.551	0.0144597	0.4524	0.6780
	50	20	4	13	5	350.375	0.0275186	0.4736	0.6206
6	20	50	4	3	3	184.571	0.00327066	0.6619	0.5042
	20	50	3	2	6	329.19	0.00360131	0.5612	0.7446

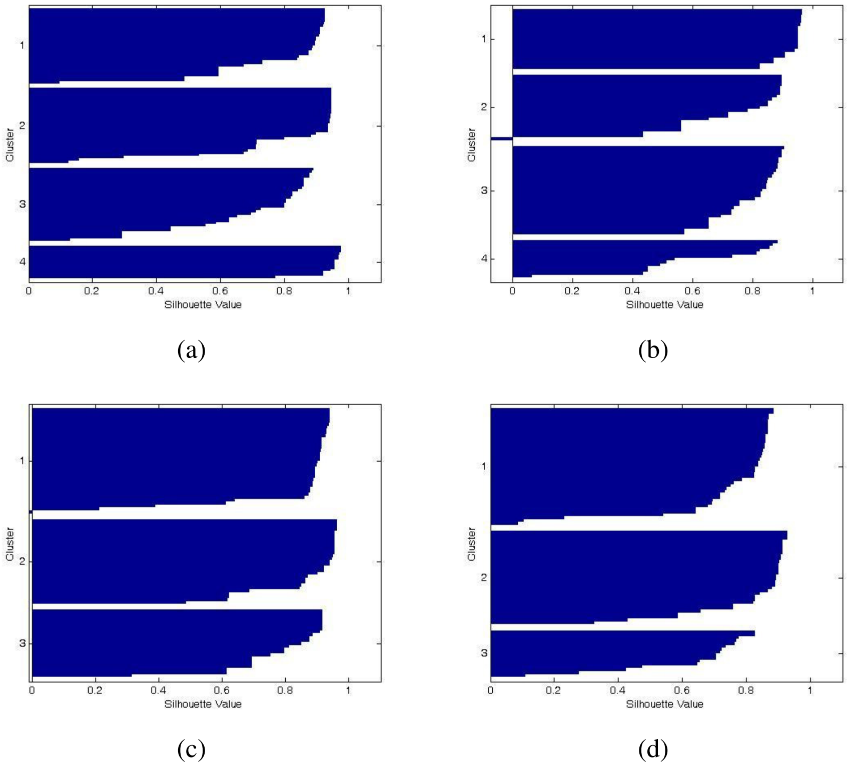


Figure 3: The silhouette plots for clusters of 50 data points (data set 4)[(a) K_o= 4 (Cosine measure) and (b) K_o=4 (squared Euclidean measure)] and 88 data points (data set 10)[(c) K_o=3 (cosine measure), and (d) K_o=3 (squared Euclidean measure)]

Table 4: Optimal K_o in the K-means algorithm

				Iterations		Total distance		Silhouette mean	
i	N	K_o	K	I_E	I_C	Td_E	Td_C	Sh_E	Sh_C
1	117	7	5	6	7	3940.84	0.146286	0.6587	**0.8110**
2	117	6	5	5	5	5767.98	0.608631	0.6837	0.7358
3	100	5	6	6	7	6142.24	0.453146	0.6687	0.7923
4	50	4	5	9	6	7338.64	0.625917	**0.7508**	0.7859
5	50	6	4	10	7	12943.3	0.967989	0.7246	0.8021
6	20	7	3	13	7	3322.21	0.137556	**0.7537**	0.7898
7	88	6	7	5	9	4017.84	0.148084	0.6869	0.8072
8	88	5	5	5	4	4210.39	0.319518	0.7433	**0.8143**
9	88	4	3	6	5	8646.92	0.46525	0.6207	0.7772
10	88	3	6	6	3	9514.41	0.64087	**0.7546**	**0.8348**

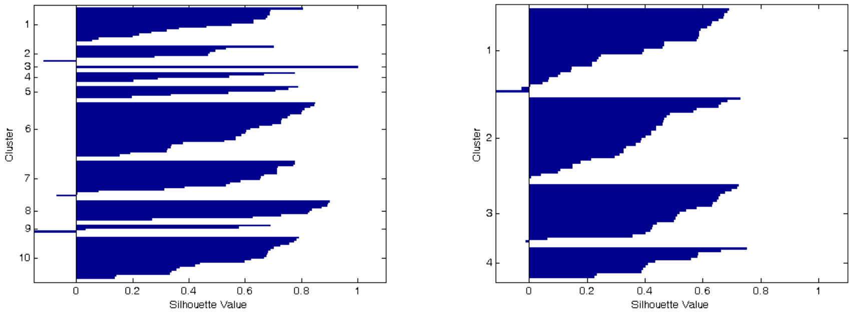
5.4 Application to reduced dimensional data with preprocessor

We now study the effect of both reduced dimension and preprocessor on 10 data sets. We first consider D_r=2 and present the results in Table 4, where the first 6 data sets are the same data sets considered before.

To see the effect of the dimension reduction we now compare the same data set in Tables 2 and 4, i.e. the data set 1 in Tables 2 and 4. Results show that K_o corresponding to this data set in both tables are very close. This proves the effect of preprocessor as well as the dimension reduction in K-means for categorical biological data sets. Notice that for data set 8, K and K_o are the same. This means that the initial K assigned to preprocessor remained the same.

We further present the silhouette values in Figure 3 for two data sets of 50 and 88 data points, respectively, with reduced dimensions. These are respectively the 4th and 10th data sets presented in Table 4. These figures clearly shows well separated clusters. The usefulness of the silhouette value in the clustering task as well as the incorporation of the preprocessor are now evident.

An obvious question that one may rise is how to identify an appropriate value for the reduced dimension, D_r. To address this question, we reproduce the values of the data set 1 in Table 3 using D_r=3. Results obtained are very similar. For example, for K=10 we obtained the following values: Td_E=5448.01, Td_C=0.1872, Sh_E=0.4210, and Sh_C=0.6062. We present the corresponding graph for K=10 in Figure 4. In addition, we present a graph for the data set 3.



(a)

(b)

Figure 4: The silhouette plots for clusters of 117 data points (data set 1)[(a) K= 10 (Euclidean measure) and (b) K=4 (Euclidean measure)] and 100 data points (data set 3)

Although preprocessor has not been used for this experiment, the graphs produced show that silhouette values are fairly acceptable. These results can be further improved by the use of preprocessor. Our experiments have shown that the optimized values are not exactly the same, for D_r=2 and 3, but they are within an acceptable level of closeness. Hence, we suggest that D_r=2 is a good value to choose.

6 Conclusion and further research

We have studied the usefulness of the K-means algorithm for clustering the categorical biological sequence data. These sequences consist of alphabets and are of high and multi-dimensional in nature. We introduced a numerical equivalence sequence of the categorical data. To reduce the effect of initial K in K-means we have introduced a preprocessor scheme. We have shown that significant gains in optimality can be achieved by using the preprocessor. In addition, we introduced a dimension reduction technique which when applied with the preprocessor produces well separated clusters.

It is necessary to state here that the work presented in this paper is not about comparing the performance of algorithms. We have not also said that K-means is better than any other algorithm. Since K-means clustering algorithm have been widely researched, we have only investigated its performance in the clustering of high and multi-dimensional categorical data (in this case biological sequence data were used). Suffice us to say that the clustering technique introduced in the paper is new and thus can be applied to many similar practical problems.

References

[1] P. Berkhin, ‘Survey of clustering data mining techniques’, Technical report 4, Accrue Software, *Inc.*, San Jose, California, 2002.

[2] D. A. Binder, ‘Cluster analysis under parametric models’, PhD thesis, University of London, 1977.

[3] J. Tou and R. Gonzalez, ‘Pattern Recognition Principles’, Addison-Wesley, Massachusetts, USA, 1974.

[4] J. B. MacQueen, ‘Some methods for classification and analysis of multivariate observations’, Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, 281–297, 1967.

[5] F. D. Smet, J. Mathys, K. Marchal, G. Thijs B. D. Moor and Y. Moreau, ‘Adaptive quality-based clustering of gene expression profiles’, Bioinformatics, 18(6),:735–748, 2002.

[6] R. H. Turi, ‘Clustering-based Colour Image Segmentation’, PhD thesis, Monash University, 2001.

[7] K. Huang, ‘A Synergistic Automatic Clustering Technique (SYNERACT) for Multispectral Image Analysis’, Photogrammetric Engineering and Remote Sensing, 1(1):33–40, 2002

[8] J. Tou, ‘DYNOC - A Dynamic Optimal Cluster-seeking Technique’, International Journal of Computer and Information Sciences, 8(6):541–547, 1979.

[9] C. Rosenberger and K. Chehdi, ‘Unsupervised Clustering Method with Optimal Estimation of the Number of Clusters: Application to Image Segmentation’, Proceedings of the International Conference on Pattern Recognition (ICPR’00), pages 1656–1659, 2000.

[10] J. C. Bezdek, ‘A Convergence Theorem for the Fuzzy ISODATA Clustering Algorithms’, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2:1-8, 1980.

[11] M. G. H. Omran, ‘Particle Swarm Optimization Methods for Pattern Recognition and Image Processing’, PhD thesis, University of Pretoria, Faculty of Engineering, Built Environment and Information Technology, Department of Computer Science, 2004.

[12] P. E. Bourne and H. Weissig. ‘In Phillip Bourne and Helge Weissig’, editors, *Structural Bioinformatics*, pages 35–49. Wiley-Liss, Inc., Hoboken, New Jersey, 2003.

[13] National Human Genome Research Institute. *The structure of Ribonucleic and Deoxyribonucleic Acids*. National Institutes of Health, Division of Intramural Research. Available online: www.nhgri.gov.

[14] M. F. Ramoni, P. Sebastiani, and I. I. Kohane. ‘Cluster analysis of gene expression dynamics’. In *Proceedings of National Academy of Science*, volume 99, pages 9121–9126, July 2002.

[15] Y. Xu, V. Olman, and D. Xu. ‘Clustering gene expression data using a graph theoretic approach: an application of minimum spanning trees’. *Bioinformatics*, 18(4):536–545, 2002.

[16] R. Xu and D. Wunsch II, ‘Survey of Clustering Algorithms’, International Journal of Intelligent Computing and Cybernetics, 16(3):601–614, 2005.

[17] S. R. Gupta, K. S. Rao and V. Bhatnagar, ‘K-means clustering algorithm for categorical attributes’, Proceedings of 1st International Conference on Data Warehousing and Knowledge Discovery, pages 203–208, Florence, Italy, 1999.

[18] MATLAB, ‘The Language of Technical Computing’,version 7.0, The Mathworks, Inc., 2004.

[19] L. Kaufman and P. Rousseuw, ‘Finding Groups in Data: An Introduction to Cluster Analysis’, Wiley, 1990.

[20] P. Rousseuw, ‘Silhouettes:A practical aid to the interpretation and validation of cluster analysis’, Computational and applied mathematics, 20, 1987.

[21] F. Azuaje, ‘Cluster validity framework for genome expression data’, Bioinformatics, 18(2), 2002.

[22] M. D. González” Teledo, ‘A comparison in cluster validation techniques’, University of Puerto Rico, Department of Mathematics(Statistics), Master of science thesis, 2005.

[23] N. Bolshakova and F. Azuaje, ‘Cluster validation techniques for genome expression data’, Signal Processing, 83:825–833, 2003.

[24] B. Andreopoulos, A. An and X. Wang, ‘Bi-level clustering of mixed categorical and numerical biomedical data’, International Journal of Data Mining and Bioinformatics, 1(1):19–56, 2006.

[25] B. B. Baridam and O. Owolabi, ‘Conceptual Clustering of RNA Sequences with the Codon Usage Model’, Global Journal of Computer Science and Technology, 10(8):41–45, 2010.

[26] T. Sonstegard, A. V. Capuco, J. White, C. P. Van Tastell, E. E. Connor, J. Cho, R. Sultana, L. Shade, J. E. Wray, K. D. Wells and Quackenbush, J.”, ‘Analysis of *bovine* mammary gland EST and functional annotation of the *Bos Taurus* gene index’, Mammary Genome, 13(7):373–379, 2002.