CHAPTER 4

Results and Discussion

This chapter describes the experimental results of the five proposed methods, i.e., sgFCMed, sgFPCMed, sgPFCMed, sgUPCMed and sgUPFCMed on four standard real-world data sets, i.e., MPEG-7 data set, Copenhagen chromosomes data set, MNIST database of handwritten digits, and USPS database of handwritten digits in section 4.1. The evaluation the quality of partitions using three cluster validity indices, i.e., PE, PC, and XB are also described for each data set and the measuring degree of overlapping data is described.

4.1 Results and discussion on Standard Data Sets

We ran our algorithms on four real data sets, i.e., MPEG-7 data set, Copenhagen chromosomes data set, MNIST database of handwritten digits, and USPS database of handwritten digits. For all of experiments, we divided each experiment into two part, i.e., single prototype and multi-prototypes clustering. In the training process of all experiments, we used ten-fold cross validation. A test string was assigned to the class of the nearest prototype in the testing process for single-prototype and the nearest multi-prototypes was used to assigned any test string to the class. Then, we evaluated the quality of partitions using three cluster validity indices, i.e., PE, PC, and XB. In the single prototype experiments, the median string of each class is used as the initial cluster prototype and we randomly select the initial cluster prototypes for multi-prototypes, the parameters setting for all experiment is shown in Table 4.1.

In our work, the *R-value* is utilized to measuring overlapping of our datasets, i.e., MPEG-7 data set, Copenhagen chromosomes data set, MNIST database of handwritten digits, and USPS database of handwritten digits. The step of *R-value* calculation is to find

the *K* nearest neighbor string and then counts the number of string that belonging to the other classes. If the summation of the counted number (*R*) is larger than 0, that means the dataset is overlapping data. The *R-value* can evaluate the degree of overlap. The higher *R-value* indicates that the dataset contains large overlapping area among its class. In our experiments, we apply *R-value* based on the Levenshtein distance. The *K* value is set equal to 7 for all experiments. The *R-value* of 4 real world data sets are shown in Table 4.2

Table 4.1 Parameter setting of our algorithms for all experiments.

parameter	sgFCMed	sgFPCMed	sgPFCMed	sgUPCMed	sgUPFCMed
m	2	2	2	2	2
η	-	2	2	-	2
γ	-	-	2	-	-
a	-	-	1	-	1
b	-	-	1	-	4
stopping criteria	0.1	0. 1	0.1	0.1	0.1
maximum number of iterations	100	100	100	100	100

Table 4.2. The *R-value* of 4 real world data sets.

Dataset	Number of samples	Number of classes	*R-value*
MPEG-7	1,400	70	0.4814
Copenhagen chromosomes	2,200	22	0.2227
MNIST database of handwritten digits	60,000	10	0.1390
USPS database of handwritten digits	7,291	10	0.1920

4.1.1 MPEG-7 Core Experiment CE-Shape-1 Part-B Data Set

The MPEG-7 data set [45-49] is a well-known shape matching evaluation database. The core experiment was divided into three parts, i.e., part A: robustness to scaling and rotation, part B: performance of the similarity-based retrieval, and part C: robustness to changes caused by non-rigid motion. In our work, we focus only on part B of MPEG-7 Core Experiment CE-Shape-1 that consist of 70 different classes of shape, each class has 20 different shapes with high intra-class variability. The whole data set, Therefore, consists of 1400 binary images. Some examples of images in this data set are shown in Figure 4.1 (seven shapes from the same class are shown in each row).

To create a string from each image, there are 4 steps as follows:

1. We first cropped each image to only cover the object and then resized it to 35 pixels in height while the width was scaled according to the original aspect ratio.
2. We extracted the boundary or the contour of the image using the Moore-neighbor tracing algorithm [50].
3. The boundary image was encoded using the 8-directional chain code [51].
4. Finally, the differential chain code [51] was used as the sequence of the image string.

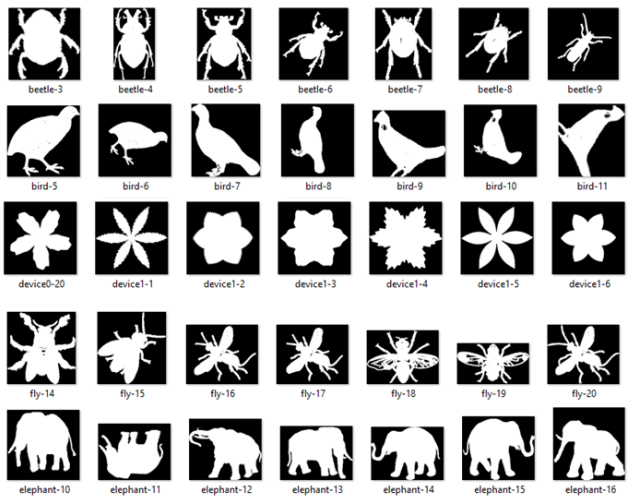
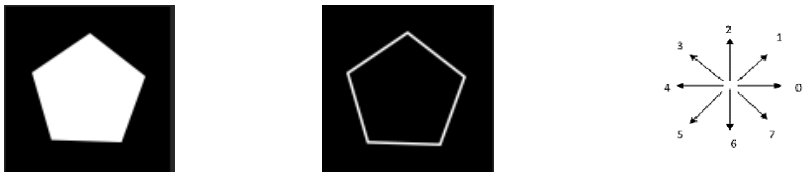


Figure 4.1 Examples from MPEG-7 shape data sets.

The process of generating a string from an image is shown in Figure 4.2. The chain code of the boundary in Figure 4.2 is 11117777555544443333 and the differential chain code of this image is 60006000600070007000.



Binary image Boundary image 8-direction chain code

Figure 4.2 The string generation of MPEG- shape data sets.

The classification accuracy rates of the ten-folds cross validation for single-prototype of the sgHCM, sgFCMed, sgFPCMed, sgPFCMed, sgUPCMed and sgUPFCMed are shown in Table 4.3. We can see that the best validation result from the sgUPFCMed m=2, η=4 is 90.71% correct classification, while that of the sgHCM, sgFCMed, sgFPCMed, sgPFCMed and sgUPCMed are 72.86%, 85.00%, 86.43%, 90.00%, and 90.00%, respectively.

Table 4.3 Single-prototype classification error rate on validation sets of MPEG-7 data set from sgHCM, sgFCMed, sgFPCMed, sgPFCMed. sgUPCMed and sgUPFCMed.

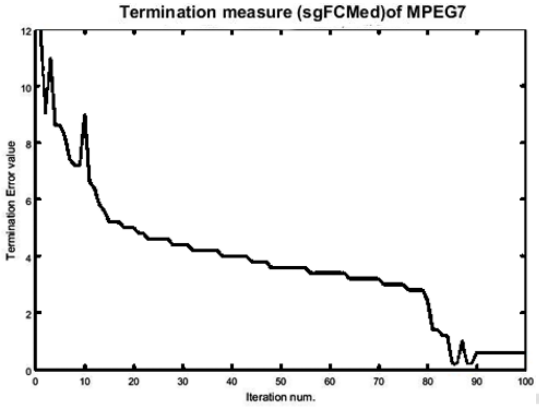
m	η	training data set (best validation set result)					
		sgHCM	sgFCMed	sgFPCMed	sgPFCMed	sgUPCMed	sgUPFCMed
1.5	1.5	27.14	23.33	18.57	16.43	12.33	13.57
1.5	2.0			15.00	12.85		10.00
1.5	3.0			15.00	13.57		12.14
1.5	4.0			15.00	15.00		12.85
2.0	2.0		15.00	15.00	12.14	10.00	10.71
2.0	3.0			13.57	10.00		10.71
2.0	4.0			13.57	12.14		**9.29**
3.0	2.0		18.33	15.71	15.00	12.14	10.00
3.0	3.0			16.43	13.57		10.71
3.0	4.0			15.00	12.86		10.71

Table 4.4 Multi-prototypes classification error rate on validation sets of MPEG-7 data set from sgHCM, sgFCMed, sgFPCMed, sgPFCMed, sgUPCMed and sgUPFCMed.

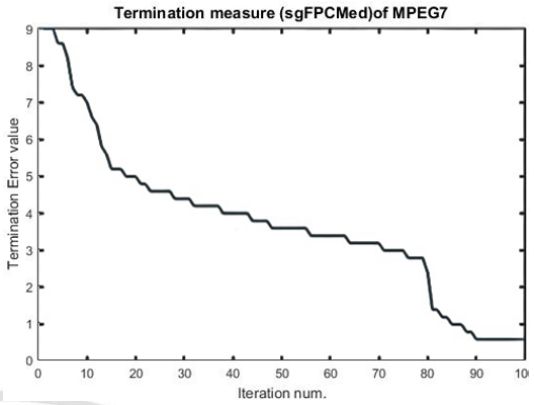
# prototype of each class	training data set (best validation set result)					
	sgHCM	sgFCMed	sgFPCMed	sgPFCMed	sgUPCMed	sgUPFCMed
2	25.00	15.00	13.57	9.29	8.57	7.86
3	24.29	15.00	12.86	8.57	7.86	8.57
4	24.29	14.29	12.86	7.86	7.14	7.14
5	18.86	11.57	8.14	5.86	5.43	3.71
6	17.86	10.71	7.86	4.29	5.00	**2.14**
7	18.57	10.71	8.57	5.71	5.71	2.86
8	20.00	11.43	8.57	6.43	6.43	2.86
9	21.43	12.86	10.00	6.43	7.14	3.57
10	22.14	13.57	11.43	6.43	7.14	4.29

In Table 4.4, we ran multi-prototypes string grammar fuzzy clustering algorithms (2-10 prototype of each class) with the best parameter from single-prototype, we can see that the best validation result on validation sets of the ten-folds cross validation from the 6 prototypes of sgUPFCMed is 97.86% correct classification, whereas that from the sgHCM, sgFCMed, sgFPCMed, sgPFCMed and sgUPCMed are 82.14%, 89.29%, 92.14%, 95.71% and 95%, respectively.

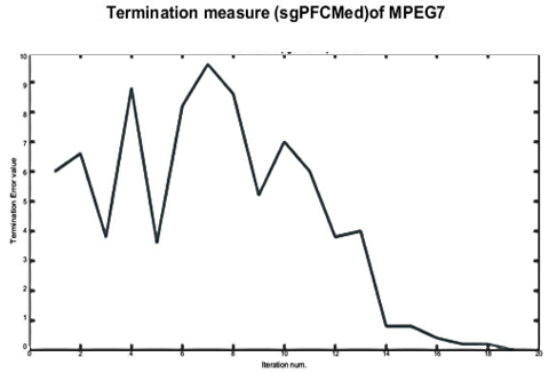
Fuzzy partitioning is carried out through an iterative optimization of the objective function of each algorithm with the update of membership and the cluster centers. Our algorithms are repeat until the algorithm has converged (that is the difference of cluster center between two iterations is no more than the given sensitivity stopping criteria or the maximum number of iteration is reached). Figure 4.3 show the sample of the termination error value of each iteration until our algorithms have converged (we only show the termination error of 5-prototypes clustering for MPEG7 dataset).



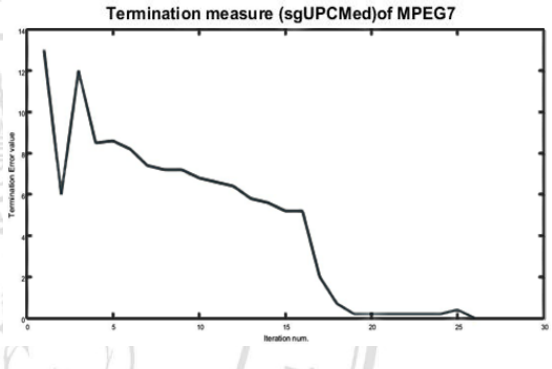
(a)Termination measure of sgFCMed



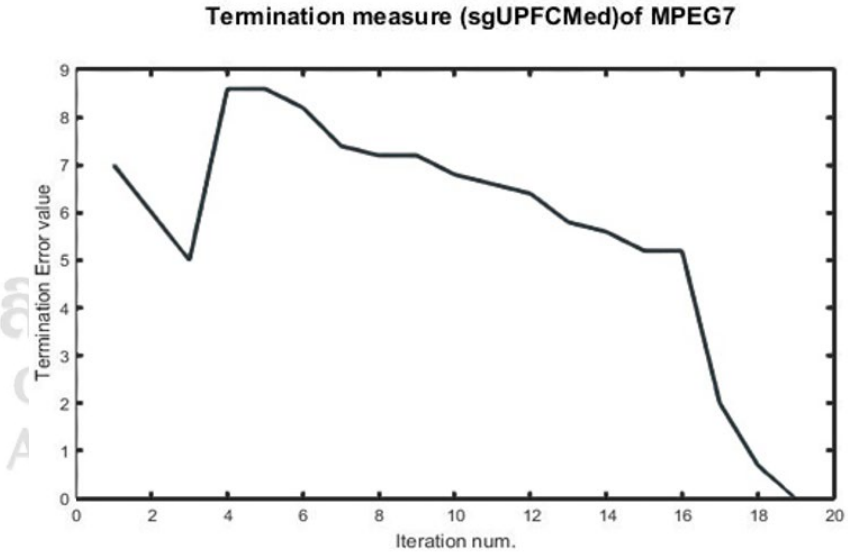
(b)Termination measure of sgFPCMed



(c)Termination measure of sgPFCMed



(d)Termination measure of sgUPCMed



(e)Termination measure of sgUPFCMed

Figure 4.3 Termination measure of MPEG7 dataset.

The performance evaluation of our five algorithms using three cluster validity indices are shown in Table 4.5. The PC of 6-prototypes of sgUPFCMed is maximum and

the PE and XB is minimum. As we expected, the 6-prototypes of sgUPFCMed provides an overall better performance in term of compact and separated clusters.

The direct and indirect comparison of the result from the MPEG-7 data set is shown in Table 4.6. We can see that our result is better than all methods in [45-49], that implement on shape or syntactic data set.

When we measured the degree of overlapping data of the dataset, the *R-value* is equal to 0.4814. Hence, this dataset is indicated that the dataset contains large area of overlapping.

In this dataset, the result of classification of five string grammar fuzzy clustering algorithms, i.e., sgFCMed, sgFPCMed, sgPFCMed, sgUPCMed and sgUPFCMed with modified fuzzy median string perform better than sgHCM for all of experiments. However, the result of sgFCMed is not good enough because the dataset contains large overlapping area among its class and contains several outlier data. The sgFCMed cannot handle outlier points in the dataset because of the constraint restriction that the sum of membership value of a data due to the object in all the clusters must be equal to one and the sgFCMed assigns membership for each object which are related to the distance of each object to the cluster centers. The sgFPCMed can solve these problems because this algorithm does not only consider the clustering problem from the viewpoint of membership theory but also considers both the clustering problem from the viewpoint of membership and possibility theories. Although sgFPCMed is created for solving noisy data, the result of sgFPCMed seems to be worse than sgPFCMed because of the constraint $\sum t_{ik} = 1$ (if the number of data is large, the typicality values of sgFPCMed tend to be very small). The sgPFCMed can improve the result of clustering in domain of string by relaxing the constraint of sgFPCMed (sum of all typicality values of all data to cluster center must be equal to one). This method has the good point of obtaining better classification accuracy than sgFCMed and sgFPCMed clustering methods for this dataset. For improving the classification accuracy result, we proposed sgUPCMed and sgUPFCMed that both objective functions based on the validity indexes PC and PE from UPCM so that the exponential membership functions are used to describe the degree of

belonging. Hence, we can detect the outlier of the dataset as well since the membership value tend to be zero where the string is too far from its prototype.

However, using only single-prototype to represent each cluster, which may not adequately model the clusters of arbitrary shape and size and hence limit the clustering performance on overlapping data. Hence, the multiple prototypes method is utilized in string grammar clustering algorithm to improve the performance of classification. The most important thing for multi-prototypes clustering is determining the number of prototypes of each class. The number of prototypes depends on the distribution, the intra-class variability and inter-class variability of each dataset. We should preserve discrimination between classes for improving the classification accuracy rate. we suggest choosing the number of prototypes of each class which large enough that noise in the data is minimized and small enough so the samples of the other classes are not included. From Table 4.4, when we ran our experiments with different number of prototype, the sgUPFCMed with 6 prototypes is the best of accuracy rate.

While we found sgUPCMed sometimes generated coincident. For example, when we ran sgUPCMed with 6-prototypes of each class, after the algorithm converged, it produced some prototypes with the same location because the algorithm can generate the same memberships values from relaxing constraint $\sum u_{ik} = 1$ because the columns and rows of the possibilistic values are independent of each other. Hence, sometimes the accuracy rate of sgUPCMed is not good enough. The sgUPFCMed can solve this problem because the objective function of sgUPFCMed based on the validity indexes PC and PE so that the exponential membership functions are used to describe the membership degree and take advantages of possibility value from PFCMed. Hence, we can detect the outlier of dataset as well and can solve coincident cluster of sgUPCMed.

Table 4.5 Cluster validity indices of sgFCMed, sgFPCMed, sgPFCMed, sgUPCMed and sgUPFCMed on MPEG-7 data set.

	sgFCMed (m=2)			sgFPCMed (m=2, η=4)			sgPFCMed (m=2, η=4)			sgUPCMed (m=2)			sgUPFCMed (m=2, η=4)		
#prototype of each class	PC	PE	XB	PC	PE	XB	PC	PE	XB	PC	PE	XB	PC	PE	XB
1	0.0311	4.7980	25.3344	0.2678	1.1342	12.3108	0.3665	1.0087	11.4964	0.3683	1.0085	11.4375	0.3772	1.0080	10.7634
2	0.0935	4.7571	25.1408	0.2945	1.1295	12.1684	0.3669	1.0085	11.3947	0.3692	1.0079	11.3278	0.3899	1.0066	10.7533
3	0.1454	4.7351	25.1198	0.3108	1.1271	12.1497	0.3692	1.0081	11.3601	0.3764	1.0071	11.3216	0.3923	1.0064	10.7516
4	0.2087	4.5761	25.0918	0.3197	1.1206	12.0741	0.3745	1.0078	11.1647	0.3792	1.0066	11.2393	0.4188	1.0061	10.7390
5	0.2673	4.5525	25.0764	0.3217	1.1131	11.8730	0.3877	1.0073	11.1548	0.3899	1.0066	11.2313	0.4256	1.0060	10.7389
6	0.3691	3.7294	19.4325	0.3852	1.0702	11.3779	0.4214	1.0034	10.1876	0.4200	1.0042	10.4025	0.5909	1.0022	9.9234
7	0.3496	3.9123	20.749	0.3723	1.0927	11.6756	0.4078	1.0038	10.2985	0.4078	1.0045	10.7796	0.5388	1.0026	9.9485
8	0.3501	4.0892	21.968	0.3779	1.0934	11.7706	0.4190	1.0040	10.2753	0.4198	1.0044	10.5703	0.4982	1.0031	9.7699
9	0.3400	4.2898	24.679	0.3698	1.1035	11.8122	0.4001	1.0043	10.3424	0.4044	1.0046	11.0492	0.4387	1.0035	10.5665
10	0.2967	4.4825	25.0593	0.3498	1.1126	11.8143	0.3898	1.0065	11.1374	0.4014	1.0054	11.0722	0.4322	1.0059	10.7237

Table 4.6 Comparison of MPEG-7 data set.

Method	Comparison method	Classification error rate (%)
sgHCM [12] with 6 prototypes	Direct	17.86
sgFCMed with 6 prototypes	-	10.71
sgFPCMed with 6 prototypes	-	7.86
sgPFCMed with 6 prototypes	-	4.29
sgUPCMed with 6 prototypes	-	5.00
sgUPFCMed with 6 prototypes	-	**2.14**
Levenshtein distance + FCM (Length of substring=4, tolerance=70%) [28]	Direct	28.24
FCM+ Levenshtein distance [29]	Direct	34.56
Curve edit distance [45]	Indirect	21.83
Shape contexts [46]	Indirect	23.49
Profile + Naïve Bayes [47]	Indirect	23.0
Profile + linear SVM [47]	Indirect	27
Shape contour descriptor + Shape similarity measures [48]	Indirect	15.67
Locally Constrained Diffusion Process (LCDP) + Inner Distance Shape Context (IDSC) [48]	Indirect	2.79

For the indirect comparison, we compared our result with those from the curve edit distance [45] shape contexts [46], Naïve Bayes [47], linear SVM [48], shape contour descriptor [49], and locally constrained diffusion process (LCDP) with inner distance shape context (IDSC) [49]. We can see that our methods provided better classification accuracy than all other methods in literatures, but not better than the method in [48]. The accuracy of the best of our method is worse than IDSC. It might be due to the synthetic points which are added to increase some information in IDSC, but our algorithm was implemented on original data set without adding any synthetic points in the dataset.

We can see that our methods provided better classification accuracy than string grammar hard clustering methods for this dataset. Our methods are more appropriate for applications where the structure of a pattern is important than numeric methods such as shape matching. However, some misclassifications have occurred in this

data set. It might be a result of there are some shapes in different class that are very similar as shown in Figure 4.4 or might be because this dataset is high intra-class variability as shown in Figure 4.5.

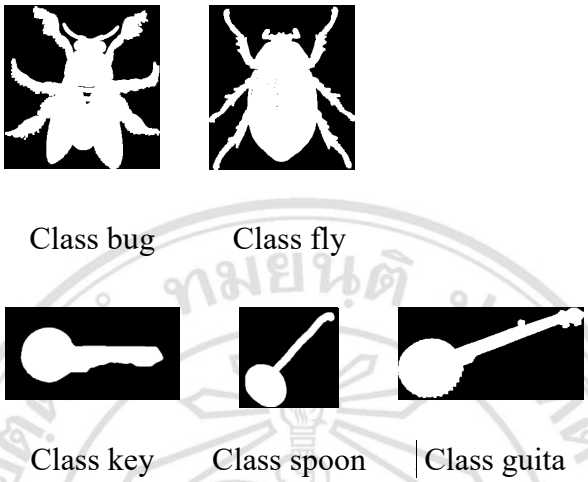


Figure 4.4 Sample of some shape in different class that are very similar.

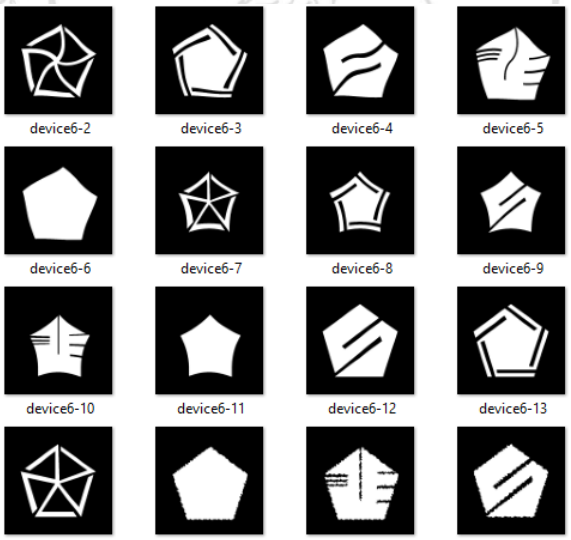


Figure 4.5 Sample of class device6 with high intra-class variability

4.1.2 Copenhagen Chromosomes Data Set

The Copenhagen chromosomes data set [52-54] was collected by Prof. Simon M. Lucas and is available for download at http://algoval.essex.ac.uk/data/sequence/.

There are 4400 strings in total from 22 non-sex chromosome type each with 200 strings. The chromosome density profiles were extracted from images of cells in the metaphases stage of cell division. The histogram of each image was encoded into a string. An example is shown in Figure 4.6. There are 44 files, each have 100 lines of the form / 7569 165 1 78 39 / A=A==a==A=a==Aa=A== a==C=====b=A=a===B==a=B====c====B==a==C==d===A==a===A==b==a , when 7569 is an identifier and 165 ∈ [1,180] is the metaphase the sample coming from. The number 1, 78, and 39 are the chromosome type, the overall string length, and the length of p-arm, respectively. The set of alphabet in this case is ∑={=,a,b,c,d,e,f,A,B,C,D,E,F}. It should be noted that we downloaded the encoded data set, not the chromosome images.

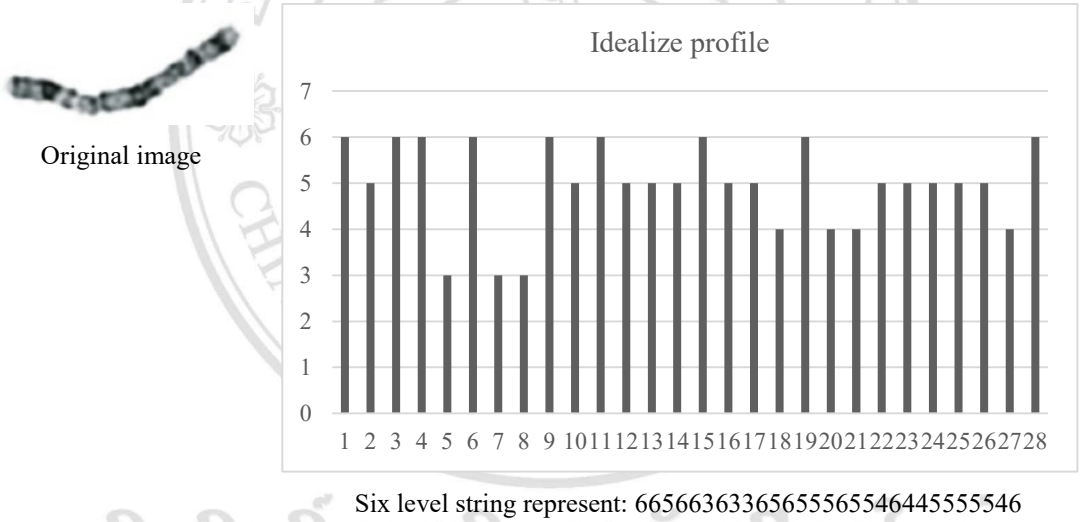


Figure 4.6 An example from Copenhagen Chromosome dataset.

The data set was divided into training and blind test data sets with 2,200 strings in each data set. The classification rates of the validation set and blind test data set of the sgFCMed, sgFPCMed, sgPFCMed and sgUPFCMed are shown in Table 4.7 to Table 4.10.

The classification accuracy rates of the ten-folds cross validation for single prototype of the sgHCM, sgFCMed, sgFPCMed, sgPFCMed, sgUPCMed and sgUPFCMed are shown in Table 4.6. We can see that the best validation result from the

sgUPFCMed m=1.5, η=2 is 90.45% correct classification, while that of the sgHCM, sgFCMed, sgFPCMed, sgPFCMed and sgUPCMed are 85.46%, 86.36%, 88.64%, 90%, and 90%, respectively. The best sgUPFCMed on test set also provides 89.66% with m=2, η=2 for the best correct classification on the blind test data set. The sgHCM, sgFCMed, sgFPCMed, sgPFCMed and sgUPCMed yield 85.46%, 84.91%, 86.37%, 87.36%, 87.41, respectively on the blind test data set. Again, the sgUPFCMed outperforms the sgHCM, sgFCMed, sgFPCMed, sgPFCMed and sgUPCMed.

Table 4.7 Single-prototype classification error rate (%) on validation sets for Copenhagen chromosomes data set.

<table>
<tr><th rowspan="2">m</th><th rowspan="2">η</th><th colspan="6">training data set
(best validation set result)</th></tr>
<tr><th>sgHCM</th><th>sgFCMed</th><th>sgFPCMed</th><th>sgPFCMed</th><th>sgUPCMed</th><th>sgUPFCMed</th></tr>
<tr><td>1.5</td><td>1.5</td><td rowspan="10">14.54</td><td rowspan="4">13.64</td><td>11.82</td><td>10.45</td><td rowspan="4">10.00</td><td>10.00</td></tr>
<tr><td>1.5</td><td>2.0</td><td>11.36</td><td>10.00</td><td>9.55</td></tr>
<tr><td>1.5</td><td>3.0</td><td>11.68</td><td>10.91</td><td>10.00</td></tr>
<tr><td>1.5</td><td>4.0</td><td>12.72</td><td>12.27</td><td>10.45</td></tr>
<tr><td>2.0</td><td>2.0</td><td rowspan="3">14.54</td><td>11.36</td><td>11.36</td><td rowspan="3">10.91</td><td>11.36</td></tr>
<tr><td>2.0</td><td>3.0</td><td>11.82</td><td>10.91</td><td>10.00</td></tr>
<tr><td>2.0</td><td>4.0</td><td>11.82</td><td>11.36</td><td>10.91</td></tr>
<tr><td>3.0</td><td>2.0</td><td rowspan="3">14.09</td><td>12.27</td><td>12.72</td><td rowspan="3">12.14</td><td>10.91</td></tr>
<tr><td>3.0</td><td>3.0</td><td>11.36</td><td>10.91</td><td>10.45</td></tr>
<tr><td>3.0</td><td>4.0</td><td>11.82</td><td>11.82</td><td>11.45</td></tr>
</table>

In order to compare the performance of these multi-prototypes string grammar clustering, i.e., sgHCM, sgFCMed, sgFPCMed, sgPFCMed, sgUPCMed and sgUPFCMed for Copenhagen Chromosome dataset. We implement these algorithms with c = 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24 and 26 prototypes with the best parameter from single-prototype of each class so that the clustering results could completely consider all situations for this Copenhagen Chromosome dataset as shown in Table 4.9 and Table 4.10.

Table 4.8 Single-prototype classification error rate (%) on blind test dataset for Copenhagen chromosomes data set.

m	η	blind test dataset					
		sgHCM	sgFCMed	sgFPCMed	sgPFCMed	sgUPCMed	sgUPFCMed
1.5	1.5	16.00	16.00	15.95	15.14	12.91	11.56
1.5	2.0			13.72	13.05		12.54
1.5	3.0			13.63	12.86		11.32
1.5	4.0			15.91	15.09		11.36
2.0	2.0		15.09	13.63	12.64	12.59	**10.34**
2.0	3.0			13.77	13.55		11.87
2.0	4.0			13.77	13.05		11.21
3.0	2.0		16.05	13.91	15.59	13.50	11.54
3.0	3.0			13.86	13.50		11.59
3.0	4.0			13.81	13.45		11.56

Table 4.9 Multi-prototypes classification error rate (%) on validation sets for Copenhagen chromosomes data set.

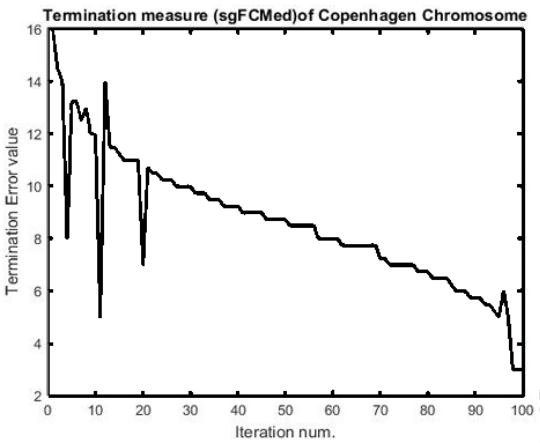
# prototype of each class	training data set (best validation set result)					
	sgHCM	sgFCMed (m=1.5)	sgFPCMed (m=1.5, η=2)	sgPFCMed (m=1.5,η=2)	sgUPCMed (m=1.5)	sgUPFCMed (m=1.5, η=2)
2	13.46	9.69	9.28	8.34	8.07	7.97
4	13.41	9.65	9.18	8.11	8.05	7.87
6	11.67	9.56	8.66	7.62	7.75	7.52
8	11.64	9.40	8.60	7.58	7.52	7.01
10	11.56	9.31	8.55	7.27	7.49	6.54
12	11.24	9.09	8.29	6.81	7.49	6.25
14	10.60	8.51	8.11	6.68	7.43	5.83
16	10.20	8.38	8.08	6.56	7.24	5.78
18	10.15	7.99	8.00	6.52	7.10	5.46
20	9.98	7.92	7.90	6.47	6.63	**4.78**
22	10.08	7.99	7.94	6.57	6.74	5.05
24	10.14	8.14	8.05	6.63	6.89	5.18
26	10.19	8.42	8.15	6.84	7.03	5.89

Table 4.10 Multi-prototypes classification error rate (%) on blind test dataset for Copenhagen chromosomes data set.

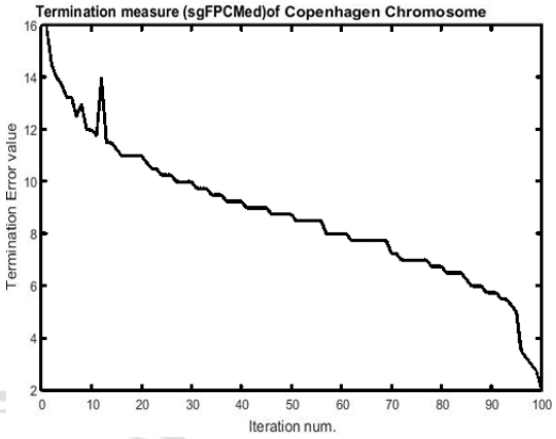
# prototype of each class	blind test data set					
	sgHCM	sgFCMed (m=2)	sgFPCMed (m=2, η=2)	sgPFCMed (m=2, η=2)	sgUPCMed (m=2)	sgUPFCMed (m=2, η=2)
2	14.88	14.79	11.32	10.22	10.32	8.07
4	14.82	13.37	10.70	9.18	9.27	7.92
6	14.61	13.15	9.04	8.94	9.26	7.91
8	14.48	13.07	8.61	8.51	9.20	7.90
10	14.35	12.18	8.31	7.54	8.99	7.82
12	14.32	12.06	8.29	7.29	7.95	7.44
14	14.12	11.94	7.93	7.14	7.77	7.36
16	13.72	11.86	7.86	7.10	7.31	6.64
18	11.68	11.17	7.60	6.95	6.97	6.59
20	10.84	10.49	7.27	6.61	6.58	**6.34**
22	10.99	10.84	7.89	6.82	6.74	6.52
24	11.13	11.08	8.12	7.02	6.93	6.63
26	11.43	11.29	8.28	7.44	7.01	6.87

We can see that 20 prototypes of sgUPFCMed gives 95.22% correct classification rate on the validation set with m=1.5, η=2. Whereas the sgHCM, sgFCMed, sgFPCMed, sgPFCMed and sgUPCMed provide 90.02%, 92.08%, 92.10, 93.53%, and 93.37% correct classification, respectively, on the best validation set. The best sgUPFCMed on test set also provides 93.66% with m=2, η=2 for the best correct classification on the blind test data set. The sgHCM, sgFCMed, sgFPCMed, sgPFCMed and sgUPCMed yield 89.16%, 89.51%, 92.73, 93.39%, and 93.42 %, respectively on the blind test data set.

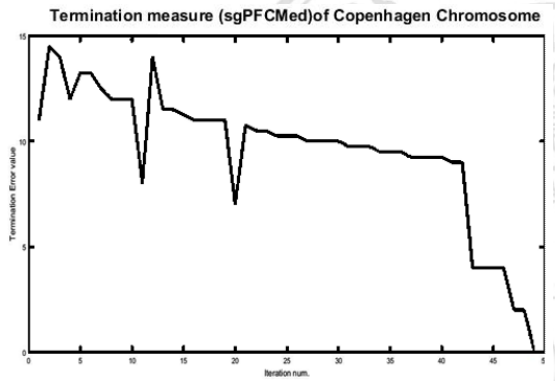
Figure 4.7 show the sample of the termination error value of each iteration until our algorithms have converged (we only show the termination error of 4-prototypes clustering for Copenhagen Chromosome dataset).



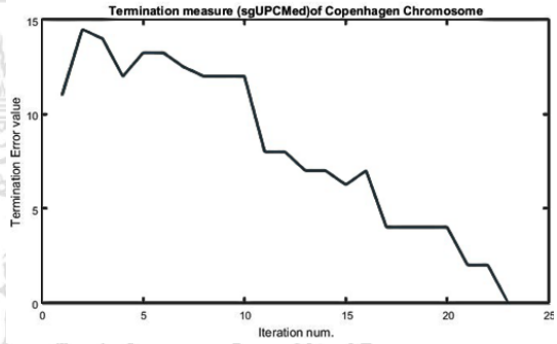
(a)Termination measure of sgFCMed



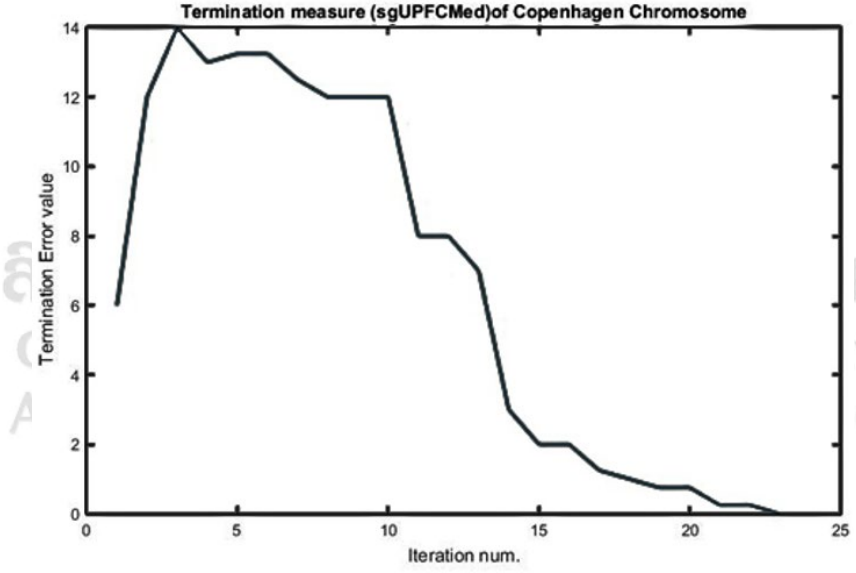
(b)Termination measure of sgFPCMed



(c)Termination measure of sgPFCMed



(d)Termination measure of sgUPCMed



(e)Termination measure of sgUPFCMed

Figure 4.7 Termination measure of Copenhagen Chromosome dataset.

The sgUPFCMed outperforms the sgHCM, sgFCMed, sgFPCMed, sgPFCMed and sgUPCMed. An overall comparison between the performances on the partition of clusters is shown in Table 4.12. As one can observe, sgUPFCMed partition has higher partition coefficient and the PE and the XB index are lower which indicates a good clustering. As expected, sgUPFCMed model outperforms sgHCM, sgFCMed, sgFPCMed, sgPFCMed and sgUPCMed models.

For indirect comparison, we compared our results with the 12-NN normalized distance [2] ECGI algorithm [55], and Multilayer Perceptron [55]. The direct and indirect comparison is shown in Table 4.11. We can see that our results are better than all other algorithms.

When we measured the degree of overlapping data of the dataset, the *R-value* was equal to 0.2227. This indicated that the dataset did not contain large area of overlapping.

Similar to the MPEG-7 data set, the result of classification of our five string grammar fuzzy clustering algorithms on Copenhagen chromosomes data set are better than sgHCM for all of experiments. The sgUPFCMed provide the best classification rate than sgFCMed, sgFPCMed, sgPFCMed, sgUPCMed with same reasons of the previous dataset (the MPEG-7 data set). Again, using only single-prototype to represent each cluster, which may not provide good result because of the limiting of the clustering performance on overlapping data. The multi-prototypes method was also utilized in this dataset to improve the performance of classification. When we also ran multi-prototypes, the results show that sgUPFCMed with 20 prototypes is the best of classification results. The number of prototypes of this dataset is large might be because the intra-class variability of this dataset is low and inter-class variability of this dataset is high.

For the indirect comparison, we compared our result with those from the 12-NN normalized distance [2] ECGI algorithm [55], and Multilayer Perceptron [55]. We can see that the best of our method with 20 prototypes provides better classification accuracy than other methods in literatures. Again, our methods provide better classification accuracy than string grammar hard clustering methods and other methods in works of literature for this dataset.

Table 4.11 Comparison of Copenhagen chromosomes data set

Method	Comparison method	Classification error rate (%)	
		Training set	Test set
sgHCM [12] with 20 prototypes	Direct	9.98	10.84
sgFCMed with 20 prototypes	-	7.27	9.86
sgFPCMed with 20 prototypes	-	7.10	9.21
sgPFCMed with 20 prototypes	-	5.82	7.24
sgUPCMed with 20 prototypes	-	6.63	6.58
sgUPFCMed with 20 prototypes	-	**4.78**	**6.34**
Levenshtein distance + FCM (Length of substring=4, tolerance=70%) [28]	Direct	19.55	20.98
FCM+ Levenshtein distance [29]	Direct	26.82	29.34
12-NN normalized distance [2]	Indirect	4.9	-
ECGI algorithm [55]	Indirect	7.5	-
Multilayer Perceptron [55]	Indirect	9.1	-

Table 4.12 Cluster validity indices of sgFCMed, sgFPCMed, sgPFCMed, sgUPCMed and sgUPFCMed on Copenhagen chromosomes data set.

	sgFCMed (m=1.5)			sgFPCMed (m=1.5, η=2)			sgPFCMed (m=1.5, η=2)			sgUPCMed (m=1.5)			sgUPFCMed (m=1.5, η=2)		
#prototype of each class	PC	PE	XB	PC	PE	XB	PC	PE	XB	PC	PE	XB	PC	PE	XB
1	0.127	2.997	18.9143	0.197	1.434	18.675	0.2440	1.398	18.322	0.2593	1.4133	17.9926	0.294	1.321	18.210
2	0.1321	2.9867	18.8616	0.1244	1.4219	18.6564	0.2463	1.3874	18.6564	0.2606	1.4112	17.9707	0.2552	1.3176	18.2069
4	0.1337	2.9026	18.5756	0.1261	1.4193	18.5118	0.2473	1.3842	18.5118	0.2632	1.4112	17.6623	0.2784	1.3152	18.1889
6	0.1340	2.8782	18.5563	0.1268	1.4011	18.3725	0.2522	1.3706	18.3725	0.2895	1.3787	17.6177	0.2826	1.3141	18.0975
8	0.1369	2.8067	18.5269	0.1305	1.3966	18.2891	0.2610	1.3688	18.2891	0.2938	1.3691	17.5643	0.2841	1.3065	17.9758
10	0.1381	2.7586	18.4625	0.1319	1.3915	18.2761	0.2617	1.3495	18.2761	0.3017	1.3682	17.4469	0.2866	1.2943	17.9106
12	0.1389	2.7040	18.4488	0.1347	1.3910	18.1591	0.2714	1.3438	18.1591	0.3059	1.3638	17.3252	0.2907	1.2921	17.8921
14	0.1392	2.6704	18.4290	0.1375	1.3786	17.9761	0.2747	1.3338	17.9761	0.3082	1.3483	17.3222	0.2941	1.2906	17.8678
16	0.1411	2.4921	18.3640	0.1384	1.3785	17.9587	0.2771	1.3282	17.9587	0.3190	1.3373	17.3099	0.3104	1.2901	17.3688
18	0.1423	2.3497	18.1483	0.1402	1.3783	17.8798	0.3089	1.3266	17.8798	0.3259	1.3365	17.2936	0.3144	1.2840	17.3201
20	0.1431	1.984	17.6844	0.3520	1.372	17.4800	0.3380	1.2800	17.410	0.3311	1.3037	17.2047	0.3870	1.2750	16.973
22	0.1425	2.254	18.3450	0.1478	1.3945	17.8798	0.3078	1.3543	17.6756	0.3254	1.3334	17.2765	0.3167	1.2876	17.334
24	0.1415	2.262	18.3678	0.1412	1.432	17.9776	0.2986	1.3964	17.6954	0.3135	1.3376	17.2965	0.3075	1.2965	17.678
26	0.1411	2.269	18.3987	0.1265	1.4865	17.9954	0.2646	1.4076	17.7054	0.3076	1.3468	17.3754	0.2896	1.3154	17.975

4.1.3 MNIST Database of Handwritten Digits

The MNIST database of handwriting digit data set is available from http://algoval.essex.ac.uk/data/sequence/ as described in [56-59] was collected by Prof. Simon M. Lucas. It contains training set of 60,000 examples, and a test set of 10,000 examples. Digits in data set ranging from 0 to 9. Table 4.13 shows the number of each digit in training and test sets. Examples are shown in Figure 4.8.

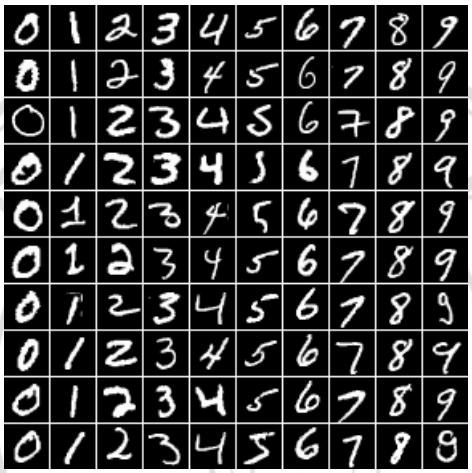


Figure 4.8 Examples of images in the MNIST data set.

Table 4.13 Number of samples in each digit in training and test sets of MNIST database of handwritten digits

	0	1	2	3	4	5	6	7	8	9
Training	5923	6742	5958	6131	5842	5421	5918	6265	5851	5949
Blind test	980	1135	1032	1010	982	892	958	1028	974	1009

Again, we acquired the string data set from the website, not the original images. The classification results on the validation set and the blind test set from the sgHCM, sgFCMed, sgFPCMed, sgPFCMed, sgUPCMed and sgUPFCMed are shown in Tables 4.14 and 4.15. We can see that the best correct classification rate on the validation set for single prototype is 98.07% from sgUPFCMed with m=1.5, η=1.5 from and the best correct classification rate on the blind test dataset for single-prototype is 98.46% from sgUPFCMed with m=1.5, η=1.5.

Table 4.14 Single-prototype classification error rate (%) on validation sets for the MNIST data set.

m	η	training data set (best validation set result)					
		sgHCM	sgFCMed	sgFPCMed	sgPFCMed	sgUPCMed	sgUPFCMed
1.5	1.5	2.98	1.97	1.97	1.97	1.97	**1.93**
1.5	2.0			1.97	1.97		1.95
1.5	3.0			1.98	1.98		1.98
1.5	4.0			2.00	1.98		1.98
2.0	2.0		2.57	1.98	1.98	1.98	1.98
2.0	3.0			2.00	2.00		1.98
2.0	4.0			2.00	1.98		1.97
3.0	2.0		2.78	2.05	2.03	2.00	2.00
3.0	3.0			2.07	2.03		1.98
3.0	4.0			2.03	2.00		1.98

Table 4.15 Single-prototype classification error rate (%) on blind test dataset for MNIST database of handwritten digits

m	η	training data set (best validation set result)					
		sgHCM	sgFCMed	sgFPCMed	sgPFCMed	sgUPCMed	sgUPFCMed
1.5	1.5	2.77	1.58	1.58	1.56	1.56	**1.54**
1.5	2.0			1.57	1.54		1.54
1.5	3.0			1.58	1.58		1.55
1.5	4.0			1.59	1.59		1.56
2.0	2.0		1.63	1.59	1.57	1.58	1.56
2.0	3.0			1.59	1.57		1.56
2.0	4.0			1.60	1.62		1.58
3.0	2.0		1.68	1.64	1.63	1.62	1.60
3.0	3.0			1.65	1.63		1.62
3.0	4.0			1.62	1.62		1.58

Table 4.16 Multi-prototypes classification error rate (%) on validation sets for MNIST database of handwritten digits

# prototype of each class	training data set (best validation set result)					
	sgHCM	sgFCMed (m=1.5)	sgFPCMed (m=1.5,η=1.5)	sgPFCMed (m=1.5, η=1.5)	sgUPCMed (m=1.5)	sgUPFCMed (m=1.5, η=1.5)
5	2.65	1.55	1.56	1.48	1.50	1.34
10	2.64	1.52	1.56	1.24	1.22	1.12
15	2.62	1.48	1.41	1.12	1.16	1.08
20	2.62	1.48	1.41	1.12	1.16	1.08
25	2.60	1.48	1.25	1.06	1.06	1.07
30	2.59	1.43	1.21	1.02	1.06	1.05
35	2.49	1.43	1.18	0.98	1.04	1.05
40	2.02	1.42	1.16	0.98	1.02	0.92
45	1.98	1.38	1.14	0.95	1.02	0.90
50	1.93	1.38	1.12	0.95	0.98	**0.88**

Table 4.17 Multi-prototypes classification error rate (%) on blind test dataset for MNIST database of handwritten digits

# prototype of each class	blind test data set					
	sgHCM	sgFCMed (m=1.5)	sgFPCMed (m=1.5, η=1.5)	sgPFCMed (m=1.5, η=1.5)	sgUPCMed (m=1.5)	sgUPFCMed (m=1.5, η=1.5)
5	2.45	1.59	1.56	1.48	1.43	1.34
10	2.37	1.56	1.53	1.46	1.29	1.28
15	2.37	1.53	1.48	1.42	1.23	1.12
20	2.37	1.52	1.46	1.37	1.22	1.08
25	2.35	1.49	1.43	1.37	1.14	1.03
30	2.34	1.45	1.41	1.34	1.13	0.95
35	2.25	1.42	1.38	1.21	1.10	0.92
40	2.14	1.38	1.38	1.15	1.10	0.87
45	2.10	1.35	1.35	1.12	1.09	0.85
50	2.09	1.32	1.32	0.98	1.08	**0.82**

We tested the performance of the following multi-prototypes string grammar clustering, i.e., sgHCM, sgFCMed, sgFPCMed, sgPFCMed, sgUPCMed and sgUPFCMed. We implement these algorithms with c = 5, 10, 15, 20, 25, 30, 35, 40, 45 and 50 prototypes of each class as shown in Tables 4.16 and 4.17. The results show that 50 prototypes of sgUPFCMed gives 99.12% correct classification rate on the validation set with m=1.5, η=1.5, Whereas the sgHCM, sgFCMed, sgFPCMed, sgPFCMed and sgUPCMed provide 98.07%, 98.62%, 98.88%, 99.05%, and 99.02% correct classification, respectively, on the best validation set. The best sgUPFCMed on test set

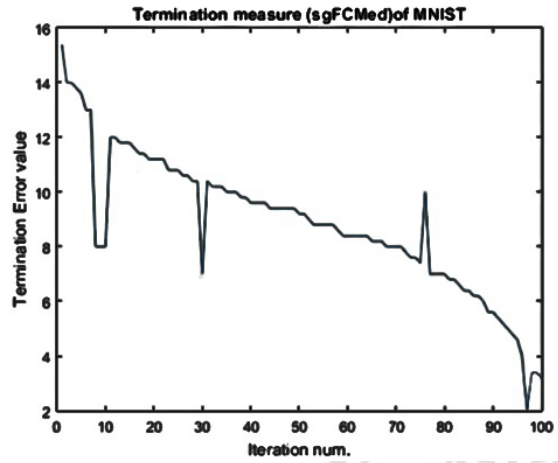
also provides 99.18% with m=1.5, η=1.5 for the best correct classification on the blind test data set. The sgHCM, sgFCMed, sgFPCMed, sgPFCMed and sgUPCMed yield 97.91%, 98.68%, 98.68%, 99.02%, and 98.92%, respectively on the blind test data set.

A comparison of the cluster validity of the algorithms is reported in Table 4.17. All three indices show that the sgUPFCMed provide more compact and separated clusters than the sgHCM, sgFCMed, sgFPCMed, sgPFCMed, and sgUPCMed. Again, we compare our result directly with that are similar algorithms including those in [28] and [29]. Since there have been several algorithms tested on this data set. However, all of them work only with numeric vectors not syntactic data set. We indirectly compare our sgUPFCMed result with some of the algorithms [56–59] as shown in Table 4.19.

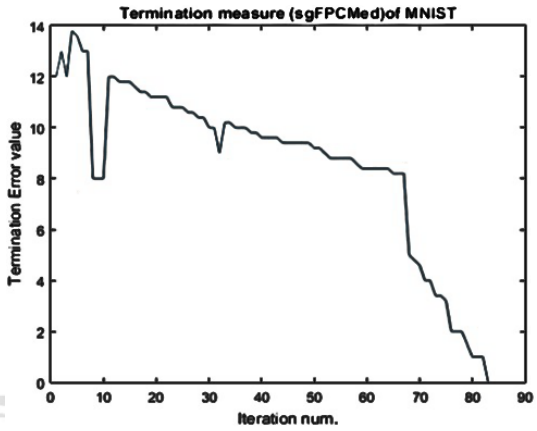
We computed the degree of overlapping of the dataset, the *R-value* was 0.1390. Hence, this dataset is indicated that the dataset contains very small area of overlapping.

Similar to the MPEG-7 and Copenhagen chromosomes data set, the result of classification of our five string grammar fuzzy clustering algorithms on MNIST data set data set are better than sgHCM for all of experiments with vary parameter m and η. Again, the sgUPFCMed provides the best classification rate than sgFCMed, sgFPCMed, sgPFCMed, sgUPCMed with same reasons of the all previous dataset. Even though this dataset contains very small area of overlapping data, using the multi-prototypes method can also improve the performance of classification on this dataset. The sgUPFCMed with 50 prototypes provide the best of classification results. Because of this dataset contains very small area of overlapping, the large number of prototype give the best accuracy rate.

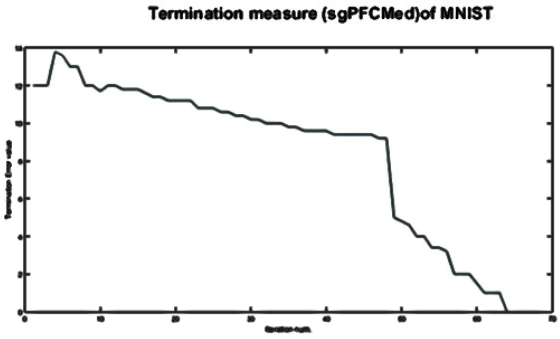
Figure 4.9 show the sample of the termination error value of each iteration until our algorithms have converged (we only show the termination error of 5-prototypes clustering for MNIST dataset).



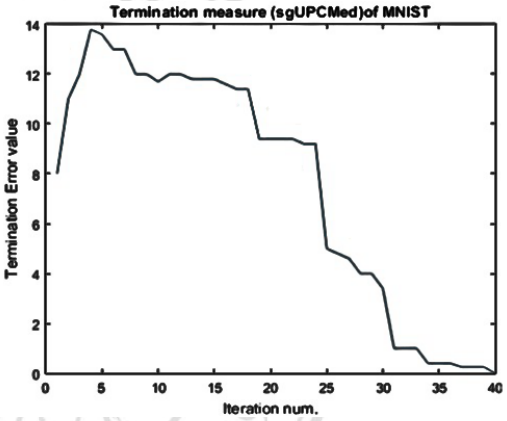
(a)Termination measure of sgFCMed



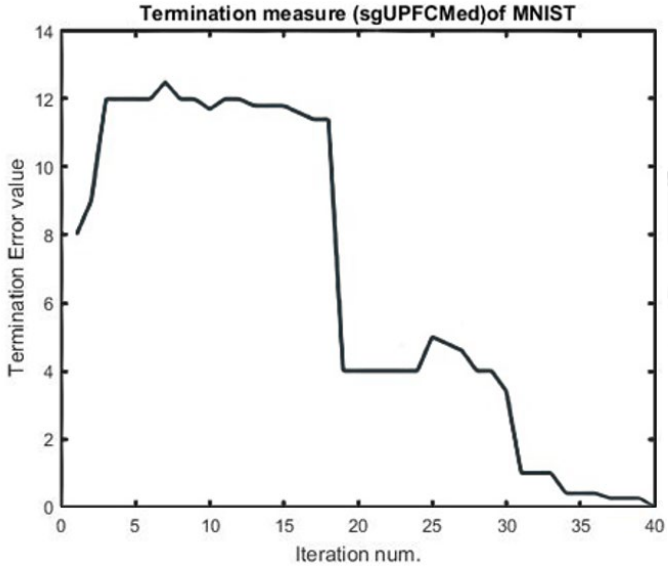
(b)Termination measure of sgFPCMed



(c)Termination measure of sgPFCMed



(d)Termination measure of sgUPCMed



(e)Termination measure of sgUPFCMed

Figure 4.9 Termination measure of MNIST dataset.

Table 4.18 Cluster validity indices of sgFCMed, sgFPCMed, sgPFCMed, sgUPCMed and sgUPFCMed on MNIST database of handwritten digits.

#prototype of each class	sgFCMed (m=1.5)			sgFPCMed (m=1.5, η=1.5)			sgPFCMed (m=1.5, η=1.5)			sgUPCMed (m=1.5)			sgUPFCMed (m=1.5, η=1.5)		
	PC	PE	XB	PC	PE	XB	PC	PE	XB	PC	PE	XB	PC	PE	XB
1	0.7966	0.9087	5.0126	0.8498	0.5010	4.4325	0.8618	0.4976	4.4989	0.8620	0.4974	4.4987	0.8623	0.497	4.4984
5	0.8009	0.9004	5.0123	0.8506	0.5004	4.3967	0.8622	0.4962	4.4966	0.8638	0.4956	4.4942	0.8628	0.4954	4.4976
10	0.8024	0.9004	5.0119	0.8530	0.5001	4.3905	0.8629	0.4918	4.4944	0.8633	0.4915	4.4940	0.8634	0.4911	4.4939
15	0.8143	0.8999	5.0119	0.8545	0.4998	4.3786	0.8634	0.4898	4.4938	0.8643	0.4900	4.4931	0.8643	0.4814	4.4921
20	0.8124	0.8997	5.0117	0.8567	0.4953	4.3690	0.8645	0.4872	4.4923	0.8647	0.4868	4.4921	0.8651	0.4865	4.4917
25	0.8208	0.8965	5.0134	0.8612	0.4924	4.3578	0.8712	0.4867	4.4367	0.8712	0.4844	4.4567	0.8689	0.4835	4.4323
30	0.8214	0.8954	5.0143	0.8635	0.4823	4.3456	0.8724	0.4831	4.4102	0.8728	0.4828	4.4098	0.8731	0.4827	4.4094
35	0.8345	0.8959	5.0118	0.8633	0.4768	4.2856	0.8742	0.4804	4.4197	0.8789	0.4793	4.4075	0.8789	0.4814	4.4045
40	0.8465	0.8967	5.0112	0.8678	0.4923	4.2394	0.8798	0.4792	4.4023	0.8802	0.4788	4.4021	0.8804	0.4787	4.4018
45	0.8532	0.8654	5.0096	0.8712	0.4865	4.2134	0.8804	0.4623	4.4002	0.8813	0.4678	4.2154	0.8823	0.4622	4.3998
50	0.8923	0.8345	5.0043	0.8767	0.4734	4.1856	0.8834	0.4589	4.3982	0.8838	0.4587	4.3981	0.8842	0.4586	4.3978

Table 4.19 Comparison of the MNIST data set.

Method	Comparison Method	Classification error rate (%)	
		Training set	Test set
sgHCM [12] with 50 prototypes	Direct	1.93	2.09
sgFCMed with 50 prototypes	-	1.38	1.32
sgFPCMed with 50 prototypes	-	1.12	1.32
sgPFCMed with 50 prototypes	-	0.95	0.98
sgUPCMed with 50 prototypes	-	0.98	1.08
sgUPFCMed with 50 prototypes	-	0.88	**0.82**
Levenshtein distance + FCM (Length of substring=4, tolerance=70%) [32]	Direct	12.78	15.12
FCM+ Levenshtein distance [33]	Direct	13.80	15.99
K-NN with non-linear deformation (P2DHMDM)[55]	Indirect	-	0.52
Product of stumps on Haar features [56]	Indirect	-	0.87
Convolutional neural network with pixel-based feature [57]	Indirect	-	0.74
SVM RBF with gradient-based feature [57]	Indirect	-	0.57
SVM linear with gradient-based feature [57]	Indirect	-	1.34
SVM polynomial with gradient-based feature [57]	Indirect	-	0.47
Extended tangent distance [58]	Indirect	-	1.00

For the indirect comparison, we compare our result with those from the K-NN with non-linear deformation (P2DHMDM) [56], Product of stumps on Haar features [57], Convolutional neural network with pixel-based feature [58], SVM RBF with gradient-based feature [58], SVM linear with gradient-based feature [58], SVM polynomial with gradient-based feature [58], Extended tangent distance [59]. The results show that the best of our method with 50 prototypes provides better classification accuracy than some methods from the numeric-based algorithms such as the P2DHMDM [56], Product of stumps on Haar features [57], SVM linear with gradient-based feature [58], and Extended tangent distance [59]. However, our sgUPFCMed not as good as the methods of Convolutional neural network with pixel-based feature [58], SVM-RBF with gradient-based feature [58], SVM polynomial with gradient-based feature [58]. This

might be because we implement our algorithm on original data without any pre-processing. Again, our methods provide better classification accuracy than string grammar hard clustering methods and decision theoretic methods in works of literatures for this dataset. However, some misclassifications have occurred in this data set. It might be a result of there are some noisy samples in the dataset as shown in Figure 4.10.

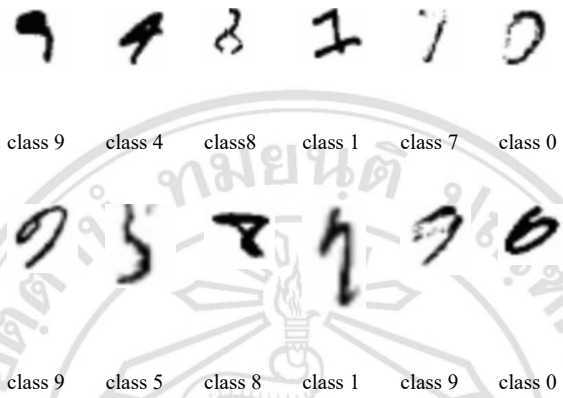


Figure 4.10 Examples of noisy data in MNIST dataset

4.1.4 USPS Database of Handwritten Digits

The USPS handwritten digit data set is a well-known US Postal Service handwritten digit recognition data set. Examples of the data set collected by Prof. Simon M. Lucas and downloaded from http://algoval.essex.ac.uk/data/sequence/ [56-59] are shown in Figure 4.11. Again, the encoded strings were downloaded from the website.

There are 9,298 strings for digit numbers 0 to 9. In this case, the data set was divided into 7,291 training strings and 2,007 blind test strings. The numbers of samples of all digit classes in both data sets are shown in Table 4.20.

Table 4.20 Number of samples in each digit in training and blind test sets of USPS database of handwritten digits.

	0	1	2	3	4	5	6	7	8	9
Training	1194	1005	731	658	652	556	664	645	542	644
Blind Test	359	264	198	166	200	160	170	147	166	177

Table 4.21 Single-prototype classification error rate (%) on validation sets for the USPS data set.

m	η	training data set (best validation set result)					
		sgHCM	sgFCMed	sgFPCMed	sgPFCMed	sgUPCMed	sgUPFCMed
1.5	1.5	8.09	6.14	4.84	4.63	4.84	4.12
1.5	2.0			4.74	4.74		4.24
1.5	3.0			4.84	4.74		4.24
1.5	4.0			4.94	4.80		4.25
2.0	2.0		6.12	4.84	4.84	4.74	4.66
2.0	3.0			4.84	4.74		4.53
2.0	4.0			4.84	4.84		4.35
3.0	2.0		5.77	4.93	4.94	4.74	4.53
3.0	3.0			5.08	4.94		4.66
3.0	4.0			4.74	4.63		**3.95**

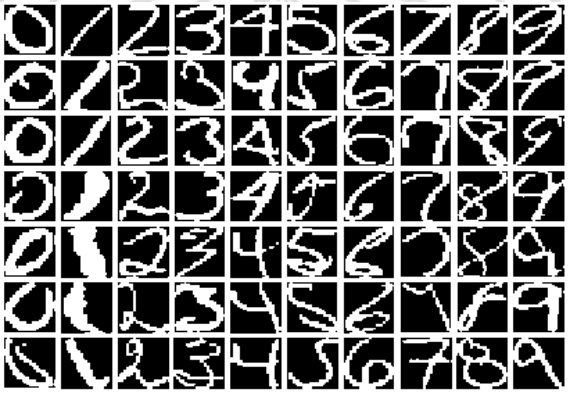


Figure 4.11 Examples of images in the USPS data set.

The classification results on the validation set and the blind test set from the sgFCMed, sgFPCMed, sgPFCMed and sgUPFCMed are shown in Tables 4.21 to 4.24.

We can see that the single-prototype sgUPFCMed gives 96.05% correct classification rate on the validation set with m=3, η=4. Whereas the sgHCM, sgFCMed, sgFPCMed, sgPFCMed and sgUPFCMed provide 91.91%, 94.23%, 95.26%, 95.37%, 95.26% correct classification, respectively, on the best validation set. The best sgUPFCMed yields 94.79% correct classification on the blind test data set with m=3, η=4. The sgHCM, sgFCMed, sgFPCMed, sgPFCMed and sgUPFCMed yield 87.90%, 92.63%, 93.97%, 94.02% and 94.02% respectively on the blind test data set. Again, the sgUPFCMed outperformed the sgHCM, sgFCMed, sgFPCMed, sgPFCMed and sgUPFCMed.

Table 4.22 Single-prototype classification error rate (%) on blind test dataset for USPS data set.

m	η	training data set (best validation set result)					
		sgHCM	sgFCMed	sgFPCMed	sgPFCMed	sgUPCMed	sgUPFCMed
1.5	1.5	12.10	6.73	6.58	6.53	6.64	6.34
1.5	2.0			6.58	6.58		6.42
1.5	3.0			6.43	6.38		6.42
1.5	4.0			6.67	6.57		6.38
2.0	2.0		6.38	6.18	6.13	6.18	6.12
2.0	3.0			6.18	6.18		6.09
2.0	4.0			6.13	6.08		5.83
3.0	2.0		7.37	6.07	6.07	5.98	5.92
3.0	3.0			6.07	6.03		5.87
3.0	4.0			6.03	5.93		**5.21**

We can see that the single-prototype sgUPFCMed gives 96.05% correct classification rate on the validation set with m=3, η=4. Whereas the sgHCM, sgFCMed, sgFPCMed, sgPFCMed and sgUPFCMed provide 91.91%, 94.23%, 95.26%, 95.37%, 95.26% correct classification, respectively, on the best validation set. The best sgUPFCMed yields 94.79% correct classification on the blind test data set with m=3, η=4. The sgHCM, sgFCMed, sgFPCMed, sgPFCMed and sgUPFCMed yield 87.90%, 92.63%, 93.97%, 94.02% and 94.02% respectively on the blind test data set. Again, the sgUPFCMed is better than the sgHCM, sgFCMed, sgFPCMed, sgPFCMed and sgUPFCMed.

Table 4.23 Multi-prototypes classification error rate (%) on validation sets for USPS.

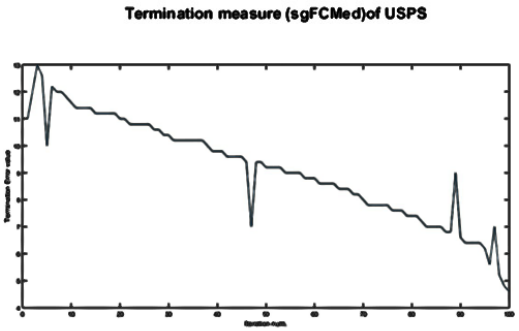
# prototype of each class	training data set (best validation set result)					
	sgHCM	sgFCMed (m=3)	sgFPCMed (m=3, η=4)	sgPFCMed (m=3, η=4)	sgUPCMed (m=3)	sgUPFCMed (m=3, η=4)
5	7.45	5.43	4.56	4.45	4.42	3.95
10	6.92	4.65	4.38	4.29	4.32	3.91
15	6.89	4.63	4.36	4.25	4.32	3.89
20	6.89	4.54	4.35	4.21	4.31	3.88
25	6.83	4.53	4.33	4.12	4.12	3.56
30	6.78	4.53	4.23	3.98	4.08	3.45
35	6.53	4.46	4.09	3.76	3.89	3.23
40	6.49	4.43	3.98	3.56	3.76	**3.07**
45	6.52	4.49	4.07	3.64	3.89	3.32
50	6.67	4.53	4.39	3.70	3.98	3.43

Table 4.24 Multi-prototypes classification error rate (%) on blind test dataset for USPS data set.

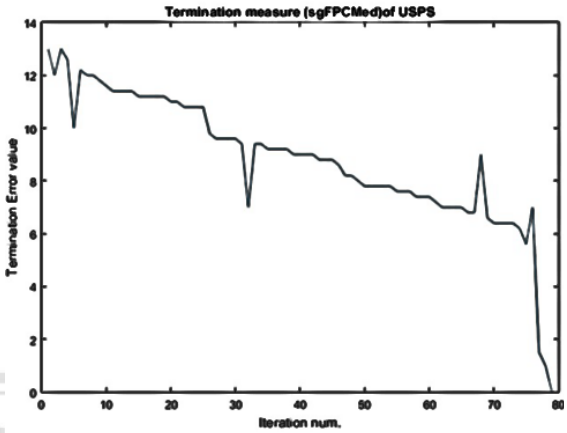
# prototype of each class	blind test data set					
	sgHCM	sgFCMed (m=3)	sgFPCMed (m=3, η=4)	sgPFCMed (m=3, η=4)	sgUPCMed (m=3)	sgUPFCMed (m=3, η=4)
5	12.00	6.24	5.93	5.76	5.34	5.21
10	11.25	5.91	5.25	4.89	4.88	4.11
15	11.25	5.91	5.25	4.89	4.88	4.11
20	11.11	5.78	5.13	4.78	4.78	3.99
25	11.10	5.72	5.09	4.65	4.68	3.93
30	11.09	5.67	5.03	4.54	4.87	3.89
35	10.76	5.58	4.56	4.34	4.56	3.76
40	10.34	5.43	4.28	3.95	4.08	**3.06**
45	10.98	5.67	4.76	4.56	4.67	3.78
50	11.23	5.89	5.14	4.87	4.98	4.03

To compare the performance of these multi-prototypes string grammar clustering, i.e. sgFCMed, sgFPCMed, sgPFCMed, sgUPCMed and sgUPFCMed. We implement these algorithms with c = 5, 10, 15, 20, 25, 30, 35, 40, 45 and 50 prototypes of each class as shown in Tables 4.23 and 4.24. We can see that 40 prototypes of sgUPFCMed gives 96.93% correct classification rate on the validation set with m=3, η=4, Whereas the sgFCMed, sgFPCMed, sgPFCMed and sgUPFCMed provide 93.76%, 95.75%, 96.02%, 96.44%, and 96.24% correct classification, respectively, on the best validation set. The best sgUPFCMed on test set also provides 96.94% with m=3, η=4 for the best correct classification on the blind test data set. The sgHCM, sgFCMed, sgFPCMed, sgPFCMed and sgUPCMed yield 89.66%, 94.57%, 95.72%, 96.05%, and 95.92%, respectively on the blind test data set. All three indices show that the sgUPFCMed with 5 prototypes gives better clusters than sgHCM, sgFCMed, sgFPCMed, sgPFCMed and sgUPCMed as shown in Table 4.26.

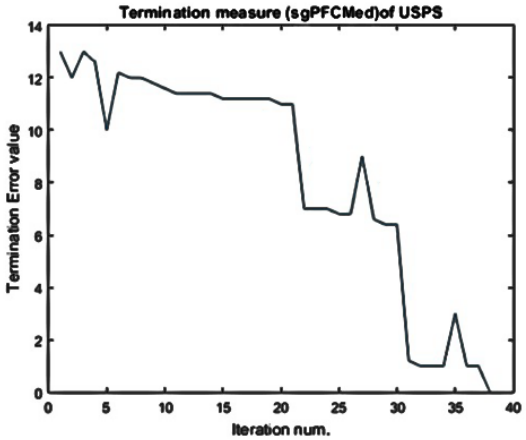
Figure 4.12 show the sample of the termination error value of each iteration until our algorithms have converged (we only show the termination error of 5-prototypes clustering for USPS dataset).



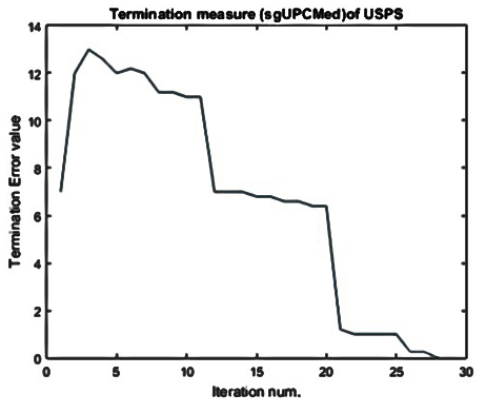
(a)Termination measure of sgFCMed



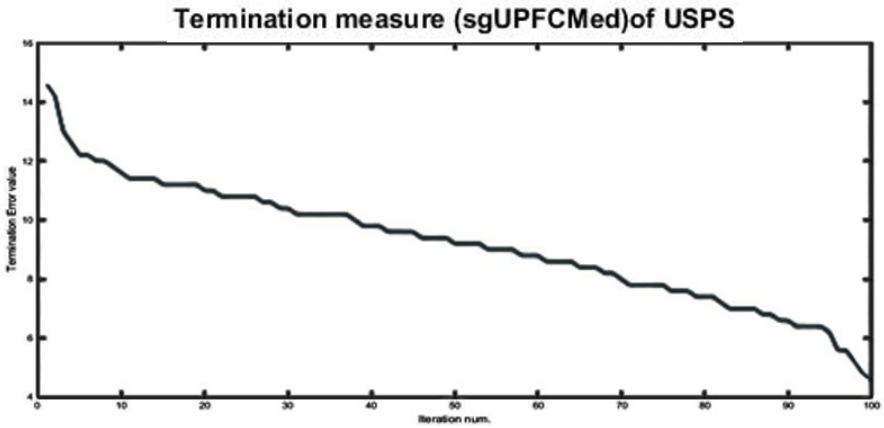
(b)Termination measure of sgFPCMed



(c)Termination measure of sgPFCMed



(d)Termination measure of sgUPCMed



(e)Termination measure of sgUPFCMed

Figure 4.12 Termination measure of USPS dataset.

Again, the direct comparison is implemented for the same algorithms including those from in [32] and [33]. For the same reason as in MNIST data set, we indirectly compare our result with those algorithms from [56-59]. Table 4.25 shows the direct and indirect comparison.

When we measure the degree of overlapping data of the dataset, the *R-value* is equal to 0.1920. Again, this dataset is indicated that the dataset contains very small area of overlapping.

Table 4.25 Indirect comparison of the USPS data set.

Method	Comparison Method	Classification error rate (%)	
		Training set	Test set
sgHCM with 40 prototypes	Direct	6.24	10.34
sgFCMed with 40 prototypes	-	4.25	5.43
sgFPCMed with 40 prototypes	-	3.98	4.28
sgPFCMed with 40 prototypes	-	3.56	3.95
sgUPCMed with 40 prototypes	-	3.76	4.08
sgUPFCMed with 40 prototypes	-	**3.07**	**3.06**
Levenshtein distance + FCM (Length of substring=4, tolerance=70%) [32]	Direct	11.40	14.67
FCM+ Levenshtein distance [33]	Direct	24.56	26.78
K-NN with non-linear deformation (P2DHMDM) [55]	Indirect	-	1.9
Product of stumps on Haar features [56]	Indirect	-	3.84
Convolutional neural network with pixel-based feature [57]	Indirect	-	3.08
SVM RBF with gradient-based feature [57]	Indirect	-	2.79
SVM linear with gradient-based feature [57]	Indirect	-	3.34
SVM polynomial with gradient-based feature [57]	Indirect	-	2.79
Extended tangent [58]	Indirect	-	2.2

Similar to the MPEG-7, Copenhagen chromosomes and MNIST data set, our five string grammar fuzzy clustering algorithms on MNIST data set perform better result of classification than sgHCM for all of experiments with vary-parameter m and η. Again, the sgUPFCMed provides the best classification rate as compared to sgFCMed, sgFPCMed, sgPFCMed, sgUPCMed with the reasons as described previously. Again, using only single-prototype to represent each cluster, which may not provide good result because of the limiting of the clustering performance on overlapping data, the multi-

prototypes method is also utilized in this dataset to improve the performance of classification. The sgUPFCMed with 40 prototypes provide the best of classification results. For this data set, 50 prototypes cannot compete with 40 prototypes might be because the objects of the other classes are might be included.

For the indirect comparison, we compare our result with those from the K-NN with non-linear deformation (P2DHMDM) [56], Product of stumps on Haar features [57], Convolutional neural network with pixel-based feature [58], SVM RBF with gradient-based feature [58], SVM linear with gradient-based feature [58], SVM polynomial with gradient-based feature [58], and Extended tangent distance [59]. We can see that the best of our method with 40 prototypes provides better classification accuracy than some methods from the numeric-based algorithms, i.e., the Product of stumps on Haar features [57], Convolutional neural network with pixel-based feature [58], and SVM linear with gradient-based feature [58]. However, does not provide as good classification as some numeric-based algorithms, i.e., P2DHMDM [56], SVM RBF with gradient-based feature [58], SVM polynomial with gradient-based feature [58] and Extended tangent distance [59]. This might be because we implement our algorithm on original data without any pre-processing. However, since our algorithm has the computations on the string, it is easier to transform each prototype digit string back into digit. This cannot happen with those numeric-based algorithms. Again, our methods provide better classification accuracy than string grammar hard clustering methods and decision theoretic methods in literatures for this dataset. However, some misclassifications have occurred in this data set. It might be a result of there are some noisy samples in the dataset as shown in Figure 4.13.

Table 4.26 Cluster validity indices of sgFCMed, sgFPCMed, sgPFCMed, sgUPCMed and sgUPFCMed on USPS data set.

	sgFCMed (m=3)			sgFPCMed (m=3, η=4)			sgPFCMed (m=3, η=4)			sgUPCMed (m=3)			sgUPFCMed (m=3, η=4)		
#prototype of each class	PC	PE	XB	PC	PE	XB	PC	PE	XB	PC	PE	XB	PC	PE	XB
1	0.7966	0.9087	5.0126	0.8567	0.5010	4.4325	0.8618	0.4976	4.4102	0.8620	0.4973	4.4100	0.8626	0.4972	4.4098
5	0.7976	0.9054	5.0123	0.8583	0.4997	4.4256	0.8709	0.4956	4.4075	0.8687	0.4956	4.4087	0.8721	0.4937	4.4078
10	0.7983	0.9021	5.0121	0.8604	0.4997	4.3854	0.8743	0.4914	4.3999	0.8745	0.4913	4.3998	0.8747	0.4911	4.3997
15	0.7985	0.9015	5.0117	0.8678	0.4986	4.3735	0.8757	0.4899	4.3995	0.8809	0.4812	4.3991	0.8795	0.4803	4.3989
20	0.7998	0.8997	5.0116	0.8687	0.4953	4.3690	0.8798	0.4831	4.3989	0.8801	0.4827	4.3985	0.8803	0.4824	4.3983
25	0.8034	0.8967	5.0114	0.8683	0.4896	4.3499	0.8798	0.4803	4.3954	0.8805	0.4809	4.3970	0.8798	0.4716	4.3958
30	0.8214	0.8954	5.0110	0.8697	0.4823	4.3456	0.8801	0.4792	4.3923	0.8828	0.4788	4.3921	0.8727	0.4786	4.3920
35	0.8532	0.8453	5.0104	0.8701	0.4799	4.2684	0.8825	0.4673	4.3897	0.8830	0.4675	4.3914	0.8801	0.4622	4.3899
40	0.8923	0.8345	5.0043	0.8767	0.4734	4.1856	0.8834	0.4589	4.3882	0.8836	0.4588	4.3878	0.8839	0.4584	4.3876
45	0.8643	0.8532	5.0134	0.8684	0.4893	4.2378	0.8711	0.4652	4.3785	0.8699	0.4789	4.3688	0.8693	0.4798	4.3975
50	0.8465	0.8967	5.0112	0.8635	0.4923	4.2394	0.8645	0.4872	4.3723	0.8648	0.4869	4.372	0.8649	0.4867	4.3996

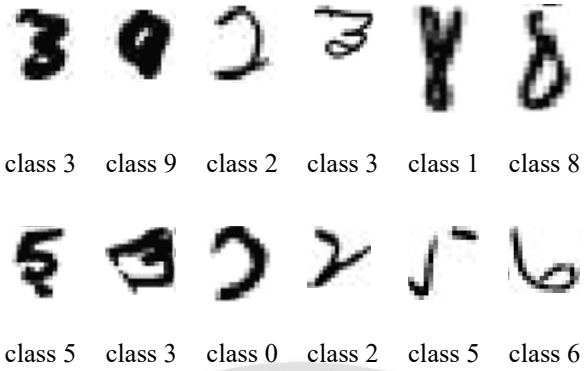


Figure 4.13 Examples of noisy data in USPS dataset

4.2 Conclusion

From the experiments, the result of classification of five string grammar fuzzy clustering algorithms, i.e., sgFCMed, sgFPCMed, sgPFCMed, sgUPCMed and sgUPFCMed with improved fuzzy median string perform better than sgHCM for all experiments. The result of classification from sgFCMed based on the Levenshtein distance may not good for some applications with overlapping data and noisy data with outlier. Hence, we develop the sgFPCMed to solve these problems. Although sgFPCMed is created for solving noisy data, it seems to be worse than sgPFCMed in large dataset. The sgPFCMed can improve the result of clustering in domain of string by relaxing the constraint of sgFPCMed. For improving the classification accuracy result, we proposed sgUPCMed that objective function based on the validity indexes PC and PE from UPCM so that the exponential membership functions are used to describe the degree of belonging. Hence, we can detect the outlier of dataset as well since the membership value tends to be zero where the string is too far from its prototype. However, we found sgUPCMed sometimes generates coincident cluster when we run multi-prototypes clustering. For example, when we run sgUPCMed with *p*-prototypes of each class, after the algorithm converges, it might produce some prototypes with the same location because the algorithm can generate the same memberships values from relaxing constraint $\sum u_{ik} = 1$. Because the columns and rows of the possibilistic values are independent of each other. To solve this problem, we proposed sgUPFCMed that objective function based on the validity indexes PC and PE so that the exponential

membership functions are used to describe the degree of belonging and take advantages of possibility value from PFCMed. Hence, we can detect the outlier of dataset as well and can solve coincident cluster of sgUPCMed. However, using only single-prototype to represent each cluster, which may not adequately model the clusters of arbitrary shape and size and hence limit the clustering performance on overlapping data. Hence, the multiple prototypes of each class technique is utilized in clustering algorithm to improve the performance of classification. For multi-prototypes clustering, choosing the number of prototypes of each class is important. The number of prototypes depends on the distribution, the intra-class variability and inter-class variability of each dataset. We should preserve discrimination between classes for improving the classification accuracy rate. We suggest choosing the number of prototypes of each class large enough that noise in the data is minimized and small enough so the samples of the other classes are not included.

For all of experiments, we utilized the *R-value* to measure the overlapping degree of each data set. The *R-value* is related to the classification accuracy. For example, i.e., the *R-value* of MPEG-7 dataset is higher than MNIST dataset, then the MPEG-7 dataset is considered to have a larger area of overlapping than MNIST dataset, and MNIST dataset yields higher classification accuracy than MPEG-7 dataset. The classification accuracy of four string grammar clustering and value of *R* for four data sets are shown in figure 4.14. We can see that the graph pattern of *R-value* is the opposite of the graph pattern of classification accuracies. This mean that *R- value* is strongly related with the accuracy of our algorithms.

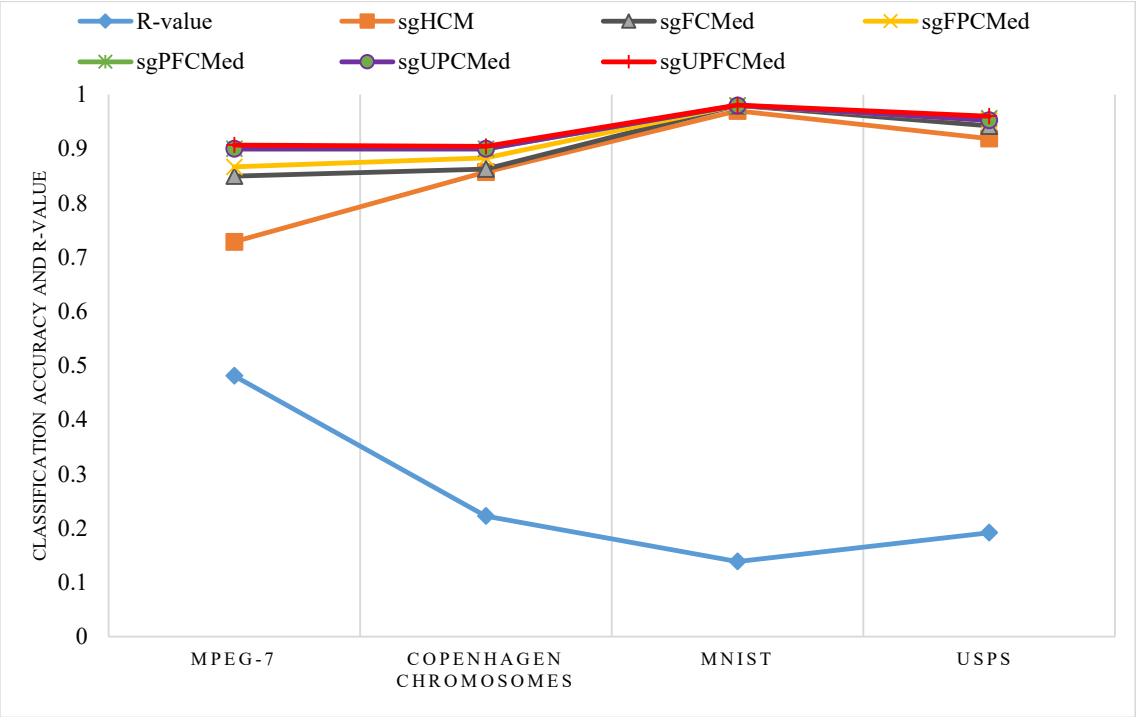


Figure 4.14 Comparison between classification accuracies of four string grammar clustering and *R-value*.

We can conclude advantages and disadvantages of each algorithm below:

1) The sgFCMed can better classify overlapping data than the sgHCM. However, the restriction that the sum of membership values of a data point in all the clusters must be equal to one tends to give high membership values for the outlier points and due to this the algorithm has difficulty in handling outlier points. Hence, sgFCMed may not be good for some applications with large overlapping data and noisy data with outlier.

2) The sgFPCMed is a hybridization of possibilistic theory and sgFCMed and it can enables clustering of noisy data samples i.e. datasets with presence of outliers or noisy points. However, the row sum constraints that may be problematic for large data sets because of the constraint $\sum t_{ik} = 1$.

3) The sgPFCMed provides an improvement to sgFPCMed by eliminating the row sum constraints of sgFPCMed. From the experiments, the sgPFCMed are less sensitive to outliers than sgFCMed and sgFPCMed. However, the accuracy depends on several parameters such as m, η, a, b and γ.

4) The sgUPFCMed has the objective functions based on the validity indexes PC and PE so that the exponential membership functions are used to describe the degree of belonging. Hence, we can detect the outlier of the dataset as well since the membership value tend to be zero where the string is too far from its prototype. However, sgUPCMed sometimes generates coincident when we run multi-prototypes clustering. It might produce some prototypes with the same location because the algorithm can generate the same memberships values from relaxing constraint $\sum u_{ik} = 1$. Hence, sometimes the accuracy rate of sgUPCMed is not good enough.

5) The sgUPFCMed can solve the problems described previously because the objective function of sgUPFCMed based on the validity indexes PC and PE so that the exponential membership functions are used to describe the degree of belonging and take advantages of possibility value from PFCMed. Hence, we can detect the outlier of dataset as well and can solve coincident cluster of sgUPCMed. However, the accuracy depends on several parameters such as m, η, a, and b.

We can see that our methods provide better classification accuracy than string grammar hard clustering methods and some other methods in literatures. Our methods are more appropriate for applications where the structure of a pattern is important than numeric methods. Unfortunately, there are some problems are detected on our algorithms.

1) One problem with our string grammar fuzzy clustering methods is that the median cluster center only suits for symmetric distribution data set, it may not be suitable for largely skewed distribution.

2) Another problem of our system is the problem of Levenshtein distance. The Levenshtein distance takes all transform operations in the same way without taking into account the character that is used in the transform operation. For example, we have three strings ‘aba’, ‘abba’, and ‘abc’ as shown in Figure 4.15. The strings ‘abba’ is in the same cluster as strings ‘aba’ but string ‘abc’ is in the other cluster. However, the distance of string ‘aba’ and string ‘abba’ is equal to 1 and the distance of string ‘aba’ and string ‘abc’ is also equal to 1. They have same values of distance because they use the same transformation operation. This drawback might affect to the result of classification.

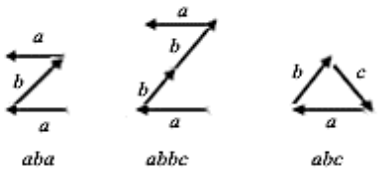


Figure 4.15 Example of three strings ‘aba’, ‘abba’, and ‘abc’.

3) Moreover, we may be need to resize image before creating string sequence because our algorithms may not be invariant to scale. For example, a string representation is shown in Figure 4.16, these are equilateral triangles structures of various sizes. String representation of these triangles are not same.

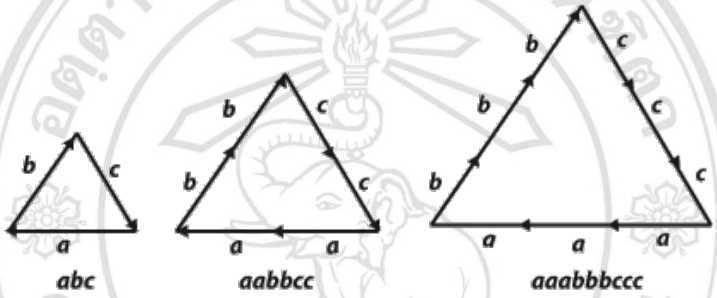


Figure 4.16 String representations of equilateral triangles.

Hence, the above issues should carefully be considered and taken into account for improving the algorithms in the future.