A Novel String Grammar Fuzzy C-Medians

Atcharin Klomsae
Computer Engineering Department,
Faculty of Engineering,
Chiang Mai University,
Chiang Mai, Thailand
atcharin.k@gmail.com

Sansanee Auephanwiriyakul
Senior Member, IEEE
Computer Engineering Department,
Faculty of Engineering,
Chiang Mai University,
Chiang Mai, Thailand
sansanee@ieee.org

Nipon Theera-Umpon
Senior Member, IEEE
Electrical Engineering Department,
Faculty of Engineering,
Chiang Mai University,
Chiang Mai, Thailand
nipon@ieee.org

Abstract—**One of the popular classification problems is the syntactic pattern recognition. A syntactic pattern can be described using string grammar. The string grammar hard C-means is one of the classification algorithms in syntactic pattern recognition. However, it has been proved that fuzzy clustering is better than hard clustering. Hence, in this paper we develop a string grammar fuzzy C-medians algorithm. In particular, the string grammar fuzzy C-medians algorithm is a counterpart of fuzzy C-medians in which a fuzzy median approach is applied for finding fuzzy median string as the center of string data. However, the fuzzy median string may not provide a good clustering result. We then modified a method to compute fuzzy median string with the edition operations (insertion, deletion, and substitution) over each symbol of the string. The fuzzy C-medians with regular fuzzy median and the one with the modified fuzzy median are implemented on 3 real data sets, i.e., Copenhagen chromosomes data set, MNIST database of handwritten digits, and USPS database of handwritten digits. We also compare the results with those from the string grammar hard C-means. The results show that the string grammar fuzzy C-medians is better than the string grammar hard C-means.**

Keywords—fuzzy median; string grammar fuzzy c-medians; Levenshtein distance; syntactic pattern recognition

I. INTRODUCTION

Pattern recognition is generally divided into two general approaches, i.e., a decision-theoretic approach and syntactic approach [1]. In the decision-theoretic approach, a pattern can be characterized by a random feature vector which is numerical description. However, for some applications, structural information is more preferable than quantitative information. Hence, the syntactic pattern recognition approach is needed. In syntactic approach, a large set of complex patterns can be described by sets of simple pattern primitives and grammatical rules. In particular a pattern can be described by a sentence, e.g., a string or a tree or a graph, in a language [2–4]. There are a few syntactic recognition algorithms that is not based on stochastic grammars and languages [1], e.g., the nearest neighbor syntactic recognition rule (sgNN) [2, 3] and the string grammar Hard C-means (sgHCM) [4, 5]. Both algorithms are an extension of the ordinary nearest neighbor and Hard C-means algorithms. The Levenshtein distance [2–5] is used as a dissimilarity measure for both algorithms. There is also a hybrid method [6] using the kernel function based on the Levenshtein distance [2]. Then the syntactic classification is done on those kernels. Although, the classification rate in this case is very good, the validity of the kernel method cannot be established.

In a clustering algorithm, after the clusters are formed, the prototype of each cluster is updated. In particular, in sgHCM, the prototype of each cluster is a median string of the strings in that cluster. The median string of a given cluster is defined as a string that minimizes sum of the distances between each string in the cluster. In general, the problem of searching the median string is an NP-Hard problem [7] and therefore no efficient algorithm can be designed to compute the median string. Hence, improved algorithms in finding the median of cluster are proposed in [8, 9].

In this paper, we introduced a string grammar fuzzy C-medians (sgFCMed). In particular, it is an extension of an ordinary fuzzy C-median [10]. The sgFCMed is utilized the Levenshtein distance [2] as a dissimilarity measure and the fuzzy median [10, 11] is utilized to calculate a cluster prototype. We also follow [9] in improving a fuzzy median -calculation.

II. STRING GRAMMAR FUZZY C-MEDIAN ALGORITHM

The fuzzy C-Medians (FCMED) [10] is one of the popular clustering algorithms. Let $\mathbf{X} = \left\{\vec{x}_j \,\middle|\, j = 1\ldots N\right\}$ be a set of N feature vectors in p-dimensional feature space. Let B = $\left(\vec{c}_1,\ldots,\vec{c}_C\right)$ represent a C-tuple of prototypes each of which characterizes one of the C clusters. The objective function is as follows:

$$J(\mathbf{U},\mathbf{X}) = \sum_{k=1}^{N}\sum_{i=1}^{C} u_{ik}^{m} d_{ik} \qquad (1)$$

with the constraint: $\sum_{i=1}^{C} u_{ji} = 1$ for j = 1 to N, where C and N are the total number of clusters and total number of input vectors, respectively, and $d_{ik} = \left\|\vec{x}_k - \vec{c}_i\right\|_1 = \sum_{j=1}^{p}\left|x_{kj} - c_{ij}\right|$. Here, $u_{ji} \in [0,1]$ is the membership value of data point $\vec{x}_j$ in cluster i.

In equation (1), $m\in$ [1,∞) is called the fuzzifier. The update equation of membership of $\vec{x}_j$ in each cluster i is:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} (\frac{d_{ij}}{d_{kj}})^{1/(m-1)}}. \tag{2}$$

The center $\vec{c}_i$ is calculated by finding a fuzzy median in cluster i with memberships u_{ij}^m using

$$\Psi_{fuzzy}(med_{il}) = \sum_{k=1}^{N} u_{ik} \operatorname{sgn}(x_{kl} - med_{il}) \quad \text{for } l = 1,...,p\ , \tag{3}$$

where u_{ik} is the membership of vector $\vec{x}_k$ in cluster i. The root (m_{il}) of equation (3) is a fuzzy median in l^{th} dimension of cluster i.

Now, we are ready for the string grammar Fuzzy C-Median (sgFCMed). Let $\mathbf{S} = \{s_1,s_2,\ldots,s_N\}$ be a set of N strings. Each string (s_k) is a sequence of symbols (primitives). For example, $s_k = (x_1x_2\ldots x_l)$, a string with length l, where each x_i is a member of a set of defined symbols or primitives ($x_i \in \Sigma$ for $i = 1,...,l$). Let $B = (sc_1, sc_2, ..., sc_C)$ represents a C-tuple of string prototypes each of which characterizes one of the C clusters. Then d_{ij} is computed from the Levenshtein distance [2] between string s_j and string prototypes sc_i ($Lev(sc_i, s_j)$ or Lev_{ij}) (a smallest number of transformations needed to derive one string from the other) between input string j and cluster prototype i). To make the problem simple, we also normalize the distance as

$$Lev(s_j,s_k) = \frac{Lev(s_j,s_k)}{\max(l_j,l_k)}\ , \tag{4}$$

where l_j and l_k are the length of string s_j and s_k, respectively. Then the objective function of sgFCMed is

$$\begin{aligned} &\min \sum_{j=1}^{N}\sum_{i=1}^{C} u_{ij}^m Lev\left(s_j, sc_i\right) \\ &s.t.: \sum_{i=1}^{C} u_{ij} = 1 \quad \text{for } j = 1,\ldots,N \ \text{ and } 0 \le u_{ij} \le 1. \end{aligned} \tag{5}$$

Then the update equation of u_{ij} is

$$u_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{Lev\left(sc_i,s_j\right)}{Lev\left(sc_k,s_j\right)}\right)^{1/(m-1)}}. \tag{6}$$

To compute each string prototype, we need to calculate a fuzzy median string. Since strings in the set S are not numeric vectors, we cannot compute fuzzy median using equation (3). However, in [12], a median string in a set of strings S of cluster i can be calculated as

$$sc_i = \arg\min_{j\in S_i} \sum_{k=1}^{N_i} Lev(s_j,s_k)\ , \tag{7}$$

where S_i and N_i are the set of strings in cluster i and the number of strings in cluster i, respectively. Then we can modify equation (7) to incorporate the idea of fuzzy median by assuming that each string can be a prototype for a particular cluster. Then we find the string that gives the minimum value of summation of Levenshtein distances between that string and other strings in the set with membership value of strings in that cluster. Therefore, the equation to find a fuzzy median string of cluster i corresponding to equations (3) and (7) is as follows:

$$sc_i = \arg\min_{j\in S} \sum_{k=1}^{N} u_{ik} Lev(s_j,s_k)\ , \tag{8}$$

where u_{ik} is the membership value of string k in cluster i. s_j is assumed to be a prototype. The string s_j that gives the minimum value of summation of Levenshtein distances to other strings in cluster j will be selected as a prototype of the cluster.

It has been proved that modified median string will give a better classification rate than the regular median string [9]. Hence, we modified the method in [9] to find fuzzy median. Let $\sum^*$ be the free monoid over the alphabet set $\sum$. Then set of strings $S \subseteq \sum^*$. Then the modified fuzzy median will be

$$sc_i = \arg\min_{j\in\Sigma^*} \sum_{k=1}^{N} u_{ik} Lev(s_j,s_k)\ . \tag{9}$$

This process is an approximation of fuzzy median finding by edition operations (insertion, deletion, and substitution) over each symbol of the string. The selected string (sc_i) will be the one that gives the minimum value. The algorithm of the modified fuzzy median string is as follows:

```
Start with the initial string s. This procedure can be iterated until s is stabilized.
For each position i in the string s
  1. Build alternative
    Substitution: Set z=s. For each symbol $a \in \Sigma$
      a) Set z′ to be the result of substituting i^th symbol with symbol a.
      b) If $\sum_{k=1}^{N} u_{ik} Lev(z',s_k) < \sum_{k=1}^{N} u_{ik} Lev(z,s_k)$, then set z=z′.
    Deletion: Set y to be the result of deleting the i^th symbol of s.
    Insertion: Set x=s. For each symbol $a \in \Sigma$
      a) Set x' to be the result of adding a at position i^th of s.
      b) If $\sum_{k=1}^{N} u_{ik} Lev(x',s_k) < \sum_{k=1}^{N} u_{ik} Lev(x,s_k)$, then set x =x′.
  2. Choose an alternative
    Select string s' from the set of strings {s,x,y,z} (these strings are from step 1) using
      $s' = \arg\min_{m \in \{s,x,y,z\}} \sum_{k=1}^{N} u_{ik} Lev(m,s_k)$.
    Then set s= s'.
```

The sgFCMed algorithm is as follows:

```
Store $n$ unlabeled finite strings $S = \{s_k;\ k = 1,\dots, N\}$
Initial string prototypes for all $C$ classes
Set $m$
Do {
    Compute Levenshtein distance ($Lev_{ij}$) between input string $j$ and cluster prototype $i$
    Update membership using equation (6)
    Update center string of each cluster $i$ ($sc_i$) using equation (8) or (9)
} Until (center strings stabilize)
```

III. Experimental Results

We implemented the sgFCMed with regular fuzzy median and improved fuzzy median on 4 real data sets collected by Prof. Simon M. Lucas and downloaded from http://algoval.essex.ac.uk/data/sequence/. The 3 sets are Copenhagen chromosomes data set, MNIST database of handwritten digits, and USPS database of handwritten digits. We also compared the results on sgHCM with regular median [4] and modified median, i.e., greedy approximate [8] and ME median string [9]. In all of the experiments, a test string was assigned to the class of the nearest prototype. We manually set m=1.5 for Copenhagen chromosome data set. For the other data sets, m was manually set to 2.

A. Copenhagen Chromosomes Data Set

To create strings from the Copenhagen chromosomes data set (Copchron) [13–15], density histograms of chromosome images were calculated. Then histogram of each image was encoded into a string. An example is shown in fig. 1. There are 44 files, each have 100 lines of the form / 5467 119 22 27 9/ AA==a==E===d==A==a=Aa=A=a=b. When 5467 is an identifier and 119 ∈[1,180] is the metaphase the sample coming from. The number 22, 27 and 9 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}. Hence, there are 4400 strings in total from 22 non-sex chromosome type each with 200 strings. It should be noted that we downloaded the encoded data set, not the chromosome images.

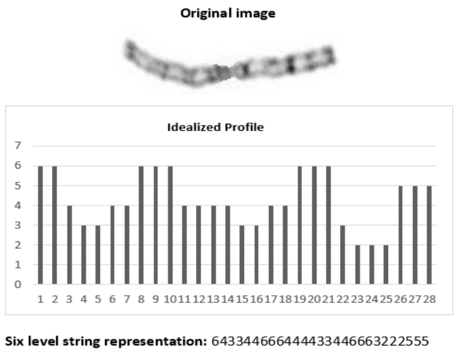

Fig.1. Example from Copenhagen Chromosome dataset

The data set was divided into training and test data sets with 2200 strings in each data set. The classification results on the training and test data sets are shown in tables 1 and 2. We can see that the classification rates on the training data set from the sgFCMed with regular fuzzy median and with improved fuzzy median are 85.77% and 86.28%, respectively. Whereas those from sgHCM with regular median, median from greedy method and median from ME median string are 85.14%, 85.18%, and 85.73%, respectively. The sgFCMed with the regular and improved median on the test data set are 84.05% and 84.91%, respectively. The sgHCM with regular median, median from greedy method, and median from ME median string yield classification rates of 83.5%, 83.18%, and 84%, respectively, on the test data set. Hence the sgFCMed with regular and improved fuzzy median perform better than sgHCM with regular median and improved median in both training and test data sets.

Although, the performance of sgFCMed does not equal to the result from [3], this method will provide the prototype of each non-sex chromosome type. We consider the result to be comparable with those in [3].

B. MNIST Database of Handwritten Digits

The MNIST database of handwritten digits as described in [16] contains 70,000 digits ranging from 0 to 9. This data set was divided into 60,000 training samples and 10,000 test samples. Table 3 shows the number of samples for each digit in both data sets. Some examples are shown in fig. 2. Again, we acquired the encoded data set from the website, not the original images.

To reduce the training time, we randomly selected only 1,000 samples from each digit. Hence, there were only 10,000 samples randomly selected to be our training data. In this case, we will have the result from training data and blind test from test set. The classification rates on training data and blind test from test set of the sgHCM with regular and improved median and the sgFCMed with regular fuzzy median and improved fuzzy median are shown in tables 4 and 5.

TABLE I. Classification error rate (%) for training set of Copenhagen chromosomes data set

sgHCM			sgFCMed	
Regular median	Greedy method [8]	ME median string [9]	Regular fuzzy median	Improved fuzzy median
14.86	14.82	14.27	14.23	13.72

TABLE II. Classification error rate (%) for test set of Copenhagen chromosomes data set

sgHCM			sgFCMed	
Regular median	Greedy method [8]	ME median string [9]	Regular fuzzy median	Improved fuzzy median
16.5	16.82	16.00	15.95	15.09

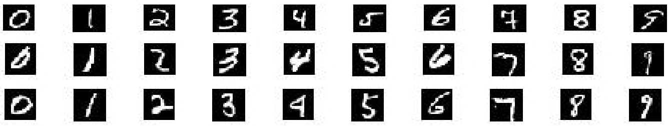

Fig.2. Example from MNIST dataset

TABLE III. NUMBER OF SAMPLES IN EACH DIGIT IN TRAINING AND TEST SETS OF MNIST DATABASE OF HANDWRITTEN DIGITS

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

TABLE IV. CLASSIFICATION ERROR RATE (%) FOR TRAINING SET OF MNIST HANDWRITTEN DATA

sgHCM			sgFCMed	
Regular median	Greedy method [8]	ME median string [9]	Regular fuzzy median	Improved fuzzy median
5.82	5.80	5.84	4.87	4.19

TABLE V. CLASSIFICATION ERROR RATE (%) FOR TEST SET OF MNIST HANDWRITTEN DATA

sgHCM			sgFCMed	
Regular median	Greedy method [8]	ME median string [9]	Regular fuzzy median	Improved fuzzy median
8.96	7.86	7.85	5.86	5.28

The sgFCMed with regular and improved fuzzy median gives 95.13% and 95.81% correct classification rate on training data. Also, the correct classification rates on blind test from test set from sgFCMed are 94.14% and 94.72%. Whereas the sgHCM with regular median, median from greedy method and ME median string provide 94.18%, 94.2% and 94.16% correct classification on the training data. For the blind test data from test set, the correct classification rates from the sgHCM with regular median, median from greedy method and ME median string are 91.04%, 92.14% and 92.15%, respectively. Again, the sgFCMed performs better than sgHCM in both training and test data sets. For this data set, the performance of string grammar clustering is better than the classification from numeric vectors [16]. However, there are several numeric method trained on 60,000 samples reported in [17] that the best correct classification rate is around 99%. We still consider our result is comparable with those numeric methods since, we only use 10,000 samples out of 60,000 samples to train the sgFCMed. The string grammar clustering can give the prototypes that can be transformed back into digits, as well.

C. *USPS Database of Handwritten Digits*

Examples of the USPS handwritten digit data set described in [16] are shown in fig. 3. Again, the encoded strings were downloaded from the website. There are 9,298 strings for digit numbers 0 to 9.

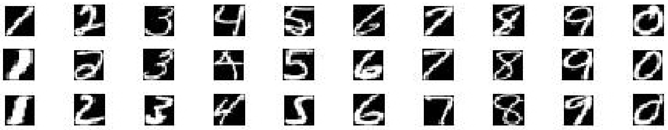

Fig.3. Example of USPS dataset

In this case, the data set was divided into 7,291 training strings and 2,007 test strings. The numbers of samples of all digit classes in both data sets are shown in table 6.

The classification results on training and test data sets from the sgHCM with regular median and improved median and sgFCMed with regular fuzzy median and improved fuzzy median are shown in tables 7 and 8. We can see that the training correct classification rates from the sgFCMed with regular and improved fuzzy median are 93.92% and 94.23%, and the testing correct classification rates from those sgFCMed are 93.87% and 92.53%. The sgHCM with regular median, median from greedy method and ME median string gives the training correct classification rate of 91.59%, 91.88% and 91.91%, whereas, the correct classification from the sgHCM with regular median, median from greedy method and ME median string are 89.69%, 87.9% and 87.9%, respectively. Again, the sgFCMed performs better than the sgHCM in both training and test data sets. Again, the performance of string grammar clustering is better than the classification from numeric vectors [16]. But the best performance of numeric method reported in [17] is around 98%. Although, the sgFCMed cannot compete with the best numeric method, the sgFCMed can give the prototypes that can be transformed back into digits, when those numeric methods cannot.

TABLE VI. NUMBER OF SAMPLES IN EACH DIGIT IN TRAINING AND TEST SETS OF USPS DATABASED OF HANDWRITTEN DIGITS

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

TABLE VII. CLASSIFICATION ERROR RATE (%) FOR THE TRAINING SET OF USPS HANDWRITTEN DATA SET

sgHCM			sgFCMed	
Regular median	Greedy method [8]	ME median string [9]	Regular fuzzy median	Improved fuzzy median
8.41	8.12	8.09	6.08	5.77

TABLE VIII. CLASSIFICATION ERROR RATE (%) FOR TEST SET OF USPS HANDWRITTEN DATA SET

sgHCM			sgFCMed	
Regular median	Greedy method [8]	ME median string [9]	Regular fuzzy median	Improved fuzzy median
10.31	12.10	12.10	6.13	7.37

IV. CONCLUSION

One approach of pattern recognition is the syntactic approach. There are a few syntactic algorithms developed so far. The syntactic clustering algorithm is one of them. However, the syntactic clustering algorithm so far is the hard clustering. Hence, in this paper, we developed a string grammar fuzzy C-medians (sgFCMed) with regular fuzzy median and improved fuzzy median. We implemented our algorithm on 3 real data sets, i.e., Copenhagen chromosomes data set, MNIST database of handwritten digits, and USPS database of handwritten digits. We compared our results with those from the string grammar Hard C-Means (sgHCM) with regular median, median computed from greedy method, and ME median string, as well. From the results, we found that the sgFCMed is better than the sgHCM.

ACKNOWLEDGMENTS

The authors would like to thank Thailand Research Fund under the Royal Golden Jubilee Ph.D. Program (Grant No. PHD/0044/2555) for financial support. We would also like to thank Prof. Simon M. Lucas for the sequence datasets.

REFERENCES

[1] R. C. Gonzalez and M. G. Thomason, *Syntactic Pattern Recognition An Introduction*, Addison Wesley Publishing Company, 1978, pp.12-13.

[2] K. S. Fu and S. Y. Lu, “A clustering procedure for syntactic patterns,” *1EEE Trans. Syst. Man Cybern*, vol. 7, pp. 734-742, October 1977.

[3] A. Juan and E. Vidal, “On the use of Normalized Edit Distances and an Efficient *k*-NN Search Technique (*k*-AESA) for Fast and Accurate String Classification,” *2000 Proceedings of 15th International Conference on Pattern Recognition*, Barcelona, pp. 676 - 679, 2000.

[4] J. C. Bezdek, J. Keller, R. Krishnapuram and N. R. Pal, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*, Kluwer Academic Publishers, USA., 1999

[5] [7] K.S. Fu, *Syntactic Pattern Recognition and Applications*, Prentice-Hall, 1982.

[6] [13] M. Neuhaus and H. Bunke ,“Edit distance based kernel functions for structural pattern classification”, *Pattern Recognition*, vol.39, pp. 1852 – 1863, 2006.

[7] [3] C. de la Higuera and F.Casacuberta, “The topology of strings: two np-complete problems,” *Theoretical Computer Science*, vol. 230, pp.39-48, 2000.

[8] [4] F.Kruzslicz, “Improved Greedy Algorithm for Computing Approximate Median Strings,” *Acta Cybern*, pp.331-339, 1999.

[9] [5] C.D. Martinez, A.Juan, and F. Casacuberta, “Use of median string for classification,” *Proc.15th Int. Conf. on Patt. Rec.*, vol.2, pp.903 - 906, September 2000.

[10] [11] P.R. Kersten, “Fuzzy order statistics and their application to fuzzy clustering,” *IEEE Trans. Fuzzy Syst.*, vol.7, pp. 708 – 712, December 1999.

[11] [6] P.R. Kersten, “The fuzzy median and fuzzy mad,” *Proc. ISUMA/NAFIPS*, College Park MD., PP.85-88, September 1995.

[12] [8] T. Kohonen, “Median strings,” *Pattern Recognition Letters*, vol.3, 1985.

[13] C. Lundsteen, J. Phillip, and E. Granum, “Quantitative andlysis of 6985 digitized trysin {G}-banded human metaphase chromosomes,” *Clinical Genetics*, vol 18, pp. 355 – 370, 1980.

[14] E Granum, M G Thomason and J Gregor, “On the use of automatically inferred {M}arkov networks for chromosome analysis,” in C. Lundsteen and J. Piper (Eds), *Automation of Cytogenetics*, pp. 233 – 251, 1989.

[15] E Granum and M G Thomason, “Automatically inferred {M}arkov network models for classification of chromosomal band pattern structures”, *Cytometry*, vol. 11, pp. 26 – 39, 1990.

[16] [14] C.D. Wang, J.H. Lai, C.Y. Suen, J.Y. Zhu, "Multi-Exemplar Affinity Propagation," IEEE Trans. on Pattern Analysis & Machine Intelligence, vol.35, no. 9, pp. 2223-2237, Sept. 2013.

[17] D. Keysers, T. Deselaers, C. Gollan, and H. Ney, “Deformation Models for Image recognition,” *IEEE Transactions on Pattern Anakysis and Machine Intelligence*, vol. 29, no. 8, pp. 1422 -1435, 2007.