ELSEVIER

Pattern Recognition Letters 19 (1998) 1285–1291

Pattern Recognition Letters

On-line hierarchical clustering

Yasser El-Sonbaty [a,*], M.A. Ismail [b]

[a] Department of Computer Engineering, Arab Academy for Science and Tech., Alexandria 1029, Egypt
[b] Department of Computer Science, Faculty of Engineering, Alexandria 21544, Egypt

Received 19 September 1997; received in revised form 3 August 1998

Abstract

Most of the techniques used in the literature for hierarchical clustering are based on off-line operation. The main contribution of this paper is to propose a new algorithm for on-line hierarchical clustering by finding the nearest k objects to each introduced object so far and these nearest k objects are continuously updated by the arrival of a new object. By final object, we have the objects and their nearest k objects which are sorted to produce the hierarchical dendogram. The results of the application of the new algorithm on real and synthetic data and also using simulation experiments, show that the new technique is quite efficient and, in many respects, superior to traditional off-line hierarchical methods. © 1998 Published by Elsevier Science B.V. All rights reserved.

Keywords: Hierarchical techniques; Numeric objects; On-line clustering; Agglomerative methods

1. Introduction

The objective of cluster analysis is to group a set of objects into clusters such that objects within the same cluster have a high degree of similarity, while objects belonging to different clusters have a high degree of dissimilarity.

The clustering of data set into subsets can be divided into hierarchical and non hierarchical or partitioning methods. The general rationale behind partitioning methods is to choose some initial partitioning of the data set and then alter cluster memberships so as to obtain better partitions according to a predefined objective function.

Hierarchical clustering procedures can be divided into agglomerative methods, which progressively merge the objects according to some distance measure in such a way that whenever two objects belong to the same cluster at some level, they remain together at all higher levels, and divisive methods, which progressively subdivide the data set (Gowda and Krishna, 1978).

Objects to be clustered usually come from an experimental study of some phenomenon and are described by a specific set of features selected by the data analyst. The feature values may be measured on different scale and these can be: *Continuous numeric*, *Symbolic*, or *Structured*.

Continuous numeric data are well known as a classical data type and many algorithms for clustering this type of data using partitioning or hierarchical techniques can be found in the literature (Jain and Dubes, 1988). Symbolic objects are extension of classical data types. In conventional

[*] Corresponding author. E-mail: aastf045@aast.egnet.net

PII: S0167-8655(98)00104-4

data sets, the objects are individualized, whereas in symbolic objects, they are more unified by means of relationship. Based on the complexity, the symbolic objects can be of Assertion, Hoard or Synthetic type (Gowda and Ravi, 1995). Some references to clustering of symbolic objects can be found in Diday, 1988; Gowda and Diday, 1991; Gowda and Diday, 1992; Gowda and Ravi, 1995; Fisher, 1987; Cheng and Fu, 1985; Michalski and Stepp, 1983; Ichino, 1988; Gennari et al., 1989; Ralambondrainy, 1995 using different methodologies like hierarchical clustering (Gowda and Diday, 1991; Gowda and Diday, 1992; Gowda and Ravi, 1995; Michalski and Stepp, 1983; Ichino, 1988), incremental clustering (Gennari et al., 1989), partitioning clustering (Ralambondrainy, 1995), and recently, fuzzy clustering (El-Sonbaty and Ismail, 1998). In the literature, the researches dealing with symbolic objects are less than those for numerical objects and this is due to the nature of such objects which is simple in construction but hard in processing. Besides, the values taken by the features of symbolic objects may include one or more elementary objects and, the data set may have a variable number of features (Gowda and Diday, 1991). Structured objects have higher complexity than continuous and symbolic objects because of their structure which is much more complex and their representation which needs higher data structures to permit the description of relations between elementary object components and facilitate hierarchical object models that describe how an object is built up from the primitives. A survey of different representations and proximity measures of structured objects can be found in El-Sonbaty et al., submitted.

Most of the hierarchical techniques introduced for clustering numeric or symbolic objects were off-line and that means these techniques require all the objects or the distance matrix to be available before the start of any hierarchical clustering routine which seems impractical in some cases. The drawbacks in using hierarchical techniques are well known in the field of data clustering. Memory size, updating the membership matrix, complexity per iteration of calculating distance function, and overall complexity of the algorithm to name a few of these difficulties faced when using any hierarchical based technique (Jain and Dubes, 1988).

The main contribution of this paper is to introduce an on-line agglomerative hierarchical technique based on the concept of single-linkage method for clustering symbolic and numeric data. The new algorithm has computational complexity $O(n^2)$ which is lower than the computational complexity of traditional hierarchical techniques reported in the literature (Jain and Dubes, 1988; Diday, 1988; Gowda and Diday, 1991; Gowda and Diday, 1992; Gowda and Ravi, 1995; Fisher, 1987; Cheng and Fu, 1985; Michalski and Stepp, 1983; Ichino, 1988; Gennari et al., 1989) that have $O(n^3)$. The proposed algorithm has also lower memory size that facilitates dealing with large data sets.

Section 2 describes the proposed algorithm. Applications and analysis of experimental results are shown in Sections 3 and 4.

2. Proposed algorithm

In this section, we introduce the new algorithm and discuss its computational complexity and required memory size.

2.1. On-line hierarchical algorithm

In on-line operation, the objects are introduced to the algorithm one by one. At each step, the new object updates the membership matrix to improve the results obtained so far. By last object, the algorithm must be ready to generate the final hierarchical dendogram.

We use very simple idea and also data structure to reduce the computational complexity and memory size for the proposed algorithm. For each object, we calculate the nearest k objects to it sorted by similarity/dissimilarity measure from closest to furthest. These nearest k objects are continuously updated by the arrival of a new object. The nearest k objects serve to follow the concept of single-linkage strategy that the proposed algorithm is based on.

By final object, we already have a group of objects and their nearest k objects, which are stored in ascending order according to their dissimilarity or in descending order when using similarity measure to generate a set of pairs. From the list of sorted pairs the hierarchical dendogram can be visualized by iteratively merging the objects contained at the same pair starting from the first pair that has the minimum distance and continuing the merge until all objects are covered.

The proposed algorithm is sensitive to the selection of the dissimilarity measure used for calculating the distance between the objects as different distance measures can generate different hierarchical dendograms for the same data set.

The recommended values of k that give almost zero error are demonstrated through experimental results.

The proposed algorithm can be summarized in the following steps:

```
[1]: Read first object.
[2]: Set the first nearest object to the same object number.
[3]: For $i = 2$ to number of objects
    [4]: For $j = 1$ to $i - 1$
        [5]: If $j$th object is closer to $i$th object than $p$th nearest object; $p = I{\cdot}{\cdot}k$
            then $d$th nearest $= d - 1$th nearest; $d = k{\cdot}{\cdot}p + 1$
                $p$th nearest $= j$th object
    [6]: Get nearest $k$ objects to $i$th object
[7]: Sort the table of nearest objects in ascending order
[8]: Draw the hierarchical dendogram
```

Illustrative example: Assume we would like to generate the hierarchical clustering of the microcomputer data shown in Fig. 1 using the similarity distance published in Gowda and Diday, 1992.

The operation sequences of the algorithm are shown through steps (*a*) to (*l*) for $k = 2$. From table (*l*), the hierarchical clustering can be easily generated by sorting the contents of this table in ascending order. The following ordered pairs were obtained from the sorting operation: (5, 4), (10, 9), (1, 0), (9, 8), (10, 3), (11, 5), (0, 9), (11, 4), (3, 1), (7, 5), (8, 10), (2, 8), (7, 3), (2, 1), (6, 2).

object #	1st nearest	2nd nearest
0	0	-

(a)

object #	1st nearest	2nd nearest
0	1	-
1	0	-

(b)

object #	1st nearest	2nd nearest
0	1	2
1	0	2
2	1	0

(c)

object #	1st nearest	2nd nearest
0	1	2
1	0	3
2	1	0
3	1	0

(d)

object #	1st nearest	2nd nearest
0	1	2
1	0	3
2	1	0
3	1	0
4	3	2

(e)

object #	1st nearest	2nd nearest
0	1	2
1	0	3
2	1	0
3	1	0
4	5	3
5	4	3

(f)

object #	1st nearest	2nd nearest
0	1	2
1	0	3
2	1	0
3	1	0
4	5	3
5	4	3
6	2	1

(g)

object #	1st nearest	2nd nearest
0	1	2
1	0	3
2	1	0
3	1	7
4	5	7
5	4	7
6	2	1
7	5	3

(h)

object #	1st nearest	2nd nearest
0	1	2
1	0	3
2	8	1
3	1	7
4	5	7
5	4	7
6	2	8
7	5	3
8	2	0

(i)

object #	1st nearest	2nd nearest
0	1	9
1	0	3
2	8	1
3	1	9
4	5	7
5	4	7
6	2	8
7	5	3
8	9	2
9	8	0

(j)

object #	1st nearest	2nd nearest
0	1	9
1	0	3
2	8	1
3	10	1
4	5	7
5	4	7
6	2	8
7	5	3
8	9	10
9	10	8
10	9	3

(k)

object #	1st nearest	2nd nearest
0	1	9
1	0	3
2	8	1
3	10	1
4	5	11
5	4	11
6	2	11
7	5	3
8	9	10
9	10	8
10	9	3
11	5	4

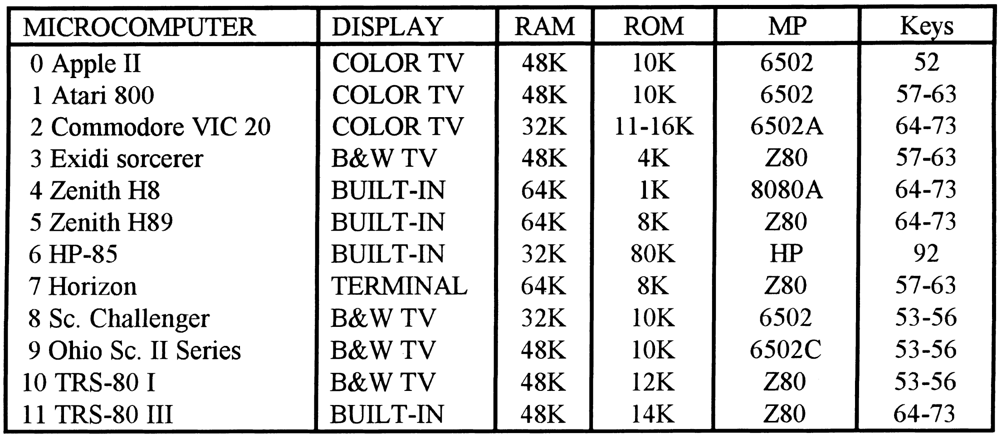
(l)

The equivalent hierarchical dendogram is shown in Fig. 2.

The dendogram shown in Fig. 2 is identical to that reported in Gowda and Diday, 1992 using off-line technique. If we cut the dendogram at an appropriate level, as shown in Fig. 2 by dotted line, we obtain the following classification: {0, 1, 3, 8, 9, 10}, {4, 5, 7, 11}, {2}, {6}. This result is the same as that published in Gowda and Diday, 1992.

2.2. Analysis of computational complexity and memory size

It is clear that the computational complexity of steps [3]–[6] is $O(kn^2)$, where n is the number of objects and k is the number of nearest objects utilized during the algorithm. The sorting done in



MICROCOMPUTER	DISPLAY	RAM	ROM	MP	Keys
0 Apple II	COLOR TV	48K	10K	6502	52
1 Atari 800	COLOR TV	48K	10K	6502	57-63
2 Commodore VIC 20	COLOR TV	32K	11-16K	6502A	64-73
3 Exidi sorcerer	B&W TV	48K	4K	Z80	57-63
4 Zenith H8	BUILT-IN	64K	1K	8080A	64-73
5 Zenith H89	BUILT-IN	64K	8K	Z80	64-73
6 HP-85	BUILT-IN	32K	80K	HP	92
7 Horizon	TERMINAL	64K	8K	Z80	57-63
8 Sc. Challenger	B&W TV	32K	10K	6502	53-56
9 Ohio Sc. II Series	B&W TV	48K	10K	6502C	53-56
10 TRS-80 I	B&W TV	48K	12K	Z80	53-56
11 TRS-80 III	BUILT-IN	48K	14K	Z80	64-73

Fig. 1. Microcomputer data.

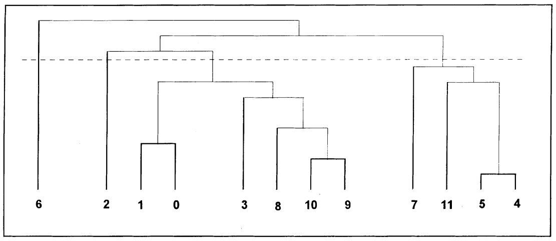


Fig. 2. Hierarchical clustering of Fig. 1.

step [7] can take O(n log n) as an average computational complexity. In general, the computational complexity of the proposed algorithm is O(kn^2). For number of objects $n \gg k$, the average computational complexity of the new algorithm can be calculated as O(n^2). The computational complexity of traditional hierarchical techniques is O(n^3) (Jain and Dubes, 1988).

Regarding the memory size required for the proposed algorithm, it needs exactly k^*n memory cells as it only stores for each object, the nearest k objects so, there is no need to keep the distance between all individuals. Meanwhile, other hierarchical techniques need at least $(0.5(n^2 - n))$ cells assuming the lower triangle of the distance matrix is only memorized.

It is obvious that the new algorithm has lower computational complexity and memory size than other hierarchical techniques. Besides, it can manipulate on-line operations.

3. Experimental results

In this section the performance of the proposed algorithm is tested and evaluated using some test data reported in the literature and some simulation experiments. The data sets used in these experiments are synthetic or real data and their classification is known from other clustering techniques (Gowda and Diday, 1991; Gowda and Diday, 1992; Gowda and Ravi, 1995; Ichino, 1988). Comparisons between results obtained from the proposed algorithm and other techniques are given. The simulation experiments are used here to demonstrate the performance of the new algorithm.

Experiment 1: The test data shown in Fig. 1 was introduced to the new algorithm and the hierarchical dendograms of the final results were sketched using different distance measures reported in (Gowda and Diday, 1991; Gowda and Diday, 1992; Gowda and Ravi, 1995; Ichino, 1988). It was found that exact results were obtained from the proposed algorithm as mentioned in (Gowda and Diday, 1991; Gowda and Diday, 1992; Gowda and Ravi, 1995; Ichino, 1988) using $k = 2$.

Experiment 2: The fat–oil data shown in Fig. 3 was introduced to the proposed algorithm. Fig. 4 shows the dendogram using the similarity measure reported in Gowda and Diday, 1992 and by setting $k = 2$. If we cut the dendogram at an appropriate

No.	Sample name	Gravity (g/cm^3)	Freezing point	io. value	sa. value	m.f. acids
0	Linseed oil	0.930-0.935	-27 to -8	170-204	118-196	L,Ln,O,P,M
1	Perilla oil	0.930-0.937	-5 to -4	192-208	188-197	L,Ln,O,P,S
2	Cotton-seed	0.916-0.918	-6 to -1	99-113	189-198	L,O,P,M,S
3	Seaame oil	0.920-0.926	-6 to -4	104-116	187-193	L,O,P,S,A
4	Camellia	0.916-0.917	-21 to -15	80-82	189-193	L,O
5	Olive-oil	0.914-0.919	0 to 6	79-90	187-196	L,O,P,S
6	beef-tallow	0.860-0.870	30 to 38	40-48	190-199	O,P,M,S,C
7	Lard	0.858-0.864	22 to 32	53-77	190-202	L,O,P,M,S,Lu

Fig. 3. Fat–Oil data.

level, as shown in Fig. 4 by dotted line, we get two classes, {0, 2, 3, 4, 5,}, {6, 7}. This result is the same as those for Gowda and Diday, 1992; Gowda and Ravi, 1995; Ichino, 1988. If the dendogram is cut at an appropriate level to give three or more clusters, the result obtained from the proposed algorithm is identical as that reported in Gowda and Diday, 1992 and is different from those in Gowda and Diday, 1991; Gowda and Ravi, 1995; Ichino, 1988 due to the difference in dissimilarity measure used in these papers. This experiment is repeated using the metric introduced by Ichino (1988) and exact results are obtained for three and more clusters.

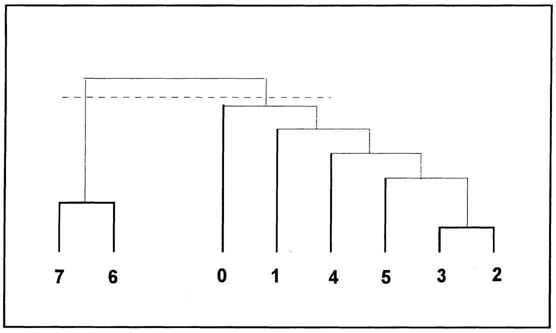


Fig. 4. Hierarchical clustering of Fig. 3.

Experiment 3: The data of this experiment is taken from Botany (Gowda and Diday, 1992; Gowda and Ravi, 1995) and shown in Fig. 5. The dendogram obtained from applying the proposed algorithm on this experiment using the similarity measure of Gowda and Diday, 1992 is shown in Fig. 6. If the dendogram is cut at an appropriate level; as shown by dotted line; we get three class {0, 1, 2}, {3, 4, 5}, {6, 7, 8}. This result is identical to that reported in Gowda and Diday, 1992; Gowda and Ravi, 1995.

Experiment 4: In this experiment, we use simulation to test the behavior of the proposed algorithm in handling small data sets ranging from 10 to 100 objects. The algorithm was tested to check the suitable value for k that gave the lowest percentage of error. The error was calculated as the percentage of number of missing pairs not generated by the proposed algorithm in comparison with the original hierarchical technique that caused the merging of some clusters. The error was calculated as the average over 100 trials. The

Class 1 (Annonaceae)	Class 2 (Caesalpiniaceae)	Class 3 (Clusiaceae)
0. degjknpCFR	3. bfgilmpBILPR	6. aehilmoswxzDFR
1. dfhjknpCFR	4. aehjlmpBIKPR	7. cfgilmoswxzDFR
2. dehjknpCFR	5. afhikmpDIKNR	8. cegjlmoswxzDFR

Fig. 5. Data of Experiment 3.

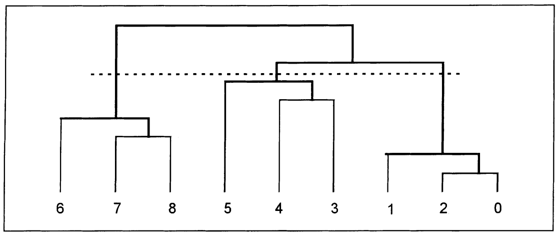


Fig. 6. Hierarchical dendogram of Experiment 3.

objects were represented by only one numeric features in the range of [1..100] as increasing the number of features has no effect on the behavior of the algorithm but increases the processing time. Fig. 7 shows the results obtained from this experiment.

From Fig. 7 it is clear that for small data sets, the best value for k is 3 as starting from $k = 3$, the percentage of error obtained was very low and was almost zero.

Experiment 5: The same simulation as experiment 4 was performed here but for large data sets ranging from 200 to 1000. Fig. 8 shows the results obtained for this experiment.

From Fig. 8. it can be concluded that for large data sets, k can be chosen in the range k 7, as we got very low percentage of error starting from $k = 7$. To clarify the power of suggested algorithm, assume we deal with 1000 objects. The memory required by proposed algorithm is $8 \times 1000 = 8000$ memory cells assuming $k = 8$, while the memory cells required by any traditional hierarchical technique is at least 499,500 which is much higher than the memory required by the new algorithm.

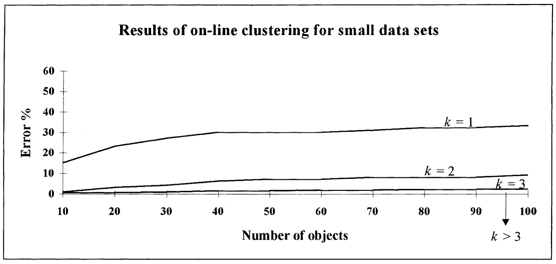


Fig. 7. Results of Experiment 4.

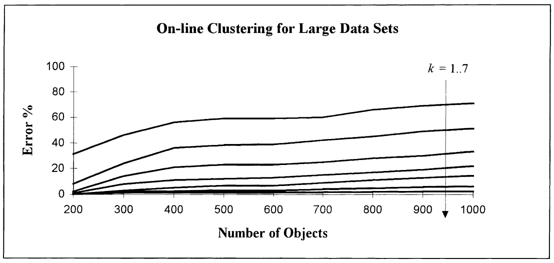


Fig. 8. Results of Experiment 5.

4. Discussions and conclusions

In this paper, a new on-line algorithm for hierarchical clustering based on the concept of single-linkage method was introduced. For each object, we calculate the nearest k objects to it. These nearest k objects are continuously updated by the arrival of a new object. By final object, we already have a group of objects and their nearest k objects, which are sorted to generate a set of pairs constructing the hierarchical dendogram. From experimental results and complexity analysis, the following points can be concluded:

1. The proposed algorithm can be used efficiently for on-line hierarchical clustering.
2. The algorithm can handle objects with numeric or symbolic features.
3. The computational complexity of the new algorithm is $O(kn^2)$. For $n \gg k$, this complexity can be reduced to $O(n^2)$. The complexity is much lower than those of conventional off-line methods that have $O(n^3)$ [2].
4. The memory required for introduced algorithm is only $k \times n$ cells which is much lower than $(0.5(n^2 - n))$ needed by other traditional off-line techniques.
5. For small data sets ranging from 10 to 100 objects, it was found that recommended value for k is k 3.
6. For large data sets ranging from 200 to 1000 objects, the suitable range for k is k 7.

References

Cheng, Y., Fu, K.S., 1985. Conceptual clustering in knowledge organization. PAMI 7, 592–598.

Diday, E., 1988. In: Bock, H.H. (Ed.), The Symbolic Approach in Clustering, Classification and Related Methods of Data Analysis. Elsevier, Amsterdam.

El-Sonbaty, Y., Ismail, M.A., 1998. Fuzzy clustering for symbolic data. IEEE Trans. on Fuzzy Systems 6 (2), 195–204.

El-Sonbaty, Y., Kamel, M.S., Ismail, M.A., submitted. Representations and proximity measures of structured features.

Fisher, D.H., 1987. Knowledge acquistion via incremental conceptual clustering. Machine Learning 2, 103–138.

Gennari, J., Langley, P., Fisher, D., 1989. Models of incremental concept formation. Artificial Intelligence 40, 11–62.

Gowda, K.C., Diday, E., 1991. Symbolic clustering using a new dissimilarity measure. Pattern Recognition 24 (6), 567–578.

Gowda, K.C., Diday, E., 1992. Symbolic clustering using a new similarity measure. IEEE Trans. Sys. Man Cybern. 22 (2), 368–378.

Gowda, K.C., Krishna, G., 1978. Dissaggregative clustering using the concept of mutual nearest neighbourhood. IEEE Trans. Sys. Man Cybern. 8, 883–895.

Gowda, K.C., Ravi, T.V., 1995. Divisive clustering of symbolic objects using the concepts of both similarity and dissimilarity. Pattern Recognition 28 (8), 1277–1282.

Ichino, M., 1988. General metrics for mixed features-the Cartesian space theory for pattern recognition, Proc. IEEE 1988 Int. Conf. Syst. Man Cybern.

Jain, A.K., Dubes, R.C., 1988. Algorithms for Clustering Data. Printice Hall, Englewood Cliffs, NJ.

Michalski, R., Stepp, R.E., 1983. Automated construction of classifications: Conceptual clustering versus numerical taxonomy. PAMI 5, 396–410.

Ralambondrainy, H., 1995. A conceptual version of the K-means algorithm. Pattern Recognition Letters 16, 1147–1157.