STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume L, Number 2, 2005

UNSUPERVISED SINGLE-LINK HIERARCHICAL CLUSTERING

DANA AVRAM LUPŞA

Abstract. There are many clustering techniques presented in the literature. The particularity of single-link clustering is that it rather discovers the clusters as chains. We aim to identify a method to apply the single link clustering technique so that: it discovers the first level clusters and the user doesn't have to provide any sort of a parameter. We focuses on clusters that are well separated, and so, which have to maximize the intra-cluster similarity and minimize the inter-cluster similarity. We evaluate the method on a two dimensional space, that is planar points.

1. Introduction

In the literature, a vast collection of clustering algorithm [6], [2] is available. There is no clustering technique that is universally applicable in uncovering the variety of structures present in multidimensional data sets. Studies of clustering were made from a long time ([4], 1987), but they also constitute recent preocupations of reserchers ([7],2002). A list of materials about detecting clusters and the number of clusters can be found in [10].

All clustering algorithms will, when presented with data, produce clusters – regardless of whether the data contain clusters or not. If the data does contain clusters, some clustering algorithms may obtain better results than others. We focus on data sets that do contain clusters. More than that, we will also request the data to be relatively uniform distributed inside the clusters.

In the literature, the classification is a method that assigns objects to predefined groups; it is a sort of supervised learning technique [5]. Clustering infers groups based on inter-object similarity; it tends to be an unsupervised learning technique. But clustering techniques needs a semi-supervised parameter - that is the number of clusters, and/or the error or/and a maximum number of steps to be executed. In this paper we suggest a method applicable to hierarchical clustering that do not need any parameter. By using single-link hierarchical clustering, the method

Received by the editors: June 27, 2005.

2000 *Mathematics Subject Classification.* 65-05, 65S05.

1998 *CR Categories and Descriptors.* I.5.2. [**Computing Methodologies**]: PATTERN RECOGNITION – *Design Methodology – Classifier design and evaluation*; I.5.3. [**Computing Methodologies**]: PATTERN RECOGNITION – *Clustering – Algorithms* .

we suggest discover the clusters in which the data is grouped, if there are such clusters and they are well identified.

2. Hierarchical Clustering

The hierarchical clustering algorithm was first defined by S.C. Johnson in *Hierarchical Clustering Schemes* , Psychometrika 1967. Given a set of N items to be clustered, and a $N * N$ distance (or similarity) matrix, the basic process of hierarchical clustering is this:

Step 1: Start by assigning each item to a cluster, so that if you have N items, you now have N clusters, each containing just one item. Let the distances (similarities) between the clusters be the same as the distances (similarities) between the items they contain.

Step 2: Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster less.

Step 3: Compute distances (similarities) between the new cluster and each of the old clusters.

Step 4: Repeat steps 2 and 3 until all items are clustered into a single cluster of size N.

Step 3 can be done in several ways, which is what distinguishes *single-linkage* from *complete-linkage* and *average-linkage* clustering.

In *single-linkage* clustering (also called the *connectedness* or *minimum* method), we consider the distance between one cluster and another cluster to be equal to the *shortest* distance from any member of one cluster to any member of the other cluster. If the data consist of similarities, we consider the similarity between one cluster and another cluster to be equal to the *greatest* similarity from any member of one cluster to any member of the other cluster.

In *complete-linkage* clustering (also called the *diameter* or *maximum* method), we consider the distance between one cluster and another cluster to be equal to the *greatest* distance from any member of one cluster to any member of the other cluster. In *average-linkage* clustering, we consider the distance between one cluster and another cluster to be equal to the *average* distance from any member of one cluster to any member of the other cluster.

The complete-link algorithm produces tightly bound or compact clusters . The single link algorithm, by contrast, suffers from a chaining effect [8]. The single-link algorithm is more versatile than the complete link algorithm.

In most of the applications, the goal of clustering is to identify some clusters in the given data. The hierarchical clustering algorithm won't stop unless we provide a *stop condition*. This can be the number of iterations of step 2 and 3, the number of clusters that we want to obtain or a certain error indicated by an evaluation of obtained clusters. Those stop conditions are chosen accordingly with

some extra information about the data of the problem we want to solve. To choose the appropriate stop condition is not always an easy task.

The problem we address in this paper is to find the clusters by using single-link hierarchical clustering (which, on the other hand is one of the most simple and intuitive methods) without having to bother about providing a stop condition. The idea is to limit the similarity values between elements that can be used to compute the similarity between 2 clusters (step 3) to the best similarities. We will refer to the chosen number of best similarities as *NBS* (Number of Best Similarities).

3. The Basic Idea

The question to which we are going to answer now is how many *similarities are used* for building k clusters for N elements, where $1 \leq k \leq N$.

In single linkage, in each formed cluster, we can build a tree formed by similarity links used to build that cluster. If n_i are the number of the elements in the cluster, than there are $n_i - 1$ similarities used. With the notation:

$$UsedSimi(n_i) = number\ of\ similarities$$

the next relation holds:

$$UsedSimi(n_i) = n_i - 1$$

Suppose there are k clusters build and the number of elements in each cluster are $n_1, n_2 \ldots, n_k$. The next relations hold:

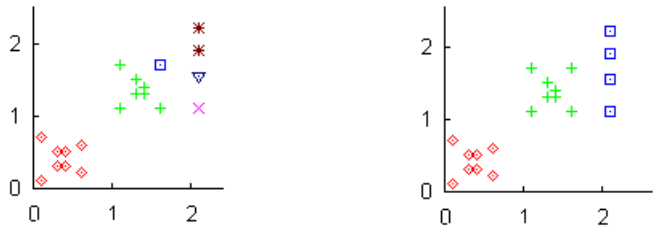
(1) $n_1 + n_2 + \ldots + n_k = N$
(2) the total number of used similarities is the sum of the numbers of used similarities for each cluster; that is:
$AllUsedSimi = \sum_{i=1}^{k} UsedSimi(n_i) = \sum_{i=1}^{k}(n_i - 1) = N - k$
(3) $1 \leq k \leq N$

The idea is to consider only the $N - k$ best similarities to build the clusters. The clustering process will end when there is no similarity left from the best NBS that can be used by the clustering process.

During the clustering process, we are not dealing only with the best similarities among elements to be the similarities that are used for building the cluster. Some of them are lost for other intra-cluster similarity values. This is one important property of the method and we are going to find a way to workaround the error introduced by this property and also to profit by this.

See, for example, the clusters that are produced by using this method and first N (= 20) best similarities, that are larger than any $N - k$, in Fig. 1(a) [1].

[1]The graphics is made by using gnuplot



(a) NBS=N (b) Adjusted clusters

FIGURE 1. Clusters without and with elimination of singular elements

Some of the best similarities are extra-links [2] inside the clusters that are built. The number $N - k$ is too small, the chosen NBS must consider also that the extra-links in the clusters are similarities values that can have large value but will not be used by the algorithm. On the other hand, our formula depends on k and this is an undesired thing. Consequently, if we consider as NBS the maximum value that can be obtained by the above formula, we accomplish two requirements: get a larger value as NBS in order to ignore some extra-links inside the cluster and also have the advantages of having a NBS value that does not depend on k:

$$NBS1: \ max_k(N-k) = N$$

On the other hand, the decision to take only the best N similarity values remains sometimes a little too strong, as we can see in Fig. 1(a). The reason is that maximization of $(N-k)$ does not cover enough extra-links inside the clusters. But we can consider that, there is still a good chance that NBS correctly identifies clusters nuclei.

If we consider that the clusters nuclei are correctly identified, we can improve the result by continuing to group clusters with one element (in single-link hierarchical manner) until the moment when the clusters that must be unified are nuclei determined by NBS.

The question that arises is what can be considered as a cluster nucleus and when an element is *singular*[3], that means that is not part of a nucleus. If the elements are grouped in clusters, we expect that there are elements enough close so that they would be put together in a cluster - for each natural cluster which the data contains. Consequently, we will consider as singular elements the ones that are singular in clusters, and as nuclei - clusters with more than one element.

[2] In this case, we have notated as extra-link the links among elements in a cluster that are not used by the single-link clustering algorithm to form the cluster

[3] We say that an element is singular if it forms a cluster by himself

By applying this method on the same data sets as in Fig. 1(a), the cluster set build in this case is indicated on Fig. 1(b).

4. The Choice of NBS

4.1. **Implementation Issues.** We evaluate our method by hand, by using sets of points in a two dimensional space and evaluating the computed clusters. We use as similarity a measure derived from the *Euclidian metric distance*. If d is the Euclidian distance between two points p and q:

$$d(p,q) = \sqrt{\sum_{i=1}^{2}(p_i - q_i)^2}$$

then the similarity between them can be computed by using the formula:

$$similarity(p,q) = \frac{1}{d(p,q)} \quad (A)$$

Of course, this metrics holds if there are not 2 elements with the same coordinates. If this condition is not satisfied, we can use:

$$similarity(p,q) = \frac{1}{d(p,q)+1} \quad (B)$$

In our experiments, we are in case when there are not 2 elements with the same coordinates, and we have taken formula (A) for computing the similarities.

Computation with real values introduce small errors. On the other hand, we are not interested to put in different clusters the elements that are closest. That is why we are interested in ignoring small variations of similarities. When we identify the best similarity values, we consider as acceptable a variation that is not very small and that depends of the similarity values. In calculus, the variation is interpreted as an acceptable error. We used an *error of 10%* from medium difference between two similarities values, computed with the next formula:

$$\begin{aligned} error &= \frac{(\max - \min)}{(\text{number of similarities})} \\ &= \frac{1}{10} \times \frac{\max - \min}{n*(n-1)/2)} \\ &= \frac{1}{5} \times \frac{\max - \min}{n*(n-1)} \end{aligned}$$

where:

max**:** maximum similarity value
min**:** the minimum similarity value, greater than 0

On the other hand, we are working with a $N * N$ similarity matrix, where the elements on the main diagonal have a special value and are not used. Each other similarity value from the matrix is repeated twice. This means that we use a number of $2 * NBS1$ values from the similarity matrix.

4.2. **Fine tuning the parameter.** The way in which we build the value $NBS = N$ would say that the choice is close to the best one, but it is not necessary the best choice. In order to test this, we will establish a measure of cluster accuracy and we will study the effects of small variation of NBS value.

As is universally accepted there is no universal measure for evaluating cluster sets. We choose for evaluation the next measure:

$$measure = min_{C_i,C_j} dist(C_i, C_j) - min_{p \in Ci}(max_{p,q \in C_i} dist(p, q))$$

and we will earn from the fact that this measure performs well if there are no singular points that must be part of a cluster and they are not. Because it is a measure of optimum (max or min), not of average, and one wrong cluster will modify the result of the evaluation. As we *continue clustering* starting from a determined set of nuclei and as long as the set of nuclei remains unchanged, we get a good chance of eliminating singular points, and so, eliminating the wrong clusters.

The method we propose is to take as the result the set built for NBS greater than N and smaller than $3/2N$ that get a score value at least double compared to the score for the set built for N, or the set built for $NBS = N$ otherwise. We ask for the score value to be double because we considered that, if the improvements are not big, than the better score could appear by cause of the natural tendency of the evaluation function to grow with the decrease of the number of the determined clusters set.

We experimented the result by taking as NBS the values that approximate *the interval*: $N-N/2 \ \ldots \ N+N/2$. That is, we considered as the first best similarities those with the values ranked between $(1; N) \ \ldots \ (1; 3N)$ from the $N*N$ similarities of the similarity matrix. The distinct sets of obtained clusters are shown in Fig. 2. Each set of clusters are accompanied by a short explanation as follows:

- on the first line: the score of the cluster set
- on the second line: the smallest value of $2*NBS$ (value ranks are between N and $3 * N$) that obtain one set of clusters

One very important thing the experiment enlightens is that the result is *not strongly dependent of* the chosen NBS value, in the sense that small variations of NBS keep the result (clusters set) unchanged. Note that there are only 9 distinct cluster sets for a NBS value that vary between $NBS = N = 46$ and $NBS = 3N = 138$, that is for 92 distinct values for NBS.

The experiment confirms that the best result is very *close to* $NBS = N$. Another thing the experiments points is that when NBS grows bigger, the clusters grow bigger too and are less well identified, while the evaluation function value grows either. That is why we cannot use only the evaluation function to identify the best set of clusters identified during the hierarchical clustering process.

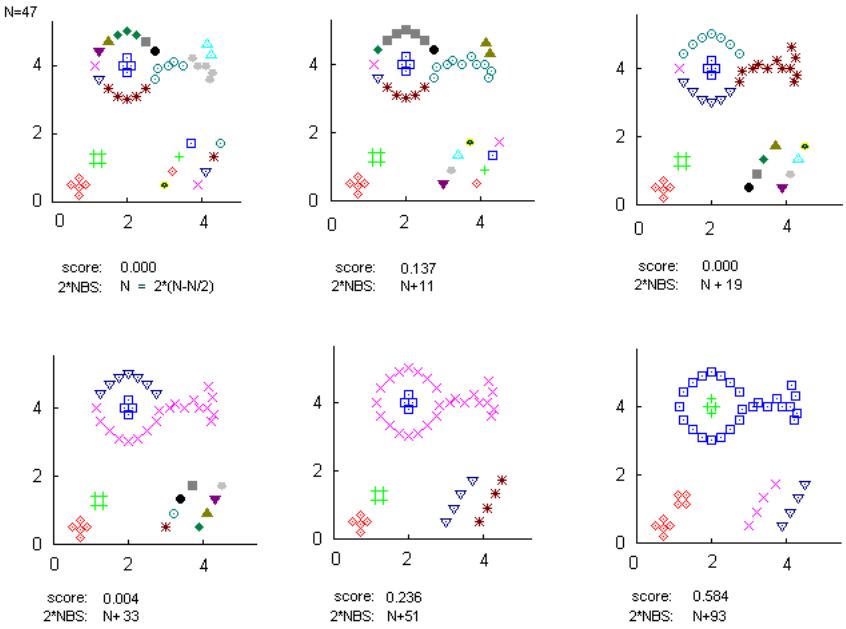


FIGURE 2. All different clusters levels when considering best similarity values, with ranks between (1 and $N/2$) and (1 and $3N/2$

We also studied the behavior of other 10 data sets. Based on the results of these experiments, we are going to improve the method by taking as result the first set of clusters built for an NBS value that satisfies $NBS \geq N$ and $NBS \leq N + N/2$, if the set is evaluated as being better than the set build for $NBS = N$.

4.3. **Best choice. Experimental results.** We have taken 5 different types of input sets, with 2 examples for each type. That would be a collection of 10 data sets. We choose data with characteristics presented in the table 1.

The results of clustering processes are presented in Fig. 3.

Evaluation. We take as precision the elements that are considered by a human subject that are well grouped. We are looking for the most general clusters a human judge would identify. As we evaluated, there are 8 correct cluster results. We indicate the sets 4A and 5A as not being correct. That would indicate a percentage of 80% correct clusters.

4.4. **Discussion of special cases.** One case of bad distributed elements is the case when there should be clusters with few elements in a cluster. The figure 4 presents cluster changes with the variation of the number of points.

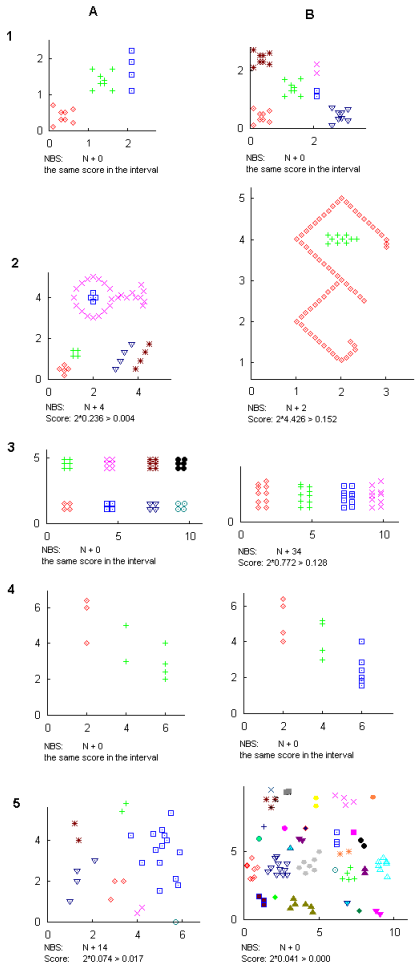


FIGURE 3. Clusters for different values for NBS

If the elements are not relatively uniform distributed inside the clusters, the method won't always obtain good results. In figures 4 and 5 we have the results from data more or less well distributed.

Figure 5 illustrates that, if the clusters are well identified, the result suffers very little from small variation of elements' coordinates.

One could say that the cluster set 3 in the fig. 5 is inaccurate. But what would be the result if the distances between the elements in clusters 3, 4, 5, 6 are modified

Data characteristic	**Identification in fig**
well grouped in clusters	1, 2, 3
well grouped in clusters and known as with problem for hierarchical clustering (there are differences between single-link and complete-link hierarchical clustering)	2
bad cluster identification many points but sparse data	5
small number of elements in a cluster	4
many clusters with a small number of elements in a cluster	3A
small number of clusters with many elements in a cluster	2B

TABLE 1. Characterisitc of the data in the figure 3

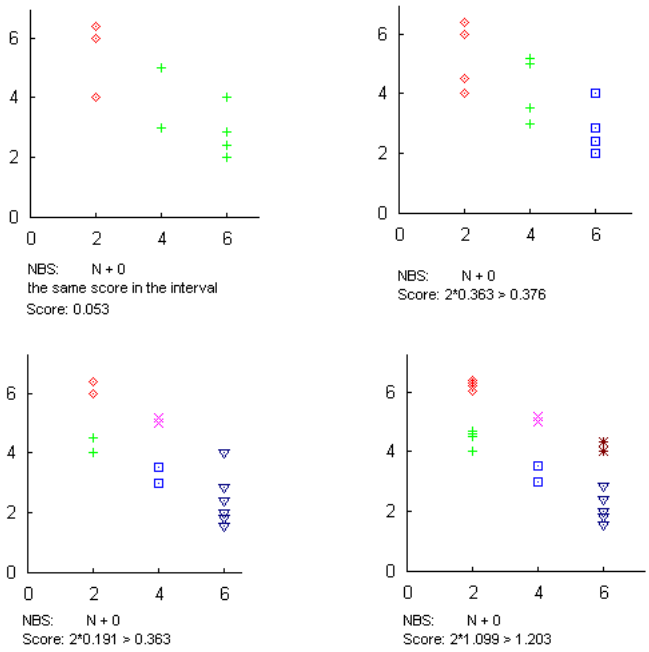


FIGURE 4. Clustering over reduced number of elements

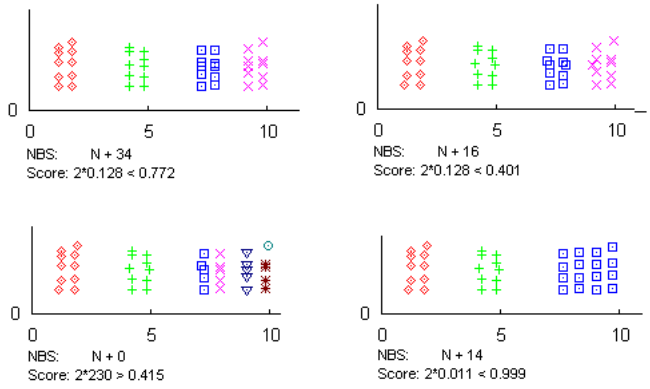


FIGURE 5. Clusters built for small variation of elements coordinates

in order to have close value? The result is represented in the 4^{th} data set in the figure.

As we can see in figure 5, the results for elements that are not uniform distributed in a cluster are disputable also for human judges. Consider the clusters from the figure 6. Which of them do we have to consider best? On the other hand, the elements coordinnates in the two images in the figure are the same. But they are represented to a different scale. In one the separation among clusters is observable, and in the other is not. A human judge won't observe that. For this data set our algorithm identifies clusters indicated in figure.

Sparse data is also an example of not relatively uniform distributed elements in a cluster. In this case, the identified clusters are not very close with those identified by a human judge. If the result are good or not is disputable, as we can see in the figure 3, sets 5A and 5B. The explanation is that the result for sparse data is better when the clusters are more compact. This corresponds to smaller value for NBS.

5. Conclusions and Future Directions

The algorithms we build have the advantage to build the clusters without the need of some stop condition, so it is a really unsupervised method. We consider the result as good, as long as the tests indicate an accuracy of about 90% for data 'well' grouped in clusters. The accuracy of the results depends on dispersion of the elements inside the 'ideal' cluster and the number of the elements inside a cluster (the bigger, the better). Usually, the algorithm does not work so well if the clusters in the data do not have many elements, because the number of all elements is small or the data is sparse.

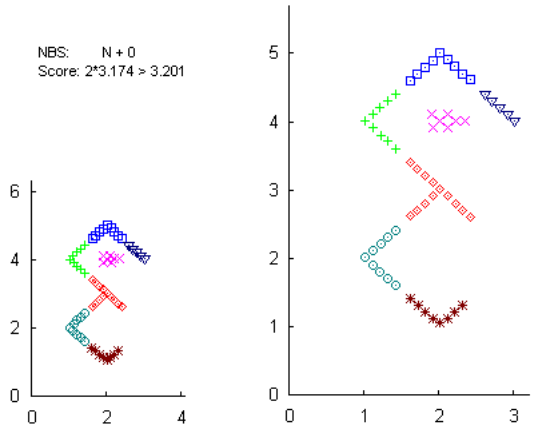


FIGURE 6

One of advantage of the method is that the result do not depend directly from the NBS, there is an interval of values for which the results are the same.

The method was built for cases when clusters are 'well identified' and the elements are relatively uniform distributed inside the clusters. But, for any sparse data, there is no guarantee that there is a human judge that identifies clusters. This is the drawback of the suggested mechanism, if we want to compare it with the absolute case of a human judge. As is known, in case of sparse data, the complete-link method is more appropriate than the single-link.

One of the future directions we are working on is to develop a similar method also for the complete-link hierarchical clustering and compare the results of the two methods.

We also intend to apply this method to pattern recognition in image processing, because we think that this is a domain where the method should apply with best result.

REFERENCES

[1] Baeza-Yates, R.A. *Introduction to data structures and algorithms related to information retrieval*, 1992;
[2] Berkin, P., *Survey of Clustering Data Mining Techniques*, 2002;
[3] Cao, F. et. al., *An a contrario approach to hierarchical clustering validity assessment*, 2004;
[4] Dubes, R.C., *How many clusters are best? - An experiment*, 1987;
[5] Hearst,M., *Applied Natural Language Processing*, Lecture Notes, 2004;
[6] Jain,A.K., M.N. Murty, *Data Clustering: A Review*, ACM Computing Surveys, Vol. 31, No.3, September 1999;
[7] Massey,L., *Determination of Clustering Tendency With ART Neural Networks*, 2002;

[8] Nagy, G., *State of the Art in Pattern Recognition*, Proc. IEEE 56, 836-862, 1968;
[9] Matteucci, M., *A Tutorial on Clustering Algorithms*,
http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/index.html
[10] Annotated Computer Vision Bibliography
http://iris.usc.edu/Vision-Notes/bibliography/pattern617.html

Babes-Bolyai University, Faculty of Mathematics and Computer Science, Department of Computer Science, Cluj-Napoca, Romania
E-mail address: davram@cs.ubbcluj.ro