Random Dendrograms

Dimitar L. Vandev*

Abstract

An attempt is made to create some statistical tests for comparing results of hierarchical cluster analysis based on the uniform distribution over the set of all possible dendrograms. Three different uniform distributions are considered according to the degree of similarity of the dendrograms. Some distances between dendrograms are defined and The solutions proposed are computational and are based on the embeding the sets of equivalent dendrograms into the set of lexicographically ordered words.

Keywords: cluster analysis, dendrogram, distance

1 Introduction

Cluster analysis attempts to group the objects of an observed set, on the basis of similarity or distance between them, into mutually exclusive subsets (clusters) which consist of close objects. These clusters may be grouped into larger sets and so on, until all points are eventually united in one cluster. The higher the level of aggregation is, the less similar are the objects in the respective cluster. These methods for cluster analysis are called hierarchical. The result of hierarchical classification can be represented graphically by a dendrogram.

An attempt is made in this paper to create some statistical tests for comparing results of hierarchical cluster analysis based on the uniform distribution over the set of all possible dendrograms.

The solution proposed is rather computational and is based on embedding the set of dendrograms into the set of lexicographically ordered words.

2 Notations and definitions

Usually the output of any cluster analysis program is an tree - like structure called dendrogram. In the computational part of the program this is a table, each row of which contains the results of one amalgamation step. If n objects are to be clustered then the table contains exactly $n-1$ rows. For example, in the table 2 the first step joins the objects 1 and 2 at distance .334 into a new object-cluster 2. At the subsequent step objects 3 and 4 are joined into object 4 and so on, until at the last we have one cluster containing all original objects. Of course when instead of distances similarities are considered, then the numbers of the third column are in a reverse order.

*Postal address: Faculty of Math.and Informatics, Sofia University, J.Boucher str.5,1000 Sofia,Bulgaria.
Supported by National Fondation of Scientific Investigations, grant M-440/94.

1	1	2	.334	6	2	4	.666	11	4	17	1.134	16	17	18	1.741
2	3	4	.434	7	6	11	.823	12	11	18	1.135	17	18	20	1.871
3	5	6	.542	8	9	20	.934	13	13	20	1.334	18	7	20	2.000
4	8	9	.553	9	12	13	.942	14	16	17	1.434	19	19	20	2.001
5	10	11	.563	10	15	16	.963	15	14	20	1.443				

Table 1: Cluster Analysis of 20 objects

Let now formalise the properties of such tables (dendrograms). Suppose that the number of objects is n. Denote by m_i and n_i the numbers in the i-th row (first two columns) and let $Z_n = \{1, 2, \ldots, n\}$. Let further suppose that the numbers in the third column are all different.

Definition 1 *We say that two dendrograms are strongly equivalent if the first two columns of the corresponding amalgamation table coincide.*

Definition 2 *We will call the vector of natural numbers* $\{m_1, n_1, m_2, n_2, \ldots, m_{n-1}\} \in R^{2n-3}$ *binary dendrogram with labels (BDL) of* n *elements, if it fulfils following conditions:*

1. $1 \leq m_i < n_i \leq n$ *for* $1 \leq i \leq n-1$;
2. $\forall i \leq j : \quad m_i \neq n_j$;
3. $\{m_1, m_2, \ldots, m_{n-1}\}$ *is a permutation of the numbers* $1, 2, \ldots, n-1$.

The number of all different vectors of this kind denote by O_n.

Theorem 1 *There exists one-to-one mapping of the set of all integer vectors* $S_n = \{m_1, n_1, m_2, n_2, \ldots, m_{n-1}\}$ *satisfying the properties 1,2,3 above to the set of all classes of strongly equivalent dendrograms.*

Proof The proof is simple and constructive. ∎
Here after we will use this presentation of strongly equivalent dendrograms.

Definition 3 *We say that two dendrograms are simply equivalent if the first two columns of the corresponding amalgamation table coincide after some permutation of the rows. The number of classes denote by* H_n.

It is clear that according this definition it is not important which of the pairs $1, 2$ or $3, 4$ is amalgamated first. But it remains important that these clusters will be joined together at the next step. So row 6 should appear in the new table after rows 1 and 2. The row 19 remains at the same place. So the permutation is not arbitrary — the result of it should be again an BDL. Some times one calls such a structure binary hierarchy of n elements.

Definition 4 *A set* $\mathcal{H}$ *of subsets of the set* X *is called hierarchy if it satisfies:*

1. $X \in \mathcal{H}$;

2. $\{x\} \in \mathcal{H}, \forall x \in X$*;*

3. If $A, B \in \mathcal{H}$*, then* $A \subset B$ *or* $B \subset A$ *or* $A \bigcap B = \emptyset$*.*

Let take all hierarchy members containing fixed $\{x\} \in X$. Then there exists unique (maximal) strictly increasing chain of members of the hierarchy:

$$\{x\} = B_0 \subset B_1 \subset B_2 \subset \cdots \subset B_k = X$$

. Denote it with $\mathcal{B}(x)$ and the length if the chain by $\mathcal{K}(x)$.

Definition 5 *A hierarchy* $\mathcal{H}$ *of the set* X *is called (binary) if it satisfies:*

$$\forall x \in X, 1 \leq i \leq \mathcal{K}(x), \quad B_i \setminus B_{i-1} \in \mathcal{H}, B_i, B_{i-1} \in \mathcal{B}(x).$$

Thus the numbers H_n represent the number of different binary hierarchies in the set of n elements. Each dendrogram belongs to (or defines uniquely) an binary hierarchy. It is clear that the number of dendrograms within the same hierarchy depends of this hierarchy. We will study in the following its distribution (given uniform distribution over hierarchies). The hierarchies are an interesting object for investigation. The following theorem gives a simple representation of an hierarchy.

Theorem 2 *There exists one-to-one mapping of the set of all integer vectors* $S_n = \{m_1, n_1, m_2, n_2, \ldots, m_{n-1}\}$ *satisfying the properties 1,2,3 above and the additional inequality*

$$n_1 \leq n_2 \leq \cdots \leq n_{n-2}$$

to the set of all hierarchies.

Proof Each node of the dendrogram represents a subset of the set of terminal nodes. Let denote this mapping by T. It is clear that many dendrograms are mapped over the same hierarchy. each correspond to a unique labelled binary tree. Then the rule of "right son - root - left son" leads to this simple modification. The dendrogram - representative of the class - is reconstructed by reading right to left the vector $\{m_1, n_1, m_2, n_2, \ldots, m_{n-1}\}$. ■

Definition 6 *We say that two dendrograms are weakly equivalent if corresponding binary hierarchies coincide after some permutation of the objects. Let denote the number of such classes by* W_n*.*

Again each dendrogram (and each hierarchy) belongs to unique weak class. It is clear that the number of dendrograms (or hierarchies) within the same class depends of this class. We will study in the following these distributions (given uniform distribution over classes).

3 The number of the dendrograms of n objects

In this section we will produce some recurrent formulas for calculating the numbers O_n, H_n, W_n. This is easy and we will see that some of them are well known. We will not use in these numbers in the statistical tests below. However, the derivation of there properties seems to be interesting combinatorial problem. Here a simple proofs of the recurrence formulas are presented for convenience. They follow the technique of [Knuth (1973)].

3.1 Hierarchies and dendrograms

Theorem 3 *The numbers O_n, H_n satisfy the following recurrent formulas:*

$$H_n = \sum_{i=1}^{n-1} \binom{n}{i} H_i H_{n-i} \qquad (1)$$

$$O_n = \frac{1}{2}\sum_{i=1}^{n-1} \binom{n}{i}\binom{n-2}{i-1} O_i O_{n-i} = \frac{n!(n-1)!}{2^{n-1}} \qquad (2)$$

The first several numbers are as follows:

$$H_1 = 1, H_2 = 1, H_3 = 3, H_4 = 15, H_5 = 105$$
$$O_1 = 1, O_2 = 1, O_3 = 3, O_4 = 18, O_5 = 180$$

Proof: Let first $n = 2m + 1$. Then each dendrogram may be splitted after deleting the root into two dendrograms (or hierarchies). So

$$H_n = \sum_{i=1}^{m} H_i H_{n-i} \binom{n}{i} \qquad (3)$$

$$O_n = \sum_{i=1}^{m} O_i O_{n-i} \binom{n}{i}\binom{n-2}{i-1} = \frac{n!(n-1)!}{2^{n-1}} \qquad (4)$$

what coincides with equations 1,2. Here the therm $\binom{n}{i}$ calculates the number of ways to choose i objects out of n. The therm $\binom{n-2}{i-1}$ equals to the number of ways of merging the levels of amalgamation — $i - 1$ into the first branch with $n - i - 1$ levels into the second.

Let now $n = 2m$. Now the last terms ($i = m$) of summations 3,4 should be divided by 2, because each splitting appears twice. Thus formulas 1,refO again are fulfilled.

The second equality in 4 is easy and straightforward. According to the definition at the first level we choose the first pair in a $\binom{}{n2}$ ways. Then at each subsequent level the number of objects decreases by 1. So

$$\binom{n}{2}\binom{n-1}{2}\cdots\binom{2}{2} == \frac{n!(n-1)!}{2^{n-1}}.$$

■

3.2 Unlabelled Dendrograms

Consider the number W_n of different unlabelled dendrograms. The classes of weakly equivalent dendrograms may be looked as binary trees with $2n-1$ unlabelled nodes and fixed root. These numbers are known ([Comtet (1970)]) as Wedderburn - Etherington numbers - "parenthesages commutatifs". It follows the interpretation of so called Catalan numbers for non commutative "parenthesages".

Theorem 4 *The numbers W_n satisfy the following recurrent formulas:*

$$W_n = \frac{1}{2}(W_{n/2} + \sum_{i=1}^{n-1} W_i W_{n-i}) \tag{5}$$

$$\tag{6}$$

The term $W_{n/2}$ is considered zero if n is odd. The first several numbers are as follows:

$$W_1 = 1, W_2 = 1, W_3 = 1, W_4 = 2, W_5 = 3$$

Proof: Let first $n = 2m + 1$ is odd number. Then each dendrogram may be splitted after deleting the root into two unequal (different) dendrograms. So

$$W_n = \sum_{i=1}^{m} W_i W_{n-i},$$

what coincides with equation 5.

Let now $n = 2m$. In this case when sizes of both branches coincide, pairs of the same classes appear twice and a correction should be made to consider only the part above (and including) the main diagonal. So the last $(i = n - i = m)$ of the above summation in this case should look like follows:

$$W_m(W_m + 1)/2$$

Combining these formulas we obtain the result formula 5. ■

In [Comtet (1970)] the generating function $p(t)$ of the numbers W_n is studied and it is shown that it satisfies the following functional equality:

$$p(t) = t + \frac{1}{2}(p^2(t) + p(t^2).$$

Using the substitution $q(t) = -1 + p(t)$ simplifies it to

$$q^2(t) + q(t^2) + 2t = 0.$$

The exact value and asymptotic behaviour of these numbers remains an open question.

4 Ordering and enumerating

4.1 Dendrograms

Let now arrange the set S_n of all classes of strictly equivalent dendrograms (BDL) in a lexicographical order. Then the set S_n may be embedded as a subset of the set of words of length $2n - 3$ with an alphabet the set Z_n. As shown in [Vandev (1996)] there exist many different possibilities to do this embedding.

In the table 4.1 the set of all dendrograms of 4 objects are shown in an lexicographic order. We need this embedding in order to create an easy enumerating algorithm. This will give us

1 2 2 3 3	1 3 3 4 2	2 3 1 4 3	3 4 1 2 2
1 2 2 4 3	1 4 2 3 3	2 3 3 4 1	3 4 1 4 2
1 2 3 4 2	1 4 2 4 3	2 4 1 3 3	3 4 2 4 1
1 3 2 3 3	1 4 3 4 2	2 4 1 4 3	
1 3 2 4 3	2 3 1 3 3	2 4 3 4 1	

Table 2: Dendrograms of 4 elements

the possibility to calculate "the probability" of different events as a simple proportion of the number of "good" dendrograms to $O_n = |S_n|$.

According [Vandev (1996)] the enumerating algorithm may be constructed in the following way:

function $next(word)$
  $k = Last\_Not\_Last(word)$;
    If $k = 0$ stop;
  $word := Increase(k, word))$;
  $word := First(k, word)$;
end

So it is enough to construct corresponding functions in this algorithm.

1. $k = Last\_Not\_Last(word)$ calculates the place of first change of a letter. In our case let take the smallest i that

   n. $n_i \neq n$ then $k = \max(2i, 0)$;

   m. $m_i \leq m_{i+1}$ then $k = \max(2i - 1, 0)$

2. $Increase(k, word)$ increases the k-th letter of the word. Form the subset $F_k \subset Z_n$ consisting of the letter n and all letters after (and including) the k-th position. Then $Increase(k, word)$ puts in k-th place next available letter from the (ordered) set F_k.

3. $First(k, word)$ states smallest possible numbers after the position k taken from the set F_k. Numbers m_i form the next available permutation, and numbers $n_i = \max(m_i + 1, \min_{F_k} n)$.

   - In last two cases if a value is assigned to m_i, then this number is extracted from the set F_k, otherwise F_k remains unchanged. The action is performed step by step increasing the pointer k.

In order to check and simplify the algorithm we need some considerations.

Lemma 1 *The set F_k contains exactly $n - i$ elements and consists of only two parts:*

- *the number* n.
- *the members of the permutation of the numbers* $(m_1, m_2, m_3, ..., m_{n-1})$, *i.e.* $\pi(1, 2, \ldots, n-1)$, *present in the word after* $k-1$ *position.*

Proof In the definition 1 of function *Last_Not_Last* there are two cases. In the case m. the statement is trivial. Consider case m. Let $k = 2i > 0$. Denote by $word(k) = n_i \neq n$. Then according item 2. of definition 2 we have: $\forall j \leq i : \quad n_i \neq m_j$. As $\{m_i, i = 1, 2, \ldots, m_{n-1}\}$ is a permutation the statement follows: $\exists j > i : \quad n_i = m_j$. ■

Lemma 2 *The numbers* m_i *when put by the function* $First$ *form always an strictly increasing sequence.*

Proof Because this is the first available permutation of the set $F_k \setminus n$. ■

Lemma 3 *When put by the function* $First$*, the numbers* $n_i, i \geq [k + 1/2]$ *form always an strictly increasing sequence.*

Proof When m_i is the first number to assign the statement is trivial — it is the smallest number in F_k and after assigning is to be extracted. So n_i is to be next number in F_k, $m_{i+1} = n_i$ and so on. When n_i is the first number to assign, it may arbitrary in F_k. So the increasing starts from the next step. ■
Comment 1. When implementing this algorithm as a program, it is convenient to form the ordered set F_k together with the evaluating the function *Last_Not_Last* simply inserting new elements m_i in the proper position. Note that $\{m_i\}$ are ordered, so this is to be done only once and in this case it is only replacement by the next element needed for performing the function $Increase$.

4.2 Hierarchies

In order to enumerate the hierarchies we will choose an element of every class and use the same presentation as for dendrograms as in theorem 2. In table 4.2 the representatives of all hierarchies of 4 elements are presented. The empty places represent the "missing" dendrograms of table 4.1 for convenience.

1 2 2 3 3	1 3 3 4 2	2 3 1 4 3	...
1 2 2 4 3	...	2 3 3 4 1	3 4 1 4 2
1 2 3 4 2	1 4 2 4 3	...	3 4 2 4 1
1 3 2 3 3	1 4 3 4 2	2 4 1 4 3	
1 3 2 4 3	2 3 1 3 3	2 4 3 4 1	

Table 3: Hierarchies of 4 elements

It is clear that the enumeration problem may be solved by the same algorithm by using seeding of the good dendrograms — having the property $n_i < n_{i+1}$. However it is easy to modify the algorithm above for enumerating hierarchies instead dendrograms. It is enough to specify the function $First$ properly: the new numbers n_i should be not less then the last nonchanged number n from the prefix part of the word (and form nondecreasing sequence).

We will not consider here the effectiveness of this algorithm. It seems that this is not an easy question. What immediately follows is that the mean number of steps down the tree (until the place of change k is reached) does not increase infinitely when $n \to \infty$. This is because of the permutation part of the word. The limit generating function of the number of steps for permutations is calculated in [Vandev (1996)].

4.3 Dual order

The ordering considered up to now is not very natural because the near dendrograms (or hierarchies) differ in the root part. It seems interesting the consider the dual ordering when the root remains fixed. Suppose now we consider words in the following transcription:

$$w = (n_n, m_n, m_{n-1}, m_{n-1}, ..., n_1, m_1)$$

.

This is according [Vandev (1996)] the exact dual order. We have then that always $n_n = n$ and for hierarchies n's form an nonincreasing sequence. The general algorithm works with the following changes:

1. $k = Last\_Not\_Last(word)$ calculates the place of first change of a letter. In our case let take the smallest i that

   m. $m_i \leq m_{i+1}$ then $k = \max(2i - 1, 0)$

   n. $n_i \neq n$ then $k = \max(2i, 0)$;

2. $Increase(k, word)$ increases the k-th letter of the word. Form the subset $F_k \subset Z_n$ consisting of the letter n and all letters after (and including) the k-th position. Then $Increase(k, word)$ puts in k-th place next available letter from the (ordered) set F_k.

3. $First(k, word)$ states smallest possible numbers after the position k taken from the set F_k. Numbers m_i form the next available permutation, and numbers $n_i = \max(m_i + 1, \max_{F_k} n)$.

   - In last two cases if a value is assigned to m_i, then this number is extracted from the set F_k, otherwise F_k remains unchanged. The action is performed step by step increasing the pointer k.

4.4 The program

Here we will use considerations of above subsections to produce a simple FORTRAN program implementing both enumerations: of dendrograms (ind=0) and hierarchies (ind=1).

	DDNT	PHRM	NURS	HOSPB	ANIM	STRCH	LFEXP
1 Algeria	129	023	350	3392	21	57	35
2 Iran	329	107	290	1113	24	60	51
3 Iraq	241	081	235	1898	28	57	54
4 Jordan	284	096	241	1712	25	49	52
5 Lebanon	933	191	564	4071	35	50	60
6 Libia	338	041	612	3215	24	55	57
7 Morocco	094	026	233	1516	21	57	53
8 Syria	254	070	140	1163	13	69	52
9 Tunizia	114	339	248	2967	21	57	53
10 Tyrkey	412	057	306	1738	16	71	55
11 UAR	483	131	454	2225	15	73	54

Table 4: Health indicators

5 Distances between dendrograms

We will consider two kinds of distances. The first one is the usual Eucledian distance between vectors representing BDL. It is clear that it is not well behaving when comparing hierarchies or unlabelled trees.

The second distance is involved by the lexicographical order. All BDL are considered as ordered on the real line at equal distances.

Here follows the natural definition of distance between hierarchies.

$$d_H(\mathcal{H}_1, \mathcal{H}_2) = \max(\max_{A \in \mathcal{H}_1} \min_{B \in \mathcal{H}_2} \sharp(A \Delta B), \max_{B \in \mathcal{H}_2} \min_{A \in \mathcal{H}_1} \sharp(A \Delta B)),$$

where $\sharp(A)$ means the number of objects in A.

6 Examples

The example of SLCAD is from BMDP 81 reference manual ([Dixon (1981)]). The data are health indicators for 11 countries given in table 6. The health indicators measured are the relative number of doctors and dentists (DDNT), of pharmacists (PHRM), of nurses (NURS) and hospital beds (HOSPB), the percent of animal fat (ANIM) and starch (STRCH) in the diet and life expectancy (LIFEXP).

The distance measure between cases is the Eucledian distance using standardized data. Table 6 contains the upper half of the distance matrix:

The amalgamation order and the tree diagram of the clusters according to the classical algorithms are given respectively in fig.1 and fig.2. The case numbers are printed below the diagram. Each horizontally line in the tree corresponds to a cluster formed in the hierarchical clustering process. The vertical axes provide a scale by which to measure the dissimilarity between two merged clusters (amalgamation distances).

	2	3	4	5	6	7	8	9	10	11
1	3.97	3.84	3.82	6.83	4.07	3.52	4.40	2.99	4.30	4.75
2		1.39	1.57	5.33	3.49	2.07	2.45	2.59	2.32	2.83
3			1.21	5.10	3.11	1.73	3.01	1.85	2.78	3.54
4				5.08	3.31	2.00	3.30	2.19	3.29	3.88
5					4.44	6.48	7.10	5.82	6.09	5.33
6						3.36	4.65	2.77	3.51	3.59
7							2.37	1.50	2.50	3.82
8								2.91	2.50	2.90
9									2.68	3.62
10										1.88

Table 5: A dissimilarity matrix for 11 objects

Amalgamation order					
3	4	1.21	10	11	1.88
2	4	1.39	9	11	2.32
7	9	1.50	6	11	2.77
8	10	1.60	1	11	2.99
4	9	1.73	5	11	4.44

Fig.1

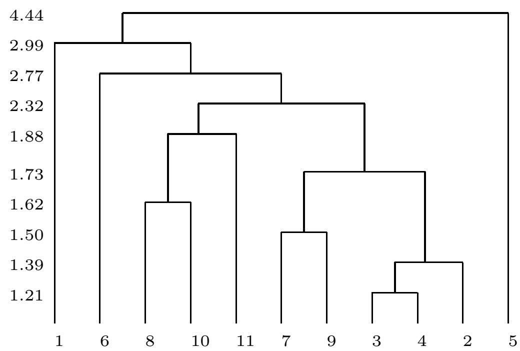

Fig.2

Let now try to test the hypotesys H_0 that this dendrogram is random against the alternative H_1 that at level of amalgamation 2.00 there exist exactly 5 clusters: $(2, 3, 4, 7, 9)$, $(8, 10, 11)$, (1), (5), (

References

[Jambu (1978)] Mishel Jambu, *Classification automatique pour l'analyse des donnees,* Bordas, Paris, 1978.

[Dixon (1981)] W. J.Dixon, M. B. Brown, L. Engelman, J. W. Frane, M. A. Hill, R. I. Jennrich, J. D. Toporek, *BMDP Statistical Software 1981,* University of California Press, Berkeley, 1981,p.456 .

[Rolf (1982)] F. James Rolf, Single-link Clustering Algorithms. In: P. R. Krishnaiah and L. N. Kanal, eds., *Handbook of Statistics, Vol. 2,* North-Holland Publishing Company (1982), 267-284.

[Gower,Ross (1974)] J. C.Gower, G. J. S. Ross, Minimum spanning trees and single-linkage cluster analysis. *Applied Statistics,* 18, 54-64.

[Knuth (1973)] Donald E.Knuth (1973) *The art of computer programming*, volume 1, Fundamental Algorithms, Addison-Wesley.

[Comtet (1970)] Louis Comtet (1970) *Analyse Combinatoir*, volume 1,2, Presses Universitaires de France, Paris.

[Evans, Minieka (1992)] James R.Evans, Edward Minieka, *Optimization Algorithms for Networks and Graphs*, Marcel Dekker, 1992, Second Edition.

[Kruskal (1956)] Kruskal, J.B.(1956) On the Shortest spanning subtree of a graph and the travelling salesman problem, *Proceedings of the American Mathematical Society* 7, 48 – 50.

[Haynatzki (1994)] Gleb R.Haynatzki (1994) Restructured Prim's algorithm for graphs with bounded neighbourhoods. *The Mathematical Scientist* 19, 78 – 86.

[Vandev (1992)] D. Vandev. Alphabetical Ordering - Useful Tool for Creating Algorithms; *8th Seminar on Statistical Data Analysis 1992*, 121-126.

[Vandev (1996)] D. Vandev. Some Examples of Lexicographic Algorithms and some Open Combinatorial Problems (submitted in).