International Journal of Pure and Applied Mathematics
Volume 21 No. 3 2005, 319-333

CLUSTERS IN METRIC SPACES

C.H. Jamison[1], C. Turner[2], G.E. Livesey[3],
H.A. Donegan[4 §], J.McC. Anderson[5]

[1,2,5]School of Electrical and Mechanical Engineering
University of Ulster
Co. Antrim, BT37 0QB, N. IRELAND, U.K.
[1]e-mail: ch.jamison@ulster.ac.uk
[3]School Education
University of Ulster
Co. Antrim, BT37 0QB, N. IRELAND, U.K.
[4]School of Computing and Mathematics
University of Ulster
Co. Antrim, BT37 0QB, N. IRELAND, U.K.

Abstract: This paper describes 3 methods which outline the construction of clusters within a metric space. We can choose a different radius for each open ball around each of the points contained within the metric space. An added bonus of the 3 methods is that the density of the clusters (and sub-clusters from which they are constructed) can be established using a straightforward calculation. There are many possible applications for these methods and an immediate application to the area of egress complexity is outlined.

AMS Subject Classification: 54E35
Key Words: metric space, cluster, egress, egress complexity, route complexity

1. Introduction - Metric Spaces

A number of clustering methods and algorithms are readily available e.g. Jain et al [1] for determining which elements of a space (metric, vector or otherwise) lie

Received: April 28, 2005
© 2005, Academic Publications Ltd.
§Correspondence author

within the same clusters in the defined space. We present here a method based on a metric space, the open ball surrounding each point in our metric space can be of a different radius for each point in the metric space (we follow Jain et al [1] using the word *method* to indicate a strategy, whilst *algorithm* denotes a program, used in the construction of a cluster). The clusters, which consist of overlapping open balls are built using the methods presented in this paper. Our particular method was introduced to quantify the dissimilarities of a range of patients' readings derived from electrocardiogram (ECG) signals in $\mathbb{R}^4$. The method is useful in those cases where a group of points in a metric space is pre-processed to produce a set of clusters which can then be compared with a new data set to determine its proximity to the pre-processed clusters. The method has the advantage of building sub-clusters of open balls around each data point in the metric space, thus making the process of density estimation of the sub-clusters, and hence the density of the cluster of which they are component parts to be ascertained.

2. Standard Definitions of a Metric Space

The definitions of a metric space and open ball (near neighbour) are available from any standard text on the subject Sutherland [2] and are also usually outlined in any papers related to the field e.g. Chávez et al [3] and Brin [4].

Metric Space. A *metric space* is a set X with a distance function $d : X^2 \to \mathbb{R}_0^+$ such that: $\forall x, y, z \in X$:

1. $d(x, y) \geq 0$ and $d(x, y) = 0$ iff $x = y$ (positivity);
2. $d(x, y) = d(y, x)$ (symmetry);
3. $d(x, y) + d(y, z) \geq d(x, z)$ (triangle inequality).

Open Ball or Near Neighbour. Given a metric space (X, d), $x \in X$, $r \in \mathbb{R}_0^+$, the *open ball* or (*near neighbours*) of x are $Y = \{y \in X : d(x, y) < r\}$

Cluster. A collection of open balls B_i each with centre x_i and distance associated with the metric r_i $\{B_i(x_i, r_i) \ : \ i \in I\}$ is called a *cluster* if $\forall$ i, $j \in I$, $B_i \cap B_j \ \neq \ \emptyset$.

3. Searching the Metric Space

Usually the dataset to be searched for proximity to a particular point changes for each point to be searched for its nearest neighbours. However, we have a slightly different problem to that outlined in most papers on the subject i.e. our initial dataset (the group) will rarely change. We essentially have a large collection of $B_i(x_i, r_i)$ $i \in I$. Thus we can pre-process our group dataset, to produce clusters which will help minimize the search times in finding group dataset points close to that current data being compared with the pre-processed group data.

The usual approach to searching metric spaces is a case of finding the k nearest neighbours $(k - nn)$ of a point z. In the approach used in this paper however our aim is to produce clusters of open balls $B_i(x_i, r_i)$ $i \in I$ (the distance r associated with the ball could be the same for each $d(x, y)$ comparison, but there is no loss of generality of the method if a different radius for the distance associated with the metric is chosen for each of the $d(x, y)$ comparisons between points).

The rationale for defining a cluster is that $d(x, y)$ is normally time consuming to compute, so we wish to minimize the number of distance calculations carried out. We should be able to build a data structure from the group data which allows us to consider the large number of points in the group data with the small number of points in the data of the subject under study. For each of the points of subject data we need to consider its proximity to all the points contained in the group data.

4. Building Clusters within the Metric Space

To pre-process the metric space group data into clusters we must first carry out the distance evaluations between the centres of all the open balls. Since $d(x_i, y_j),\ \ x_i, y_j \in X,\ \ i, j \in I,\ \ i \neq j$ is symmetric we need only carry out one of the two possible distance evaluations for each pair of points. This will give us a total number of comparisons consisting of

$$S_n = \sum_{k=1}^{n-1} k = \frac{n(n-1)}{2}$$

values.

Suppose that the open balls under consideration are indexed by $i \in I$ then $\{B_i : B_i = B(x_i, r_i), i \in I\}$. There are 2 cases to consider for each pair of open balls, these being either: (i) when the open balls intersect or, (ii) when the open balls *do not* intersect. We are interested in case (i) as this is where clusters have started to form. More formally for the comparison between each pair of open balls we have

$$C_{ij} = \begin{cases} 0 : B_i \cap B_j = \emptyset\,, \\ 1 : B_i \cap B_j \neq \emptyset \end{cases}$$

(again in this case we need only make the comparison between each pair of open balls once as $B_i \cap B_j = B_j \cap B_i$). The elements $C_{ij}(i, j \in I)$ elements form a matrix $\mathbf{C}$.

Consider the columns of the matrix $\mathbf{C}$. In each column $j \in I$ we have a sequence of $i \in I$ rows of 0's and 1's, therefore we can immediately deduce that the open ball B_i forms a cluster with each of the B_j open balls where $C_{ij} =$ 1. However this process does not give us the final set of clusters, but rather a subset of clusters from which we can build our final set. The columns of matrix $\mathbf{C}$ consist of a list of those open balls that intersect directly with the open ball associated with a particular column. Thus the matrix $\mathbf{C}$ represents the pairwise of each open ball in the metric space with all other open balls in the metric space.

5. Building Clusters – Method A

For each $i, j, k \in I,\ j \neq k$ we compare C_{ij} with C_{ik} in turn, where i represents the row and j, k the columns of $\mathbf{C}$ under consideration. If $C_{ij} = C_{ik} = 1$ then we can deduce that the 2 sub-clusters represented by the 2 columns intersect at the open ball B_i. Thus we may merge column j of $\mathbf{C}$ with column k of $\mathbf{C}$ and replace each row C_{ij} with a 1 when $C_{ik} = 1$. At this stage the column j of $\mathbf{C}$ contains the two sets of open balls merged from the original columns j and k of $\mathbf{C}$, and we can dispose of the column k.

This process is repeated in turn for each column, (*both* newly amalgamated columns and the remaining un-compared columns). Thus we are left with a number of columns which represent the clusters of open balls within the metric space. For each value in the resultant matrix $\mathbf{C_{ij} = 1}$ indicates B_i and B_j are in a cluster together.

6. Building Clusters – Method B

For each $i, j \in I,\ i \neq j$ we check each to see if $C_{ij} = 1$ in turn row by row, column by column, where i represents the row and j the columns of $\mathbf{C}$ under consideration. If $C_{ij} = 1$ then we can deduce that the 2 sub-clusters represented by the 2 columns i and j of $\mathbf{C}$ intersect at the open ball B_j.

We then merge *columns* j and i of **C** and replace each row in column i of **C** with a 1 in turn if the corresponding row of j is equal to 1. At this stage the column i contains the two sets of open balls merged from the original i and j, and we can dispose of column j.

This process is repeated in turn for each column, (*both* newly amalgamated columns and the remaining un-compared columns). The presence of the open balls B_i and B_j in the cluster is given by $\mathbf{C_{ij} = 1}$.

7. Building Clusters – Method C

For each $i, j \in I,\ i \neq j$ we check each to see if $C_{ij} = 1$ in turn for each column, row by row, where i represents the row and j the columns of **C** under consideration. If $C_{ij} = 1$ then we can deduce that the 2 sub-clusters represented by the 2 columns i and j of **C** intersect at the open ball B_j - this is the same as for Method A and Method B above.

Consider column 1 of **C**. Create two sets as follows. The first set n_0 will contain the indices $f \in I$, where $C_{1x} = 0$ $, x \in I$ whilst the second set n_1 will contain the indexes $g \in I$, where $C_{1x} = 1$ $, x \in I$.

For each $g \in n_1$ we look at column g and check each row $g \in n_0$ to see if $C_{fg} = 1$. For each case, where $C_{fg} = 1$ add the current value of f to n_1, set $C_{1g} = 1$ and delete f from n_0. Repeat for each $f \in n_0$ then delete g from n_1 and discard column g of **C**.

Repeat this procedure for column 2 and subsequent columns of **C** ignoring previously discarded columns.

The un-discarded columns of **C** represent the clusters of open balls within the metric space, the presence of an open ball in a cluster is indicated where $C_{ij} = 1$ for a particular value of column j.

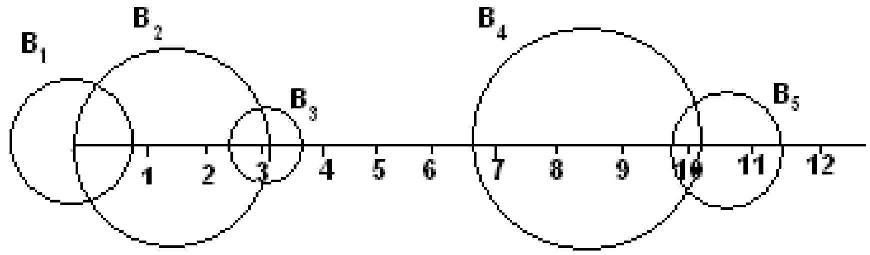

Figure 1: Example of two clusters formed in $\mathbb{R}^2$ from five open balls

8. Example Used to Demonstrate the Three Clustering Methods in Operation

To assist in the visualization of the process of building the clusters we now introduce an example. We could choose any set in $\mathbb{R}^n$ with any metric, as our metric space but concentrate for simplicity on the metric space $(\mathbb{R}^2, d)$ where d is equal to the Euclidean metric in $\mathbb{R}^2$. The indexing set $I = \{1, 2, \ldots, 5\}$. Consider Figure 1. It consists of two clusters: O_1 is comprised of the open balls $B_1((0,0), 0.8)$, $B_2((1.5,0), 1.5)$, $B_3((3,0), 0.6)$ whilst O_2 is comprised of the open balls $B_4((8.5,0), 1.75)$, $B_5((10.5,0), 0.75)$. We will use the clustering method to derive the fact that only 2 clusters exist and find the open balls associated with each cluster.

Carrying out the various distance evaluations between the five points x_1 to x_5 at the centre of each of the open balls B_1 to B_5 gives us a set of values that we can present in the form of a lower triangular matrix (LTM).

d	x_1	x_2	x_3	x_4	x_5
x_1	0	0	0	0	0
x_2	1.5	0	0	0	0
x_3	3	1.5	0	0	0
x_4	8.5	7	4	0	0
x_5	10.5	9	7.5	2	0

From the LTM we can construct the full distance matrix $\mathbf{D}$ by symmetry about the leading diagonal of the LTM.

$$\mathbf{D} = \begin{bmatrix} 0 & 1.5 & 3 & 8.5 & 10.5 \\ 1.5 & 0 & 1.5 & 7 & 9 \\ 3 & 1.5 & 0 & 4 & 7.5 \\ 8.5 & 7 & 4 & 0 & 2 \\ 10.5 & 9 & 7.5 & 2 & 0 \end{bmatrix}.$$

From this distance matrix and the radii r_1 to r_5 of our open balls B_1 to B_5 we can deduce whether or not 2 open balls intersect.

We do this by taking the first column of the distance matrix. The first column of D has the open ball B_1 associated with it, the second column of D has the open ball B_2 associated with it, etc. Now each of the 5 rows of each column also has an open ball $B_j, j = 1 \ldots 5$ associated with it. Now if $D_{ij} < (r_i + r_j)$ this corresponds mathematically with $B_i \cap B_j \neq \emptyset \rightarrow C_{ij} = 1$ and we can construct the matrix $\mathbf{C}$ from the C_{ij} $i, j = 1, \ldots, 5$ comparisons carried out. For our example,

$$\mathbf{C} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

9. Clustering Using Method A

Using the example outlined above we can now construct our final set of clusters. Compare each row of column C_{j1} with each row of C_{j2} in turn. Row 1 of both columns is equal to 1, therefore we copy each row of column 2, where $C_{j2} = 1$ to the corresponding row of column 1 and then discard column 2. This gives us the matrix $\mathbf{C_A}$ below

$$\mathbf{C_A} = \begin{bmatrix} 1 & X & 0 & 0 & 0 \\ 1 & X & 1 & 0 & 0 \\ 1 & X & 1 & 0 & 0 \\ 0 & X & 0 & 1 & 1 \\ 0 & X & 0 & 1 & 1 \end{bmatrix}.$$

Using $\mathbf{C_A}$ we now compare the augmented column 1 with column 3 (column 2 having been discarded) and this gives us

$$\mathbf{C_B} = \begin{bmatrix} 1 & X & X & 0 & 0 \\ 1 & X & X & 0 & 0 \\ 1 & X & X & 0 & 0 \\ 0 & X & X & 1 & 1 \\ 0 & X & X & 1 & 1 \end{bmatrix}.$$

We then compare column 1 of $\mathbf{C_B}$ with column 4 of $\mathbf{C_B}$. None of the 5 rows $i = 1 \ldots 5$ of $C_{Bi1} = C_{Bi4} = 1$ therefore, we then compare column 1 of $\mathbf{C_B}$ with column 5 of $\mathbf{C_B}$. Again we find that none of the 5 rows $i = 1 \ldots 5$ of $C_{Bi1} = C_{Bi5} = 1$. At this stage we have completed all the possible evaluations of column 1 with the other columns, so we carry on and consider the next available column, column 4.

We compare column 4 of $\mathbf{C_B}$ with column 5 of $\mathbf{C_B}$. Row 4 of each column is equal to 1, therefore we copy each row of column 5, where $C_{Bi5} = 1$ to the corresponding row of column 4 and then discard column 5. This gives us the matrix

$$\mathbf{C_C} = \begin{bmatrix} 1 & X & X & 0 & X \\ 1 & X & X & 0 & X \\ 1 & X & X & 0 & X \\ 0 & X & X & 1 & X \\ 0 & X & X & 1 & X \end{bmatrix}.$$

All possible column comparisons of the original matrix $\mathbf{C}$ are now complete and we are left with 2 columns which correspond with the 2 clusters as is seen in matrix $\mathbf{C_C}$, cluster $O_1 = B_1 \cup B_2 \cup B_3$ and cluster $O_2 = B_4 \cup B_5$.

10. Clustering Using Method B

We now take the same set of clusters as used in the previous example, and utilise a different method of building the sub-clusters into the final set of clusters. We will use clustering method B to derive the fact that only 2 clusters exist and find the open balls associated with each cluster.

The construction of the LTM, matrix $\mathbf{D}$ and matrix $\mathbf{C}$ are exactly the same as for the first example.

We can now construct our final set of clusters. Check each row of column 1 of $\mathbf{C}$ to see whether or not it is equal to 1. If it is equal to 1 then copy column

i of $\mathbf{C}$ to column 1, row by row (where i is the row number of column 1 where $C_{i1} = 1$), and then discard column i.

This gives us $\mathbf{C}_A$ from $\mathbf{C}$ as follows

$$\mathbf{C_A} = \begin{bmatrix} 1 & X & 0 & 0 & 0 \\ 1 & X & 1 & 0 & 0 \\ 1 & X & 1 & 0 & 0 \\ 0 & X & 0 & 1 & 1 \\ 0 & X & 0 & 1 & 1 \end{bmatrix}.$$

Using $\mathbf{C_A}$ we now compare the augmented column 1 row by row to check if equal to 1, and hence transfer column 3 to column 1 (column 2 having been discarded), and then discard column 3. This gives us

$$\mathbf{C_B} = \begin{bmatrix} 1 & X & X & 0 & 0 \\ 1 & X & X & 0 & 0 \\ 1 & X & X & 0 & 0 \\ 0 & X & X & 1 & 1 \\ 0 & X & X & 1 & 1 \end{bmatrix}.$$

No further building of the sub-clusters is possible for column 1. Therefore we go to the next column that has not been discarded (column 4) and repeat the process. This gives us the matrix

$$\mathbf{C_C} = \begin{bmatrix} 1 & X & X & 0 & X \\ 1 & X & X & 0 & X \\ 1 & X & X & 0 & X \\ 0 & X & X & 1 & X \\ 0 & X & X & 1 & X \end{bmatrix}.$$

All possible column comparisons of the original matrix $\mathbf{C}$ are now complete and we are left with 2 columns which correspond with the 2 clusters as is seen in matrix $\mathbf{C_C}$, cluster $O_1 = B_1 \cup B_2 \cup B_3$ and cluster $O_2 = B_4 \cup B_5$.

11. Clustering Using Method C

Again using the same set of clusters as used in the previous 2 examples, and carrying out a different method of building the sub-clusters into the final set of clusters. We will use clustering method C to derive the fact that only 2 clusters exist and find the open balls associated with each cluster.

The construction of the LTM, matrix **D** and matrix **C** are exactly the same as for the first 2 methods.

Consider column 1 of **C**. After row by row comparison of column 1 the set $n_0 = \{3, 4, 5\}$ whilst the set $n_1 = \{2\}$ (*1* is not an element of n_1 as $i \neq j,\ i, j \in I$).

From the set n_1 we have index column number 2. Comparing each of the row index elements in the set n_0 we see that $C_{23} = 1$, therefore we make $C_{13} = 1$, add row index element 3 to the set n_1 and delete 3 from the set n_0. This leaves us with

$$\mathbf{C_A} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

No further rows of n_0 in column 2 are equal to 1 therefore we discard column 2 and delete 2 from n_1. Thus

$$\mathbf{C_B} = \begin{bmatrix} 1 & X & 0 & 0 & 0 \\ 1 & X & 1 & 0 & 0 \\ 1 & X & 1 & 0 & 0 \\ 0 & X & 0 & 1 & 1 \\ 0 & X & 0 & 1 & 1 \end{bmatrix}.$$

We are still examining column 1 and its associated sets $n_1 = \{3\}$, and $n_0 = \{4, 5\}$. Neither of the the two rows 4 and 5 of column 3 are equal to 1, therefore we discard column 3 and have $n_1 = \emptyset$. This gives

$$\mathbf{C_C} = \begin{bmatrix} 1 & X & X & 0 & 0 \\ 1 & X & X & 0 & 0 \\ 1 & X & X & 0 & 0 \\ 0 & X & X & 1 & 1 \\ 0 & X & X & 1 & 1 \end{bmatrix}.$$

As $n_1 = \emptyset$ we choose the next column that has not been discarded (column 4) and construct the sets $n_1 = 5$ and $n_0 = 1, 2, 3$ for it. None of the row index elements of n_0 in column 5 are equal to 1, therefore we delete column 5 from the set n_1 and discard column 5 of the matrix **C** which leaves us with

$$\mathbf{C_D} = \begin{bmatrix} 1 & X & X & 0 & X \\ 1 & X & X & 0 & X \\ 1 & X & X & 0 & X \\ 0 & X & X & 1 & X \\ 0 & X & X & 1 & X \end{bmatrix}.$$

No further un-discarded columns are present in **C** thus we are left with 2 columns representing our two original clusters.

12. Outline Flow of the Partition Clustering Methods

12.1. Method A

From the proximity matrix, starting with the first column we can compare each row in this column with each row of the other columns, row by row, column by column. Once all comparisons are complete we then consider the next column after the first and compare it with each of the subsequent columns, again row by row, and so on.

Step 1. Choose the first (column j) and second (column k) columns, set $n =$ the total number of open balls.
Step 2. Set the row counter i equal to 1.
Step 3. If ($c_{ij} \neq 1$ and $c_{ik} \neq 1$) then go to Step 7.
    Step 4. Whilst $i \leq n$, $c_{ij} = \max(c_{ik}, c_{ij})$ else go to Step 6.
    Step 5. $i = i + 1$ go to Step 4.
    Step 6. If $j < n$ then discard column k, $k = j + 1$ go to Step 2, otherwise go to Step 10.
    Step 7. If $i < n$ then $i = i + 1$ and go to Step 3.
    Step 8. If $k < n$ then $k = k + 1$ and go to Step 2.
    Step 9. If $j < (n - 1)$ then $j = j + 1$, $k = j + 1$ and go to Step 2.
Step 10. End

12.2. Method B

Alternatively for Method B, again using the proximity matrix and starting with the column 1 we have:
Step 1. Column $j = 1$, set $n =$ the total number of open balls.
Step 2. Row count $i = 1$.
Step 3. If $C_{ij} = 1$ and $i \neq j$ go to Step 7.
Step 4. If $i < n$, then $i = i + 1$ and goto Step 3.
Step 5. If $j < n - 1$, then $j = j + 1$ and go to Step 2.
Step 6. End.
    Step 7. $k = i, m = 1$
    Step 8. $c_{mj} = \max(c_{mk}, c_{mj})$
    Step 9. If $m < n$, $m = m + 1$ and go to Step 8.

Step 10. Discard column k and go to Step 2.

12.3. Method C

This time for Method C, again using the proximity matrix and starting with the column 1 we have:

Step 1. Column $j = 1$, set $n =$ the total number of open balls.

Step 2. If column j has been discarded go to Step 9.

Step 3. Row count $i = 1$, $n_1 = \emptyset$ and $n_0 = \emptyset$.

Step 4. If $C_{ij} = 1$ and $i \neq j$ go to Step 6.

Step 5. Add i to the set n_0 go to Step 7.

Step 6. Add i to the set n_1

Step 7. If $i < n$, $i = i + 1$ and go to Step 4.

Step 8. For each $f \in n_1$ and $g \in n_0$, if $C_{fg} = 1$ then $C_{gj} = 1$, add the element g to n_1, discard column element (and column) f and repeat Step 8.

Step 9. If $j < n - 1$, $j = j + 1$ and go to Step 2.

Step 10. End.

13. Density of Clusters within the Metric Space

Various methods can be employed to derive the density of the clusters within the metric space. The approach used in this paper of constructing a distance matrix **D** and a proximity matrix **C** allows density calculations to be carried out easily.

The columns of the proximity matrix **C**, before it is merged, indicate those open balls that are clustered immediately adjacent to the open ball associated with the column currently under consideration. Knowing this and the radius of each open ball we can calculate the density of the clusters (represented by each column of the **C** before it is merged).

After the proximity matrix **C** is merged its columns represent the clusters of open balls within the whole metric space. Again we can work out the density of each cluster using an appropriate method, either utilizing the density of sub-clusters previously calculated, or by using the radius of the open balls associated with each cluster.

k	Number of non-isomorphic trees	Number of non-isomorphic rooted trees
1	1	1
2	1	1
3	1	2
4	2	4
5	3	9
6	6	20
7	11	48
8	23	115
9	47	286
10	106	719
11	235	1,842
12	551	4,776
13	1,301	12,486
14	3,159	32,973
15	7,741	87,811
16	19,320	235,381
17	48,629	634,847
18	123,867	1,721,159
19	317,955	4,688,676
20	823,065	12,826,228

Table 1: The numbers of non isomorphic trees and rooted trees for $k \leq 20$

14. Application Area – Egress Complexity

An immediate area of application for this algorithm lies in the field of egress complexity, which is concerned with the assessment of egress capability from buildings. Essentially, a typical floorplan is considered as a series of nodes and arcs, the nodes representing the centroid of habitable rooms and the arcs representing the available egress pathways to an exit or exits. There are two approaches to the problem, one based on metric arcs and another, which considers the arrangement as a collection of information steps between nodes, represented by a lower semi-lattice. The former is described in Kisko et al [5] and the latter is fully documented in Livesey [6]. We are primarily concerned with an application of the algorithm to a hybrid concept, which embraces both the metric and non-metric approaches.

When seeking a normalised value of egress complexity the modeller is faced with the task of calculating the complexity of all of the corresponding non-isomorphic k-node rooted trees. The following table [7] gives some indication of the magnitude of the problem when the number of compartments is as few as $k = 20$.

Clearly, as the number of compartments increase beyond 20 the problem becomes even more daunting. For this reason it is proposed that the spatial layout should be partitioned into clusters of order ≤ 10 open balls, as described above. Each cluster would have a non-metric complexity value – effectively a neighbourhood complexity value [6] and the cluster of clusters would have a global non-metric complexity value, which would be indexed to the floorplan. At this time it has not been decided how best to arrive at the open ball radius so that the partitioning constraint mentioned above is maintained.

Pollock et al [8] studied the problem of interfacing architectural CAD data with KBS technology in order to evaluate floorplans from differing perspectives. It is proposed to adapt this work to the determination of intersecting open balls in order to characterise the clusters so that the complexity can be calculated.

15. Conclusion

The use of clusters in metric spaces to minimize search times is not a new idea. This paper has described 3 methods for the construction of clusters within a metric space in which all open balls use the same metric but each open ball can have a different radius associated with it. The application of this approach is to pre-process data into clusters in those application areas where the data from which the clusters are built changes infrequently. This allows rapid comparison of the clustered data with a new set of data against which it is to be compared, thus minimizing search times. Additionally the method of construction of the clusters within the metric space allows the densities of the clusters and sub-clusters to be easily calculated using a simple method. The methods presented have an immediate application to the calculation of egress complexity and it is believed that potentially, many more application areas exist.

References

[1] A.K. Jain, R.C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, New Jersey, USA (1988).

[2] W.A. Sutherland, *Introduction to Metric and Topological Spaces*, Clarendon Press, Oxford (1975).

[3] E. Chávez, E. Navarro, G. Baeza-Yates, R. Marroquín, Searching in metric spaces, *Technical Report*, tr/dcc-99-3, Dept. of Computer Science, Univ. of Chile (1999); To Appear in: *ACM Computing Surveys*; Available in: http:/citeseer.ist.psu.edu/avez99searching.html, University of Chile.

[4] S. Brin, Near neighbor search in large metric spaces, In: *Proceedings of the 21-th International Conference on Very Large Data Bases*, Morgan Kaufmann Publishers Inc. (1995), 574-584.

[5] T.M. Kisko, R.L. Francis, C.R. Nobel, *EVACNET4 User's Guide*, University of Florida (1998).

[6] G.E. Livesey, *Advancing Egress Complexity in Support of Rule-Based Evacuation Modelling*, PhD Thesis, University of Ulster (2003).

[7] F. Harary, *Graph Theory*, Addison-Wesley (1969).

[8] A.J. Pollock, *Intelligent Interpretation of CAD Drawings for Building Evaluation*, PhD Thesis, University of Ulster (1995).