Hierarchy of Governance Stations (Cities) Determination Using Agglomerative Hierarchical Clustering

Chandra Kumar Nimmana[1], G Prasanna Kumar[2], Manoj Kumar Pakki[3]

[1]*M.Tech Structures Student, Dept. of Civil Engineering, Aditya Institute of Technology & Management*

[2,3]*Assistant Professor, Dept. of Civil Engineering, Aditya Institute of Technology & Management*

Abstract- **In governmental administration from the Centre to the State and State to the Districts and Districts to the villages, we often find hindrances due to non-optimal structural location of governmental offices. The inconvenient distances sometimes pose a bottleneck problem to smooth administration. However such problems can be avoided if we choose the Structural Administrative Offices location optimally. This research is to find the Structural Coordinate Determination For Optimal Administrative Offices Stationing Using Agglomerative Hierarchical Clustering. Needless to mention, such Hierarchy found is to be mimicked as the Hierarchy for Governmental Administration. From the Cluster Dendrogram, we can note the hierarchical gradation based on the distance between the cities. Such a hierarchical gradation can be used for efficient governance.**

Index Terms- **Hierarchical Agglomerative Clustering, Latitude, Longitude**

I. INTRODUCTION

In the following paragraphs of this section, the author has detailed literature survey on Hierarchical Agglomerative Clustering with its varied applications.

The world is becoming linked more and more. A shift of researching focus can be observed recently, from "city as a system" to "systems of cities," given the context of fast-changing communicating technologies such as high-speed railways (physical) as well as social media over the internet (nonphysical). Flows play essential roles for a city network, indicating the trends of position and functions within the network. In [1], the authors adopt new type of aggregated positioning data of massive internet users in China to explore the spatial patterns of cities during the Spring Festival in 2015. By introducing new clustering algorithm highlighting spatial constraints, models output hierarchic results with vary regional zones containing different number of cities. The higher layer of results with less members is not similar to the conventional delineation according to the conditions of physical and economic geography of China. Nevertheless, the very differences suggest hidden forces driving cities connected intensely across the administrative boundaries such as sharing mutual regional cultures or employment markets. These facts grounded for a general picture for the study on polycentric urban regions over the whole national territory.

It has often been asserted that since hierarchical clustering algorithms require pairwise interobject proximities, the complexity of these clustering procedures is at least O(N2). Recent work has disproved this by incorporating efficient nearest neighbour searching algorithms into the clustering algorithms. A general framework for hierarchical, agglomerative clustering algorithms is discussed in [2], which opens up the prospect of much improvement on current, widely-used algorithms. This 'progress report' details new algorithmic approaches in this area, and reviews recent results.

The Traveling Salesman Problem(TSP) is one of the most intensively studied problem in computational mathematics. To solve this problem a number of algorithms have been developed using genetic algorithms. But these algorithms are not so suitable for solving large-scale TSP. In [3], the authors proposes a new solution for TSP using hierarchical clustering and genetic algorithm.

The problem of allocating resources in spatial locations such as within an urban city or large regions in geographical sense has attracted much research efforts recently. Some applications include but not limit to city-planning for examples of building patrol stations in a city, establishing medical clinics or schools in a town, deploying guards for security patrol in a zone, and budgeting on the quantity of street lamps to lit up an urban area. These problems are generalized as spatial resource allocation, where they commonly share the characteristics of meeting certain demands by a limited amount of resources. The demands are usually distributed, unevenly in a confined spatial area. Traditionally clustering algorithms in data mining were used to solve these problems. In[4] the authors proposed a grid-based hierarchical clustering approach that was designed specifically for this kind of resource allocation decision-support. The grid-based feature makes the data extraction process which is usually from maps efficient. The hierarchy of clusters as outputs provides an advantage over normal clustering techniques because the resultant clusters can be zoomed in or out in different resolutions or abstractions at will.

For 3D global visualization systems such as Google Earth, it is important to be able to render city-sized collections of relatively simple building models at fast speeds without losing spatial coherence. Since traditional mesh simplification algorithms are not designed for collections of simple models, in [5], the authors introduce a method of simplification through merging of similar objects. We incorporate the concept of "urban legibility" from architecture and city-planning as a guideline for simplifying city models. The authors algorithm can be broken down into five steps. Hierarchical clustering, cluster merging, polyline simplification, and hierarchical texturing are performed during pre-processing, while at runtime, the levels-of-detail (LOD) process selects the appropriate models to render. It is authors belief that many applications can benefit from their algorithm. Google Earth (and other 3D geographical information systems) as well as any spatial data visualization applications (including scatter plots) can all use logical, simplified clusters to represent large amounts of spatial information.

Most of the data collected by organizations and firms contains multi-attribute and temporal data. Identifying temporal relationships (e.g., trends) in data constitutes an important problem that is relevant in many business and academic settings. Data mining techniques are used to discover patterns in such data. Temporal data can take many forms, most commonly being general transactional (multi)attribute-value data, for which time series or sequence analysis methods are not particularly well suited. In [6] the authors present the clustering algorithm with performance and implementation of dataset based on distances in miles between US cities.

Clustering is a common technique for statistical data analysis, which is used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics. Clustering is the process of grouping similar objects into different groups, or more precisely, the partitioning of a data set into subsets, so that the data in each subset according to some defined distance measure. [7] covers about clustering algorithms, benefits and its applications. It concludes by discussing some limitations.

Hierarchical clustering algorithms are typically more effective in detecting the true clustering structure of a data set than partitioning algorithms. However, hierarchical clustering algorithms do not actually create clusters, but compute only a hierarchical representation of the data set. This makes them unsuitable as an automatic pre-processing step for other algorithms that operate on detected clusters. This is true for both dendrograms and reachability plots, which have been proposed as hierarchical clustering representations, and which have different advantages and disadvantages. In [8], the authors first investigate the relation between dendrograms and reachability plots and introduce methods to convert them into each other showing that they essentially contain the same information. Based on reachability plots, the authors then introduce a technique that automatically determines the significant clusters in a hierarchical cluster representation. This makes it for the first time possible to use hierarchical clustering as an automatic pre-processing step that requires no user interaction to select clusters from a hierarchical cluster representation

The visual appearance of city neighborhoods helps us to mentally map urban spaces. For instance, from the visual features of a city or neighborhood, we gain perspectives on local identity as might be described by their functions, demographics, or affluence. An effective way to summarize and present this information would be useful, e.g., for urban design and planning. The authors explore whether these perspectives can be automatically learned from street-level imagery using a deep neural network and build a visual analytics tools to explore what is learned. Starting with a dense geo-sampling of city Google Street View data, the authors train a neural network to learn visual features. Then, the authors cluster these features using unsupervised learning to build a similarity hierarchy of visual appearance. Existing approaches for exploring this kind of geographically-embedded cluster data often have difficulty in addressing the need to compare across both the visual hierarchy and the geography of the different neighborhoods. To improve this situation, the authors develop a visualization scheme which allows users to keep track of both the geographical and semantic interpretations of the data. In doing so, the authors aim to provide an exploration tool to aid in the visual study of urban environments.

II. HIERARCHICAL AGGLOMERATAIVE CLUSTERING ALGORITHM OVERVIEW

Hierarchical Agglomerative Clustering Method

In datamining and statistics, hierarchicalclustering (al so called hierarchical cluster analysis or HCA) is a method of cluster analysis which seeks to build a hierarchy of clusters. Agglomerative Clustering is a type of Hierarchical Clustering.

Agglomerative: This is a "bottom up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
In general, the merges and splits are determined in a greedy manner. The results of hierarchical clustering are usually presented in a dendrogram.
It Produces a set of nested clusters organized as a hierarchical tree. It Can be visualized as a dendrogram. The Clusters are represented by a tree-like diagram that records the sequences of merges or splits which is known as a Dendrogram.

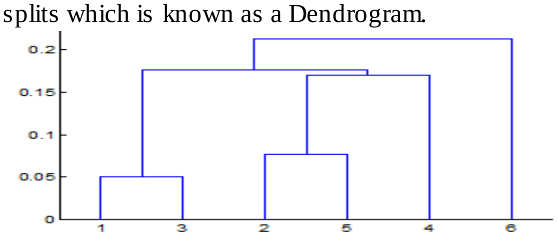

Fig 1- Example of Dendrogram for the dataset of first 6 numbers

Strengths of Hierarchical Clustering

There are no assumptions on the number of clusters. Any desired number of clusters can be obtained by 'cutting' the dendogram at the proper level. Hierarchical clusterings may correspond to meaningful taxonomies. For Example, in biological sciences (e.g., phylogeny reconstruction, etc), web (e.g., product catalogs) etc
The Two main types of hierarchical clustering are:

Agglomerative:

We start with the points as individual clusters. At each step, merge the closest pair of clusters until only one cluster (or **k** clusters) left.

Divisive:

We start with one, all-inclusive cluster. At each step, split a cluster until each cluster contains a point (or there are **k** clusters). Traditional hierarchical algorithms use a similarity or distance matrix. Here, we Merge or split one cluster at a time.

Complexity of hierarchical clustering

A Distance matrix is used for deciding which clusters to merge/split. It is at least quadratic in the number of data points. It is not usable for large datasets

Agglomerative clustering algorithm

It is the most popular hierarchical clustering technique

Basic algorithm

1. Compute the distance matrix between the input data points
2. Let each data point be a cluster
3. Repeat
4. Merge the two closest clusters
5. Update the distance matrix
6. Until only a single cluster remains

The Key operation is the computation of the distance between two clusters. Different definitions of the distance between clusters lead to different algorithms

Distance between two clusters

Each cluster is a set of points.
We define distance between two sets of points in the following fashions:

1. Single-link distance between clusters C_i and C_j is the minimum distance between any object in C_i and any object in C_j

The distance is defined by the two most similar objects

$$D_{sl}\left(C_i, C_j\right) = \min_{x,y}\left\{d(x,y) \middle| x \in C_i, y \in C_j\right\}$$

It is determined by one pair of points, i.e., by one link in the proximity graph.

2. Complete-link distance between clusters C_i and C_j is the maximum distance between any object in C_i and any object in C_j

The distance is defined by the two most dissimilar objects

$$D_{cl}\left(C_i, C_j\right) = \max_{x,y}\left\{d(x,y) \middle| x \in C_i, y \in C_j\right\}$$

3. Group average distance between clusters C_i and C_j is the average distance between any object in C_i and any object in C_j

$$D_{avg}\left(C_i, C_j\right) = \frac{1}{\left|C_i\right| \times \left|C_j\right|} \sum_{x \in C_i, y \in C_j} d(x,y)$$

Centroid distance between clusters C_i and C_j is the distance between the centroid r_i of C_i and the centroid r_j of C_j

$$D_{centroids}\left(C_i, C_j\right) = d(r_i, r_j)$$

Ward's distance between clusters C_i and C_j is the difference between the total within cluster sum of squares for the two clusters separately, and the within cluster sum of squares resulting from merging the two clusters in cluster C_{ij}

$$D_w\left(C_i, C_j\right) = \sum_{x \in C_i}\left(x - r_i\right)^2 + \sum_{x \in C_j}\left(x - r_j\right)^2 - \sum_{x \in C_{ij}}\left(x - r_{ij}\right)^2$$

r_i: centroid of C_i
r_j: centroid of C_j
r_{ij}: centroid of C_{ij}
Ward's distance for clusters is
-Similar to group average and centroid distance
-Less susceptible to noise and outliers
-Biased towards globular clusters
-Hierarchical analogue of k-means
-It can be used to initialize k-means

Hierarchical Clustering: Time and Space requirements

For a dataset X consisting of n points
$O(n^2)$ space; it requires storing the distance matrix
$O(n^3)$ time in most of the cases
There are **n** steps and at each step the size $\mathbf{n^2}$ distance matrix must be updated and searched
Complexity can be reduced to $\mathbf{O(n^2 \log(n))}$ time for some approaches by using appropriate data structures

III. HIERARCHICAL AGGLOMERATAIVE CLUSTERING RESULTS

Data Pre-Processing

We have considered 104 cities in the former Andhra Pradesh State and recorded their Latitudes and Longitudes. Using the formulae* stated below, we computed the Distances between each of the 104 cities to each of the 104 cities. That is, we have calculated 104x104=10,016 distances. We have used R program to compute the same. These distances are detailed in the All Distances File given at https://docs.google.com/spreadsheets/d/1WCxwGs2T-VrrMPDvwEvsGDxwapyfGogrWUzfwAaIKVQ/edit#gid=130852057

Latitudes & Longitudes Of Former Andhra Pradesh State In India

(1, 'Adilabad', '19.68 N', '78.53 E', 'Andhra Pradesh'),
(2, 'Adoni', '15.63 N', '77.28 E', 'Andhra Pradesh'),
(3, 'Alwal', '17.50 N', '78.54 E', 'Andhra Pradesh'),
(4, 'Anakapalle', '17.69 N', '83.00 E', 'Andhra Pradesh'),
(5, 'Anantapur', '14.70 N', '77.59 E', 'Andhra Pradesh'),
(6, 'Bapatla', '15.91 N', '80.47 E', 'Andhra Pradesh'),
(7, 'Belampalli', '19.06 N', '79.49 E', 'Andhra Pradesh'),
(8, 'Bhimavaram', '16.55 N', '81.53 E', 'Andhra Pradesh'),
(9, 'Bhongir', '17.52 N', '78.88 E', 'Andhra Pradesh'),
(10, 'Bobbili', '18.57 N', '83.37 E', 'Andhra Pradesh'),
(11, 'Bodhan', '18.66 N', '77.88 E', 'Andhra Pradesh'),
(12, 'Chilakalurupet', '16.10 N', '80.16 E', 'Andhra Pradesh'),
(13, 'Chinna Chawk', '14.47 N', '78.83 E', 'Andhra Pradesh'),
(14, 'Chirala', '15.84 N', '80.35 E', 'Andhra Pradesh'),
(15, 'Chittur', '13.22 N', '79.10 E', 'Andhra Pradesh'),
(16, 'Cuddapah', '14.48 N', '78.81 E', 'Andhra Pradesh'),

(17, 'Dharmavaram', '14.42 N', '77.71 E', 'Andhra Pradesh'),
(18, 'Dhone', '15.42 N', '77.88 E', 'Andhra Pradesh'),
(19, 'Eluru', '16.72 N', '81.11 E', 'Andhra Pradesh'),
(20, 'Gaddiannaram', '17.36 N', '78.52 E', 'Andhra Pradesh'),
(21, 'Gadwal', '16.23 N', '77.80 E', 'Andhra Pradesh'),
(22, 'Gajuwaka', '17.70 N', '83.21 E', 'Andhra Pradesh'),
(23, 'Gudivada', '16.44 N', '81.00 E', 'Andhra Pradesh'),
(24, 'Gudur', '14.15 N', '79.84 E', 'Andhra Pradesh'),
(25, 'Guntakal', '15.18 N', '77.37 E', 'Andhra Pradesh'),
(26, 'Guntur', '16.31 N', '80.44 E', 'Andhra Pradesh'),
(27, 'Hindupur', '13.83 N', '77.48 E', 'Andhra Pradesh'),
(28, 'Hyderabad', '17.40 N', '78.48 E', 'Andhra Pradesh'),
(29, 'Jagtial', '18.80 N', '78.91 E', 'Andhra Pradesh'),
(30, 'Kadiri', '14.12 N', '78.16 E', 'Andhra Pradesh'),
(31, 'Kagaznagar', '19.34 N', '79.48 E', 'Andhra Pradesh'),
(32, 'Kakinada', '16.96 N', '82.24 E', 'Andhra Pradesh'),
(33, 'Kallur', '15.69 N', '77.77 E', 'Andhra Pradesh'),
(34, 'Kamareddi', '18.32 N', '78.35 E', 'Andhra Pradesh'),
(35, 'Kapra', '17.37 N', '78.48 E', 'Andhra Pradesh'),
(36, 'Karimnagar', '18.45 N', '79.13 E', 'Andhra Pradesh'),
(37, 'Karnul', '15.83 N', '78.03 E', 'Andhra Pradesh'),
(38, 'Kavali', '14.92 N', '79.99 E', 'Andhra Pradesh'),
(39, 'Khammam', '17.25 N', '80.15 E', 'Andhra Pradesh'),
(40, 'Kodar', '16.98 N', '79.97 E', 'Andhra Pradesh'),
(41, 'Kondukur', '15.22 N', '79.91 E', 'Andhra Pradesh'),
(42, 'Koratla', '18.82 N', '78.72 E', 'Andhra Pradesh'),
(43, 'Kottagudem', '17.56 N', '80.64 E', 'Andhra Pradesh'),
(44, 'Kukatpalle', '17.49 N', '78.41 E', 'Andhra Pradesh'),
(45, 'Lalbahadur Nagar', '17.43 N', '78.50 E', 'Andhra Pradesh'),
(46, 'Machilipatnam', '16.19 N', '81.14 E', 'Andhra Pradesh'),
(47, 'Mahbubnagar', '16.74 N', '77.98 E', 'Andhra Pradesh'),
(48, 'Malkajgiri', '17.55 N', '78.59 E', 'Andhra Pradesh'),
(49, 'Mancheral', '18.88 N', '79.45 E', 'Andhra Pradesh'),
(50, 'Mandamarri', '18.97 N', '79.47 E', 'Andhra Pradesh'),
(51, 'Mangalagiri', '16.44 N', '80.56 E', 'Andhra Pradesh'),
(52, 'Markapur', '15.73 N', '79.28 E', 'Andhra Pradesh'),
(53, 'Miryalaguda', '16.87 N', '79.57 E', 'Andhra Pradesh'),
(54, 'Nalgonda', '17.06 N', '79.26 E', 'Andhra Pradesh'),
(55, 'Nandyal', '15.49 N', '78.48 E', 'Andhra Pradesh'),
(56, 'Narasapur', '16.45 N', '81.70 E', 'Andhra Pradesh'),
(57, 'Narasaraopet', '16.24 N', '80.04 E', 'Andhra Pradesh'),
(58, 'Nellur', '14.46 N', '79.98 E', 'Andhra Pradesh'),
(59, 'Nirmal', '19.12 N', '78.35 E', 'Andhra Pradesh'),
(60, 'Nizamabad', '18.68 N', '78.10 E', 'Andhra Pradesh'),
(61, 'Nuzvid', '16.78 N', '80.85 E', 'Andhra Pradesh'),
(62, 'Ongole', '15.50 N', '80.05 E', 'Andhra Pradesh'),
(63, 'Palakollu', '16.52 N', '81.75 E', 'Andhra Pradesh'),
(64, 'Palasa', '18.77 N', '84.42 E', 'Andhra Pradesh'),
(65, 'Palwancha', '17.60 N', '80.68 E', 'Andhra Pradesh'),
(66, 'Patancheru', '17.53 N', '78.27 E', 'Andhra Pradesh'),
(67, 'Piduguralla', '16.48 N', '79.90 E', 'Andhra Pradesh'),
(68, 'Ponnur', '16.07 N', '80.56 E', 'Andhra Pradesh'),
(69, 'Proddatur', '14.73 N', '78.55 E', 'Andhra Pradesh'),
(70, 'Qutubullapur', '17.43 N', '78.47 E', 'Andhra Pradesh'),
(71, 'Rajamahendri', '17.02 N', '81.79 E', 'Andhra Pradesh'),
(72, 'Rajampet', '14.18 N', '79.17 E', 'Andhra Pradesh'),
(73, 'Rajendranagar', '17.29 N', '78.39 E', 'Andhra Pradesh'),
(74, 'Ramachandrapuram', '17.56 N', '78.04 E', 'Andhra Pradesh'),
(75, 'Ramagundam', '18.80 N', '79.45 E', 'Andhra Pradesh'),

(76, 'Rayachoti', '14.05 N', '78.75 E', 'Andhra Pradesh'),
(77, 'Rayadrug', '14.70 N', '76.87 E', 'Andhra Pradesh'),
(78, 'Samalkot', '17.06 N', '82.18 E', 'Andhra Pradesh'),
(79, 'Sangareddi', '17.63 N', '78.08 E', 'Andhra Pradesh'),
(80, 'Sattenapalle', '16.40 N', '80.18 E', 'Andhra Pradesh'),
(81, 'Serilungampalle', '17.48 N', '78.33 E', 'Andhra Pradesh'),
(82, 'Siddipet', '18.11 N', '78.84 E', 'Andhra Pradesh'),
(83, 'Sikandarabad', '17.47 N', '78.52 E', 'Andhra Pradesh'),
(84, 'Sirsilla', '18.40 N', '78.81 E', 'Andhra Pradesh'),
(85, 'Srikakulam', '18.30 N', '83.90 E', 'Andhra Pradesh'),
(86, 'Srikalahasti', '13.76 N', '79.70 E', 'Andhra Pradesh'),
(87, 'Suriapet', '17.15 N', '79.62 E', 'Andhra Pradesh'),
(88, 'Tadepalle', '16.48 N', '80.60 E', 'Andhra Pradesh'),
(89, 'Tadepallegudem', '16.82 N', '81.52 E', 'Andhra Pradesh'),
(90, 'Tadpatri', '14.91 N', '78.00 E', 'Andhra Pradesh'),
(91, 'Tandur', '17.25 N', '77.58 E', 'Andhra Pradesh'),
(92, 'Tanuku', '16.75 N', '81.69 E', 'Andhra Pradesh'),
(93, 'Tenali', '16.24 N', '80.65 E', 'Andhra Pradesh'),
(94, 'Tirupati', '13.63 N', '79.41 E', 'Andhra Pradesh'),
(95, 'Tuni', '17.35 N', '82.55 E', 'Andhra Pradesh'),
(96, 'Uppal Kalan', '17.38 N', '78.55 E', 'Andhra Pradesh'),
(97, 'Vijayawada', '16.52 N', '80.63 E', 'Andhra Pradesh'),
(98, 'Vinukonda', '16.05 N', '79.75 E', 'Andhra Pradesh'),
(99, 'Visakhapatnam', '17.73 N', '83.30 E', 'Andhra Pradesh'),
(100, 'Vizianagaram', '18.12 N', '83.40 E', 'Andhra Pradesh'),
(101, 'Vuyyuru', '16.37 N', '80.85 E', 'Andhra Pradesh'),
(102, 'Wanparti', '16.37 N', '78.07 E', 'Andhra Pradesh'),
(103, 'Warangal', '18.01 N', '79.58 E', 'Andhra Pradesh'),
(104, 'Yemmiganur', '15.73 N', '77.48 E', 'Andhra Pradesh')

Computation of The Distance Between Two Pairs of Latitude And Longitude

The website https://www.movable-type.co.uk/scripts/latlong.html allows one to calculate the distance between two pairs of latitude and longitude.

Alternately, we have written an R program to compute the same using the formulae

Haversine Formula	$a = Sin^2\left(\frac{\Delta\varphi}{2}\right) + Cos(\varphi_1)Cos(\varphi_2)Sin^2$ where $\Delta\varphi = (\varphi_2 - \varphi_1)$ and $\Delta\lambda = (\lambda_2 - \lambda_1)$
	$c = 2a\tan 2\left(\sqrt{a}, \sqrt{1-a}\right)$
	$d = Rc$
where	φ is Latitude, λ is longitude, R is earth's radius (mean radius = 6,371km); note that angles need to be in radians to pass to trig functions!

Table 1- Formulae for Computing the distance between Latitudes and Longitudes of two different locations

Implementing the Hierarchical Agglomerative Clustering Algorithm

The Agglomerative Hierarchical Clustering Algorithm is used to find the Hierarchical Grading among the 104 cities based on the Nearest Distances. The R program is detailed below:

```
d <- read.csv('distances')
dists <- data.frame(row.names = d$City, Latitude = d$Latitude, Longitude = d$Longitude)
R = 6373.0
radians <- function(deg){
  return(deg*pi/180)
}
dist2 <- function(x = c(lat1,lon1),y = c(lat2,lon2)){
  lat1 = x[1]; lon1 = x[2]; lat2 = y[1]; lon2 = y[2];
  lat1 = radians(lat1)
  lon1 = radians(lon1)
  lat2 = radians(lat2)
  lon2 = radians(lon2)
  dlon = lon2 - lon1
```

```
 dlat = lat2 - lat1
 a = sin(dlat/2)**2 + cos(lat1) * cos(lat2) *
sin(dlon/2)**2
 c = 2 * atan2(sqrt(a), sqrt(1 - a))
   distance = R * c
 return(distance)
}
head(d)
dist2(c(15.63,77.28), c(17.50,78.54))
library(proxy)
all_dists <- proxy::dist(dists, method = dist2)
clusters = hclust(all_dists)
plot(clusters)
ncls = 15
centers = cutree(clusters, ncls)
for(x in c(1:ncls)){
 print(centers[centers == x])
}
```

Results of the Hierarchical Clustering Algorithm

The aforedetailed Agglomerative Hierarchical Clustering yielded the following results: It basically hierarchically graded the 104 cities.

Please see the link below for a clearer view of the Cluster Dendrogram.

https://drive.google.com/file/d/1JGd32P1fHyDGvnJd1R_WFk7f0LYhpGE/view?usp=sharing

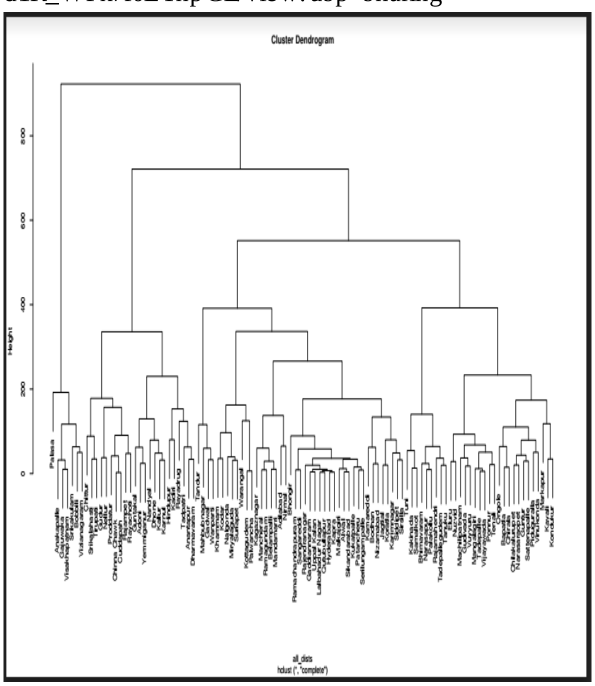

Figure 2 – The Summarized Output of the Hierarchical Agglomerative Clustering Algorithm

CONCLUSIONS

From the Cluster Dendrogram, we can note the hierarchical gradation based on the distance between the cities. Such a hierarchical gradation can be used for efficient governance.

ACKNOWLEDGMENT

The authors are grateful for the encouragement, advice and support offered by Prof G.T. Naidu, (HOD, Department Of Civil Engineering, Aditya Institute Of Technology & Management (AITAM), K. Kotthuru, Tekkali, Andhra Prdesh State India). The authors also are grateful to the former HOD Prof Ch. Kannam Naidu (AITAM). The authors also express their gratitude to Prof Murali Monangi, (formerly at AITAM).

REFERENCES

[1] 'Measuring by Movements: Hierarchical Clustering of Cities in China Based on Aggregated Massive Positioning Data', Dong Li, Menghe Wu, Bingruo Duan, Yuheng Cai, Big Data Support of Urban Planning and Management pp 63-77

[2] 'A Survey of Recent Advances in Hierarchical Clustering Algorithms', F. Murtagh, THE COMPUTER JOURNAL, VOL. 26, NO. 4,1983

[3] 'Solving Travelling Salesman Problem Using Hierarchical Clustering And Genetic Algorithm', S N S Kalyan Bharadwaj.B, Krishna Kishore.G, Srinivasa Rao.V, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (3) , 2011, 1096-1098, ISSN 09759646

[4] 'Grid-Based Hierarchical Clustering For Spatial Resource Allocation', Liyang Liu, Simon Fong and Andy Ip, IADIS International Conference e-Society 2011.

[5] 'Hierarchical Simplification of City Models to Maintain Urban Legibility', Remco Chang, Thomas Butkiewicz, Caroline Ziemkiewicz, Zachary Wartell http://graphics.cs.cmu.edu/nsp/papers/ChangSketch06.pdf

[6] 'Analysis of Dendrogram Tree for Identifying and Visualizing Trends in Multi-attribute

Transactional Data', D.Radha Rani, A.Vini Bharati, P.Lakshmi Durga Madhuri, M.Phaneendra Babu, A.Sravani, International Journal of Engineering Trends and Technology- Volume3Issue1- 2012

[7] 'An Overview On Clustering Methods', T. Soni Madhulatha, IOSR Journal of Engineering Apr. 2012, Vol. 2(4) pp: 719-725

[8] 'Automatic Extraction of Clusters from Hierarchical Clustering Representations', Jörg Sander, Xuejie Qin, Zhiyong Lu, Nan Niu, Alex Kovarsky http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.642.4559&rep=rep1&type=pdf

[9] 'Hierarchical Visual Feature Analysis for City Street View Datasets', Lezhi Li, James Tompkin, Panagiotis Michalatos, and Hanspeter Pfister https://vadl2017.github.io/paper/vadl_0106-paper.pdf

[10] https://www.ibm.com/support/knowledgecenter/en/SSLVMB_22.0.0/com.ibm.spss.statistics.help/spss/base/idh_clus.htm