Geometric Embedding of Road distance as a Metric in Spatial Analysis

A Project Report

submitted by

SHASHWATHY KANNAN

in partial fulfilment of requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS

MAY 2023

THESIS CERTIFICATE

This is to certify that the thesis titled **Geometric Embedding of Road distance as a Metric in Spatial Analysis**, submitted by **Shashwathy Kannan**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology**, is a bona fide record of the research work done by her under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof
N S Narayanaswamy
Professor
Dept. of CSE
IIT-Madras, 600 036

Place: Chennai

Date: May 15, 2023

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere thanks to my guide, Prof. Narayanaswamy, for his exceptional guidance, encouragement, and expertise throughout my project journey. His insights and feedback have been instrumental in shaping my research methodology, analysis, and interpretation of findings.

I also owe a debt of gratitude to my parents and sister for their unwavering support and encouragement throughout my academic pursuits. Their constant motivation and belief in me have been a source of strength.

I would also like to acknowledge my colleague, Manishnantha (CS19B031), for engaging in meaningful discussions in the field of logistics. Additionally, I am grateful to Madhav Mittal (CS19B029), for his prior work in the area of routing algorithms in the logistics lab, which has provided a strong foundation for my research.

ABSTRACT

KEYWORDS: Vehicle Routing Problem ; Inverse Map Projection ; Distance Metric ; Geometrical Embedding.

Routing algorithms play a crucial role in modern transportation systems, but their effectiveness is constrained by the availability and quality of Road distance data. This thesis aims to address this issue by developing a methodical approach for building Road distance based data sets using existing Euclidean distance-based data sets. Transportation providers and end-users will benefit by improving the accuracy and efficiency of routing algorithms. Moreover, this paper proposes statistical metrics to investigate the topological characteristics and underlying network's closeness to existing standard distance metrics of road networks from selected cities in India. The four primary objectives of this project are to develop a systematic approach for converting Euclidean distance-based coordinates to Road distance based coordinates for a specified urban area, conduct a comparative analysis of the initial and transformed embedding, evaluate the routing algorithm's efficacy on both types of embedding, and suggest a simple criterion or criteria to assess the connectivity and efficiency of road networks in Indian urban areas. The study's findings will offer valuable insights into the potential benefits of using Road distance based data sets and statistical metrics in urban planning and transportation management in India.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

VRP	Vehcile Routing Problem
CVRP	Capacitated Vehicle Routing Problem
Lat	Latitude
Lon	Longitude

NOTATION

r	Radius
$d(x, y)$	Distance between point x, point y
R	Radius of Earth, 6378 km
λ	Longitude
ϕ	Latitude

CHAPTER 1

INTRODUCTION

1.1 Motivation

The use of Vehicle Routing algorithms has become integral in transportation systems ranging from logistics to ride-sharing services. These algorithms rely on distance data to determine the optimal route between two points. However, the precision of routing algorithms is restricted by the quality and availability of training data sets that use Road distance data. Given the importance of routing algorithms in modern transportation systems, it is critical to address this **lack of real-world distance data**.

Through this project, I aim to develop a systematic method for building Road distance based data sets using the existing Euclidean distance based data sets. I believe this will contribute to the development of more accurate and efficient routing algorithms, benefiting both the transportation providers and end-users.

Urban road networks are vital in underpinning our society and economy. Particularly, nowadays with the growing popularity of technologies related to smart cities, physical roads and networks are receiving increasing attention. This paper proposes a few statistical metrics that investigates topological characteristics and underlying network's closeness to existing standard distance metrics of road networks from selected cities in India.

1.2 Objectives

The objectives of this project are as follows:

1. To devise a methodical approach for converting Euclidean distance based coordinates to Road distance based coordinates to fit inside a specified urban area.
2. To conduct a comparative analysis of the initial and transformed embedding.
3. To evaluate the routing algorithm's efficacy on both types of embedding.

4. To suggest a simple criterion (or criteria) to assess the connectivity and efficiency of road networks in Indian urban areas.

1.3 Research Methodology

This research project has employed two major types of research methodologies, **quantitative and theoretical** methods, to comprehensively investigate various aspects of road networks.

A key area of interest is the comparison of different metrics used to analyze road networks, such as Road distance and Haversine distance. To this end, we utilized quantitative methods to conduct a comparative analysis of these metrics using an available data set.

In evaluating the routing algorithm's performance using total cost of the optimal route, distance matrix etc. numerical analysis was used. We also generated several graphs to gain valuable insights into the performance of these metrics and their impact on road network efficiencies.

Another area of our research project involved the formalization of Haversine and Road distance as a metric, which was approached qualitatively. We applied appropriate metric embedding formulas to predict the bounds of these metrics and better understand their theoretical foundations.

Our mixed-methods research provides a comprehensive understanding of road networks' structure, performance, and metrics. The implications of our findings include optimizing road network design and development of accurate routing algorithms. Additionally, our theoretical analysis of metrics such as Haversine and Road distance contributes to understanding their potential applications in different fields.

1.4 Dataset

1.4.1 Input Data

The *CVRP* can be defined as follows. Input to the *CVRP* consists of $\boldsymbol{n}$ locations (a depot and a set of ***n-1*** customers), an $\boldsymbol{n} \times \boldsymbol{n}$ symmetric matrix $\boldsymbol{D}$ specifying the distance to travel between each pair of locations, a quantity $\boldsymbol{q_i}$ that specifies the demand for some resource by each customer $\boldsymbol{i}$, and the maximum quantity, $\boldsymbol{Q}$, of the resource that a vehicle can carry.

A feasible solution to the *CVRP* consists of :

- A set of routes that begin and end at the depot
- Each customer is visited on exactly one route
- The total demand by the customers assigned to a route does not exceed the vehicle capacity Q

An optimal solution for *CVRP* is a feasible solution that minimizes the total combined distance of the routes.

Here is a sample instance file.

NAME : A-n6-k5
COMMENT : (Augerat et al, No of trucks: 5, Optimal value: /*cost*/)
TYPE : CVRP
DIMENSION : 6
EDGE_WEIGHT_TYPE : EUC_2D
CAPACITY : 50
NODE_COORD_SECTION
1 82 76
2 96 44
3 50 5
4 49 8
5 13 7
6 29 89
DEMAND_SECTION
1 0
2 19
3 21
4 6
5 19
6 7
DEPOT_SECTION
1
-1
EOF

Benchmark datasets such as Augerat (1995) and Uchoa (2017) were used from the CVRPLIB hosted by DIMACS, Rutgers University.

The number of nodes ***n*** range from 30 to 80, while the number of trucks ***k*** range from 5 to 10. The capacity ***Q*** of each truck is kept constant as 100. There are a total of 27 data sets used for the embedding, analysis of algorithmic performance and to study road networks in Indian cities.

Here is a sample input and output plot of the given Euclidean-distance based input data points on the x-y coordinate system obtained from CVRPLIB.

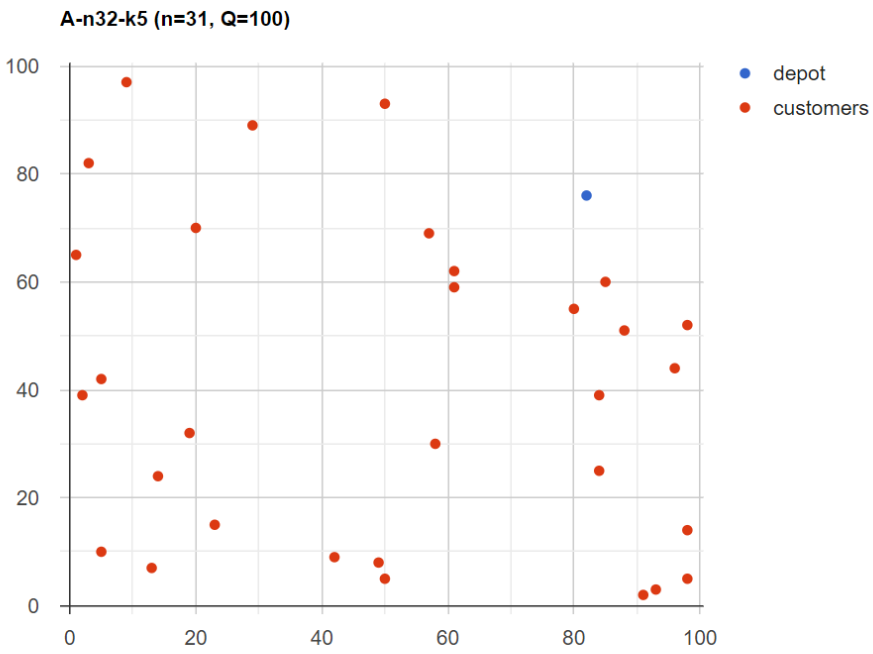

Figure 1.1: Sample input of 32 points
CVRPLIB (2023)

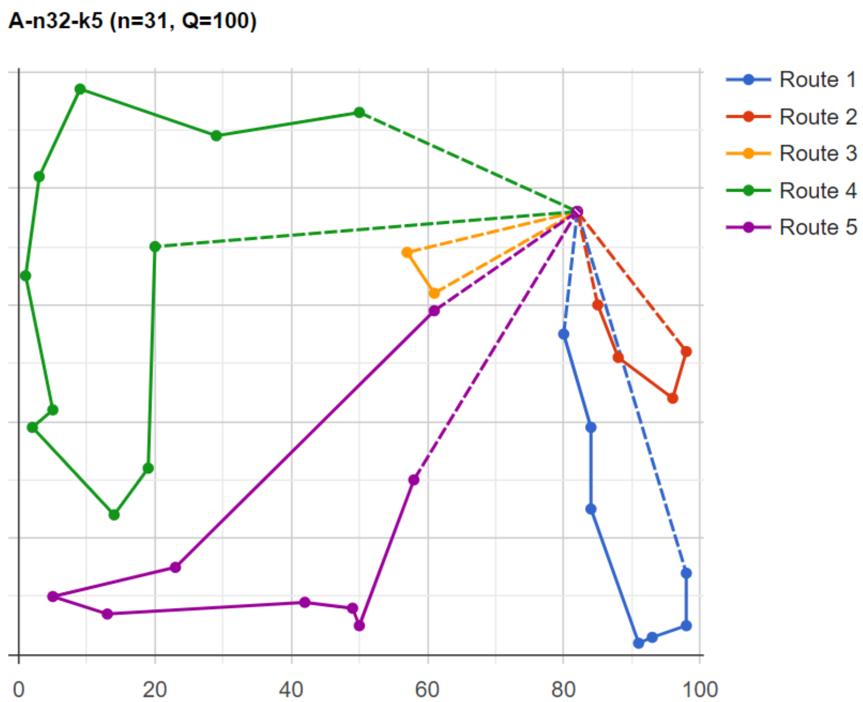

Figure 1.2: Sample output with routes for each truck
CVRPLIB (2023)

1.4.2 Classes and Storage

Table 1.1: Vehicle Class

Vehicle	
Member	Use
ID	Unique Identifier
Maximum Capacity	Physical property
Available Capacity	Configuration in some state
Consignment List	Assigned Consignments
Route	Assigned route with list of stops
Loc	Current Location
Serviceable area, Assigned depot	Constraints

Table 1.2: Consignment Class

Consignment	
Member	Use
ID	Unique Identifier
Weight	Physical property
Pickup Loc	Depot
Drop-off Loc	Drop Location

Table 1.3: Depot Class

Depot	
Member	Use
ID	Unique Identifier
Loc	Location
Vehicle List	Assigned vehicles

Table 1.4: Location Class

Location	
Member	Use
(Lat, Lon)	Geographical
(X, Y)	Euclidean

CHAPTER 2

LITERATURE REVIEW

Metric embedding has found many applications in computer science and related fields, including data compression and algorithm design. One of the most famous applications of metric embedding is in the design of approximation algorithms for NP-hard optimization problems, where the embedded space can be used to design a simpler algorithm that works with the lower-dimensional data while still producing good-quality solutions. Metric embedding is finding a mapping of the objects into another space such that the pairwise distances in the embedded space are as close as possible to the original distances.

2.1 Metric

2.1.1 Definition

The idea of metric needs to be formalized in order for us to proceed further.

A **metric space** is a pair (X, d_X), where X is a set and d_X : X × X $\rightarrow$ R is a metric satisfying the following axioms (x, y, z are arbitrary points of X):

$$d_X(x, y) \geq 0, \tag{2.1}$$

$$d_X(x, x) = 0, \tag{2.2}$$

$$d_X(x, y) > 0 \; \forall \; x \neq y, \tag{2.3}$$

$$d_X(y, x) = d_X(x, y), and \tag{2.4}$$

$$d_X(x, y) + d_X(y, z) \geq d_X(x, z). \tag{2.5}$$

Although, metrics like the Euclidean, Manhattan and Haversine might satisfy the above definition of a metric space. Our focus is on formalizing the road-distance as a metric. It is evident that the distance metric being directional and possibility of several distance measures between two points, need **not necessarily satisfy the triangle**

inequality in 2.5. Below is the definition of a ***graph metric*** obtained from Matousek (2013).

2.1.2 Graph Metric

Given a graph G (simple, un-directed) with vertex set V , the distance of two vertices u, v is defined as the length of a **shortest path** connecting u and v in G, where the length of a path is the number of its edges. (We need to assume G is connected).

More generally, we can consider a weighted graph G, where each edge e $\in$ E(G) is assigned a positive real number w(e), and the length of a path is measured as the **sum of the weights of its edges**.

The possibility of an urban road network being perceived as a **weighted graph** and hence Road distance as a graph metric is discussed further in this project.

2.2 Embedding

2.2.1 Isometric Embedding

A mapping f: (X, d_X) $\rightarrow$ (Y, d_Y) of one metric space into another is called an isometric embedding or isometry if

$$d_Y(f(x), f(y)) = d_X(x, y) \ \ \forall x, y \in X \tag{2.6}$$

This is an **exact representation** of one metric space in another.

To visualize a metric space, exact representation (isometric) of distances is not necessary, and often not even possible. An approximate embedding that allows for some margin of error would suffice, but it is important to **quantify and control the error**.

2.2.2 Approximate Embedding and Distortion

Let (X, d_X) and (Y, d_Y) be metric spaces. An injective mapping f: (X, d_X) $\rightarrow$ (Y, d_Y) is called a **D-embedding**, where D $\geq$ 1 is a real number, if there is a number r $>$ 0 such that $\forall$ x, y $\in$ X,

$$r \cdot d_X(x, y) \leq d_Y(f(x), f(y)) \leq D \cdot r \cdot d_X(x, y) \tag{2.7}$$

The **greatest lower bound** of the numbers D such that f is a D-embedding is called the **distortion** of f.

Quantitative analysis was done on limited available data sets to explore the potential for metric embedding between Euclidean and Haversine distances, as well as between Euclidean and Road distances.

2.3 The Routing Algorithm

The VRP is a well-known **NP-hard** combinatorial optimization problem that has been studied extensively in the field of operations research and computer science. Despite its practical relevance, VRP is considered a difficult problem to solve due to its combinatorial nature and the large number of possible solutions. Therefore, researchers have developed a variety of algorithms and techniques to solve different variants of the VRP, one of which is the CVRP. The **Savings Algorithm** is a popular **heuristic** approach for solving the CVRP.

2.3.1 The Savings Heuristic

The Savings algorithm was first introduced by G.Clarke (1964) and has been practiced widely due to its simplicity and effectiveness. The Savings algorithm works by first calculating the savings achieved by combining two customers into a single route, rather than serving them separately. **The savings are defined as the reduction in distance between the two customers that results from the combination of their routes**. The savings can be calculated as follows:

$$s[i, j] = d[dp, i] + d[i, j] - d[dp, j] \quad \forall i, j \in N, i < j \tag{2.8}$$

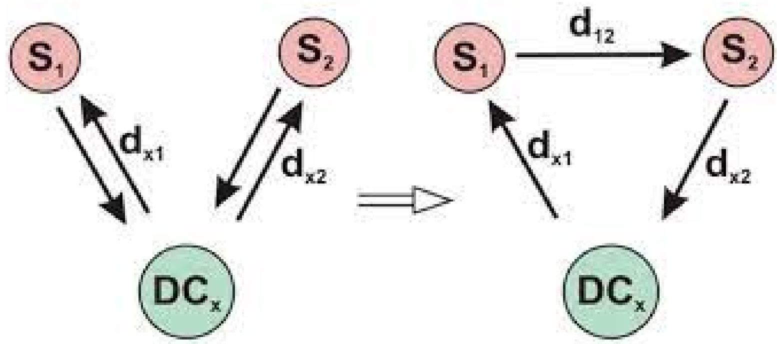

Figure 2.1: Demonstration of savings for any three nodes
Martin *et al.* (2015)

where d[i,j] is the distance between nodes i and j, dp is the depot node and N is the set of nodes including the depot. This is depicted in Fig 3.1.

The savings values are then sorted in decreasing order and the algorithm proceeds to construct a set of routes that satisfy the capacity constraints of the vehicles. Starting with the largest saving value, the algorithm attempts to merge the two customers into a single route, subject to the capacity constraints of the vehicles. If the merge is feasible, the two routes are combined and the corresponding savings value is added to the total savings. This process is repeated until all customers have been assigned to a route.

Sorting the savings (of order n^2) takes $\mathbf{O(n^2 \cdot log(n))}$ time where n is the number of customers.

2.3.2 Implementation

The python code and implementation of the savings algorithm discussed earlier were obtained from the work of my colleague, Madhav Mittal, in the logistics lab during his UGRC. The specifics of this work are documented in Mittal (2022)

CHAPTER 3

THE TRANSFORMATION ALGORITHM

Our objective is to map a set of Euclidean coordinates onto a desired land mass on Earth by implementing a systematic procedure. The input data comprises x and y coordinates on a 2D plane, whereas the output consists of latitude and longitude values that fall within the given boundary on Earth (3D). Our aim is to minimize deviations from linearity, ensuring that corresponding distances in the distance matrix remain proportional.

3.1 The Process

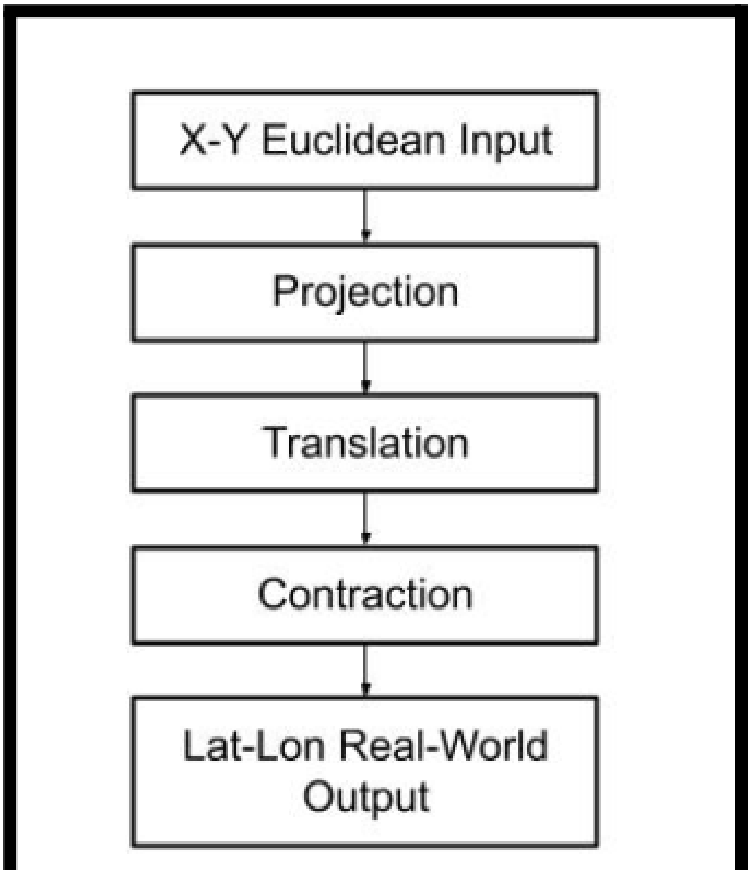

Figure 3.1: Step-by-step transformation of 2D inputs to 3D coordinates

3.1.1 Projection

Projection (here) refers to the **Inverse Map Projection**. It is a process used to convert points on a 2D map to their corresponding points on the Earth's surface. Some of the properties that can be measured on the Earth's surface independent of its geography are:

1. Area
2. Shape
3. Direction
4. Distance

A 2D plane is **not isometric** to the earth's surface. So we preserve some of these properties at the expense of others.

It is helpful to think about map projections and back track our process from there. Most projections transform part of the globe to one of three "developable surfaces" (flat or can be made flat) :

- Cylinder
- Cone
- Plane

It can be visualised as shown in Fig 4.2.

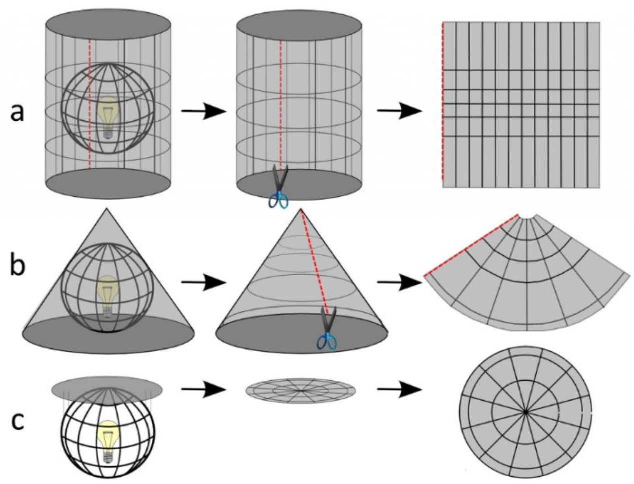

Figure 3.2: Developable Surfaces in Map Projection
University of Minnesota Libraries

The tangent point or line is where the developable surface touches the globe. Maps are most accurate at these tangent points, with distortion increasing farther away due to shearing and compression. **Hence, cylinders are used for equatorial regions, cones for mid-latitudes, and planes for polar regions for this reason.**

Since India is located in between the equator and the mid-latitudes (as shown in 3.3), cylindrical and conical projection methods are appropriate choices for creating a map that almost accurately represents its features such as distance and angles. It results in less distortion in this region compared to other map projections.

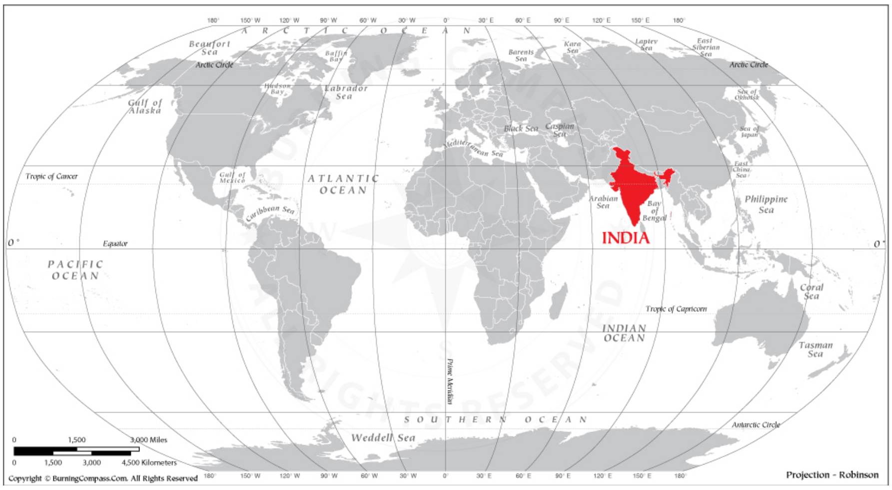

Figure 3.3: Location of India on the globe between equator and mid-latitudes Burning Compass (2023)

Various map projections were taken into consideration accordingly from USGS (2000). Upon referring to the National_Grid_System (2023), India uses:

1. Lambert Conformal Conic Projection
2. Mercator Projection - Cylindrical

Lambert Conformal Conic Projection

The Lambert conformal conic projection is a map projection that preserves the shape and size of small areas, making it suitable for mapping mid-latitude regions.

Definition of Conformal : *In cartography, a conformal map projection is one in which every angle between two curves that cross each other on Earth (a sphere) is preserved in the image of the projection, i.e. the projection is a conformal map in the mathematical sense. For example, if two roads cross each other at a 27° angle, then their images on a map with a conformal projection cross at a 27° angle.*

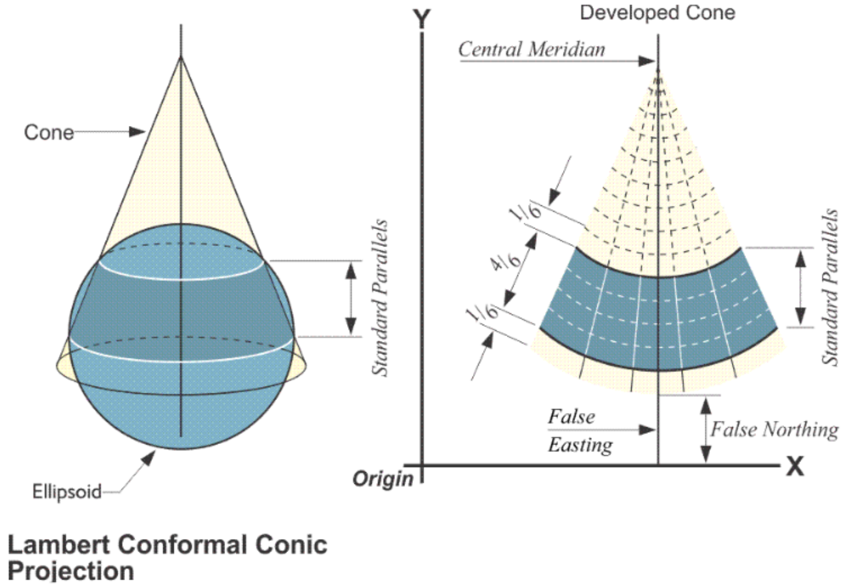

Figure 3.4: Lambert conical projection explained
Jan Van Sickle

The transformation of planar coordinates to spherical coordinates via the Lambert conformal conic projection is given as follows.

Here are the definitions of the symbols used,

- λ - Longitude
- λ_0 - Reference Longitude
- ϕ - Latitude
- ϕ_0 - Reference Latitude
- ϕ_1, ϕ_2 - Standard parallels

We consider the standard values for India (in deg) as (from Daumiller (2018)),

- λ_0 - 82.5 longitude (central meridian)
- ϕ_0 - 8 (latitude of origin)
- ϕ_1 - 17.7 latitude
- ϕ_2 - 27.3 latitude

The projection formula is given by,

$$\phi = 2 \cdot \arctan((\frac{F}{\rho})^{\frac{1}{n}}) - \frac{\pi}{2} \tag{3.1}$$

$$\lambda = \lambda_0 + \frac{\theta}{n} \tag{3.2}$$

where,

$$\rho = sgn(n) \cdot (x^2 + (\rho_0 - y)^2)^{\frac{1}{2}} \tag{3.3}$$

$$\theta = \arctan{(\frac{x}{\rho_0 - y})} \tag{3.4}$$

with,

$$F = \frac{\cos{(\phi_1)} \cdot \tan^n(\frac{\pi}{4} + \frac{\phi_1}{2})}{n} \tag{3.5}$$

$$n = \frac{\ln{(\cos{(\phi_1)} \cdot \sec{(\phi_2)})}}{\ln{(\tan{(\frac{\pi}{4} + \frac{\phi_2}{2})} \cdot \cot{(\frac{\pi}{4} + \frac{\phi_1}{2})})}} \tag{3.6}$$

$$\rho_0 = F \cdot \cot^n{(\frac{\pi}{4} + \frac{\phi_0}{2})} \tag{3.7}$$

Mercator Projection (Cylindrical)

This is a cylindrical map projection method, which is the standard used in navigation. It preserves shapes, local direction implying that it is conformal, while we compensate for the size of areas away from the equator ashow in 3.5. The corresponding formula for inverse projection is :

$$\phi = 2 \cdot \arctan{(\exp{(\frac{y}{R})})} - \frac{\pi}{2} \tag{3.8}$$

$$\lambda = \lambda_0 + \frac{x}{R} \tag{3.9}$$

The value λ_0 is the longitude of an arbitrary central meridian that is usually Greenwich (i.e., zero). R stands for the radius of Earth taken as 6378 km (it's equatorial radius).

Figure 3.5: Areas away from the equator get disproportionately enlarged. Wikipedia (2023)

3.1.2 Translation

After projecting our 2D coordinates onto the surface of the Earth as latitudes and longitudes, it is necessary to shift all these points by a constant to center them around the desired city. Each input coordinate is translated using the following formula:

$$lat_{new} = lat_{old} + (city\_center\_lat - old\_center\_lat) \tag{3.10}$$

$$lon_{new} = lon_{old} + (city\_center\_lon - old\_center\_lon) \tag{3.11}$$

where the ***shifting distance*** is calculated as the difference between the desired city center and the previous center's corresponding latitude and longitude.

The old center is determined by computing the average of the minimum and maximum values of x and y values respectively.

The *geolocator* function from *Nominatim* API is used to obtain the *city_center* value by inputting the city's name, for instance, "Chennai".

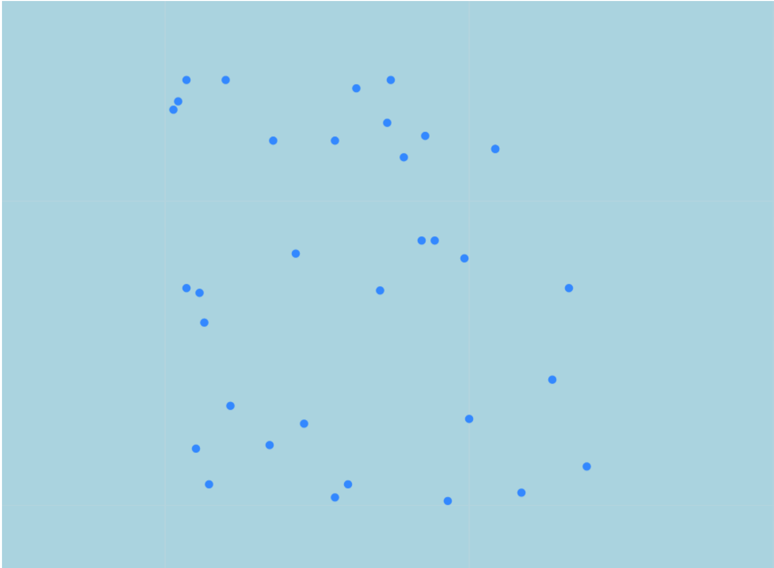

Figure 3.6: Map obtained after projection using Mercator

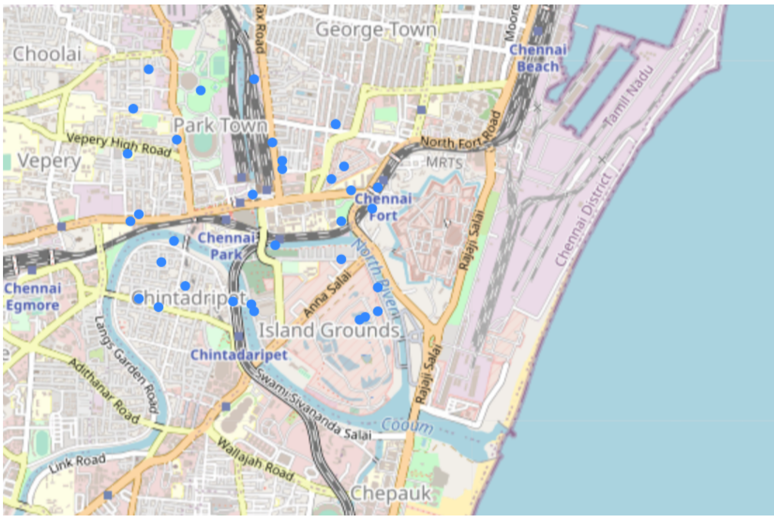

Figure 3.7: Map obtained after translation and fit into Chennai city

3.1.3 Contraction

To project the points within a desired city, it is necessary to compress them until they fit within the city's ***bounding polygon***.

This involves ***radial compression*** of the points relative to the city's center coordinate.

This process is ***iterative***, with each new point calculated midway (*step* = 0.5) between the old point and the city's center.

The city's bounding polygon can be obtained from the *geocode_to_gdf* function in the *OSMnx* library by providing the city's location, such as "Chennai, Tamil Nadu, India".

Here (3.8) is the bounding polygon figure and data points of Chennai city.

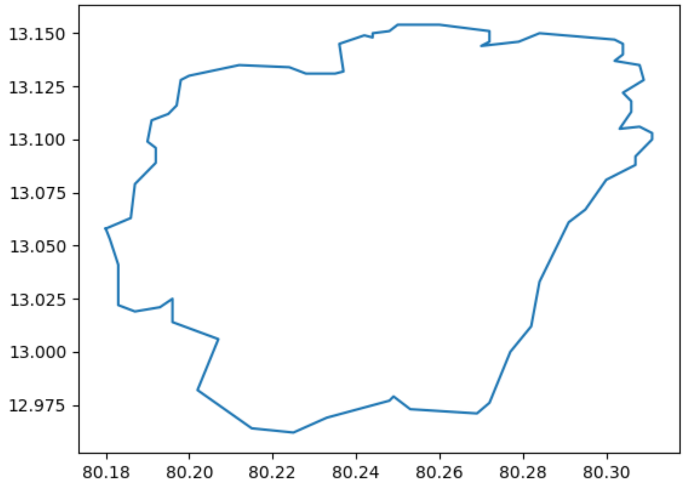

Figure 3.8: Chennai City's bounding polygon

The polygon data points can be obtained and visualized by inputting the desired location's OSRM ID ("1766358" - Chennai city corporation) on Polygon creation. It also provides features to increase or decrease the number of vertices in the polygon, thereby allowing us to obtain a simpler or a more complex boundary.
To verify whether each coordinate lies within the bounding polygon, the *contains* function from the *Shapely* library's *polygon* module is used, which returns a boolean value. One can store the final compression metric as

$$contraction\_factor = (number\_of\_iterations \cdot step) \tag{3.12}$$

The compression method can be modified by altering the **step** factor. Additionally, optimization of the process can be achieved by identifying the largest polygon contained within the city and compressing the data points to fit inside that polygon.

The converted latitudes and longitudes on the Earth's spherical surface for each location in the 2D input data are now available in the final list.

3.1.4 Analysis

The main objective of the transformation algorithm is to retain the values in distance matrix proportionately. The effectiveness of projection was analysed using metrics such as :

1. **RMSE : Root Mean Squared Error**

$$rmse = (\frac{\sum_{i=1}^{N}(Trasnformed_i - Euclidean_i)}{N})^{\frac{1}{2}} \quad (3.13)$$

Here, transformed distance could be either Haversine or the Road distance obtained from OSRM. Based on the analysis of the graphs presented below, which display the mean values of the data as dotted lines, we may conclude that :

(a) The Mercator projection method is suitable while considering Haversine distance (3.9)

(b) The Lambert conical conformal projection is a better fit when considering Road distance in the case of Chennai city (3.10)

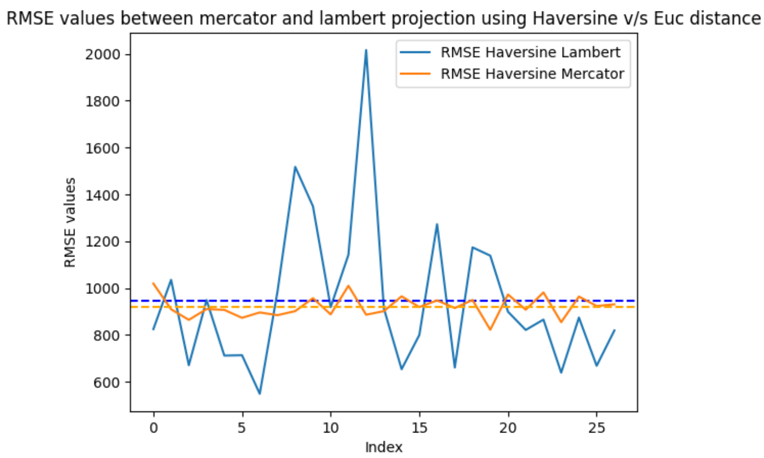

Figure 3.9: RMSE values on Haversine Distance

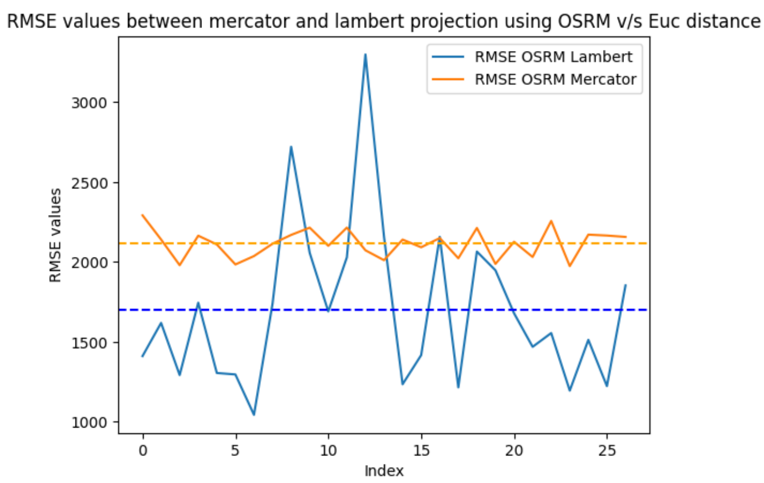

Figure 3.10: RMSE values on Road distance

2. **Correlation Coefficient**

The Correlation Coefficient is a measure of the degree of linear association between the transformed and original distances.
The results show that the Mercator projection exhibits a distinct linear relationship between the transformed and original distances for Road distance metrics as displayed in the below graphs (3.11).

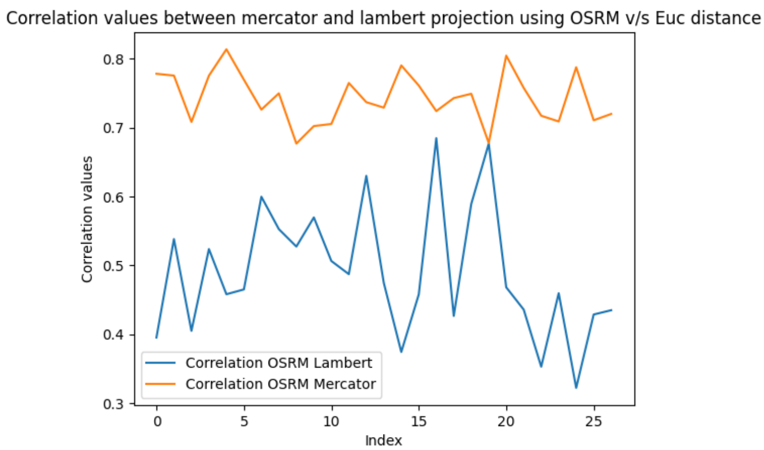

Figure 3.11: Corrrelation Coefficient

After conducting an analysis of the RMSE, Correlation values for both Mercator and Lambert projection methods in Chennai city, we have found that the Mercator projection method provides a higher RMSE but a higher Correlation value too compared to the Lambert projection method. We shall further investigate the Distortion values of these two embedding methods in 4.1.4.

CHAPTER 4

EMBEDDING AND PERFORMANCE

4.1 Geometric Embedding

4.1.1 Haversine Distance as a Metric

The Haversine distance is a formula used to calculate the shortest distance between two points on the surface of a sphere, such as the Earth. It takes into account the radius of the sphere and the latitude and longitude of the two points in question. The formula is based on the law of haversines, which relates the sides and angles of a spherical triangle.

Definition

Mathematically, the Haversine distance d and the Haversine function are given by:

$$d = R \cdot \theta \tag{4.1}$$

$$hav(\theta) = \sin^2(\frac{\theta}{2}) \tag{4.2}$$

where,

θ - central angle between two points on the sphere

d - distance between two points along a great circle of the sphere

R - Radius of Earth

The *Haversine formula* that computes Haversine of θ in terms of the latitude of longitude of the two points is given by:

$$hav(\theta) = hav(\phi_2 - \phi_1) + cos(\phi_1) \cdot cos(\phi_2) \cdot hav(\lambda_2 - \lambda_1) \tag{4.3}$$

where,

λ_1, λ_2 - longitudes of two points

ϕ_1, ϕ_2 - latitudes of two points

On solving for d [A.1], we get

$$d = 2 \cdot r \cdot \arcsin(\sin^2(\frac{\phi_2 - \phi_1}{2}) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2(\frac{\lambda_2 - \lambda_1}{2}))^{1/2} \tag{4.4}$$

Metric Properties

As we can see, the Haversine distance follows properties (2.2) to (**??**). The triangle inequality (2.5) is verified below.

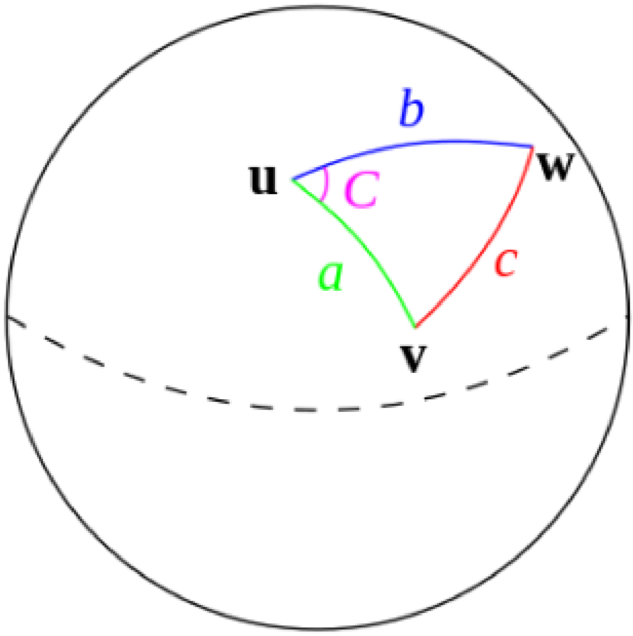

Figure 4.1: Spherical triangle

$$a = d(u, v), b = d(v, w), c = d(w, u) \tag{4.5}$$

Using the spherical law of cosines, the cosine addition theorem and using Haversine formula, gives the equation, where C is the angle of the spherical triangle at u,

$$hav(a + b) = hav(c) + sin(a) \cdot sin(b) \cdot cos(C/2)^2 \tag{4.6}$$

Fro detailed proof refer to [A.2].

Hence, we conclude that Haversine distance can be formulated as a metric space.

4.1.2 Road distance as a metric?

Road distance can be formalized as a metric space by defining a distance function that satisfies the properties of a metric. In this context, distance function would measure the distance between any two points on a road network, taking into account the actual road layout and relevant factors such as one-way streets. As we can have several routes between the two given points, implying the possibility of different distances, we need to fix the distance function to proceed further.

Graph Metric

Given a graph ***G,*** with vertex set ***V***, the distance of two vertices ***u,*** v is defined as the **length of the shortest path** connecting u and v in G, where the length of a path is the ***sum of the weights of its edges***. Each edge can be assigned a weight that represents the length of the road segment. Once we have the representation, we can compute the shortest path between two points using standard graph algorithms such as Dijkstra's.

The resulting funtion would satisy the properties of a metric namely:

1. Non-negativity : Distance between any two points on the road network is non-negative.
2. Identity : Distance between a point and itself is zero.
3. Symmetry : The distance between two points is the same regardless of the order in which they are considered (undirected, two-way roads).
4. Triangle inequality : Because the shortest path is always the path that minimizes the total distance between two vertices, it is impossible for the distance between two vertices to be greater than the sum of the distances between those vertices and any intermediate vertices. In other words, the triangle inequality holds for graph metrics by construction.

4.1.3 Embedding Road distance with Euclidean distance

As we already saw, an isometric embedding of Road distance with euclidean in (5.6), seems practically impossible.

$$d_{road}\left[f(x), f(y)\right] = d_{euc}\left[x, y\right] \tag{4.7}$$

We incline towards investigating **approximate embedding**. Road distance d_Y (f(x), f(y)) can be modeled as a D-embedding to Euclidean distance d_X (x, y) with a distortion ***D***, where D $\geq$ 1 is a real number, if there is a number ***r*** $>$ 0 such that for all x, y $\in$ X

$$r \cdot d_X(x, y) \leq d_Y(f(x), f(y)) \leq D \cdot r \cdot d_X(x, y) \tag{4.8}$$

The value of distortion in our case can depend on the type of inverse projection adopted, contraction factor required etc.

4.1.4 Analysis

In this study, the effectiveness of approximate embedding from Euclidean distance to Road distance was evaluated for each dataset by computing the values of r and D. The r value, also known as the greatest lower bound, was determined using the Shepard-Kruskal algorithm, which utilized the minimize_scalar module of the SciPy library. On the other hand, the D value, also known as the distortion, was calculated by taking the sum of the mean squared error and dividing it by the sum of the Euclidean distance squares.

After obtaining the values of r and D, a comparison was made between the Lambert and Mercator projections. A lower value of D indicated a better embedding, and thus, the Lambert projection was found to be a superior embedding method for the Road distance metric in Chennai city as shown in 4.2.

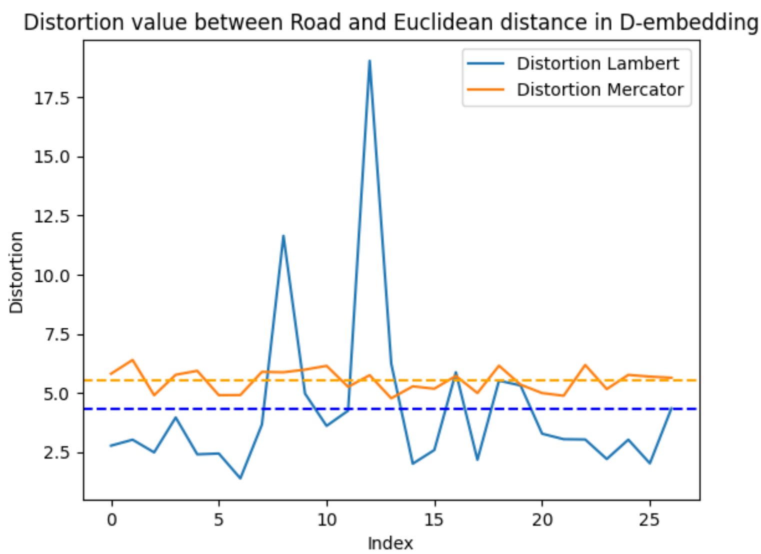

Figure 4.2: Approximate Embedding Analysis

4.2 Routing Algorithm Performace

4.2.1 Implementation

Although the code works based on savings's algorithm, the difference comes into picture while calculating the distance matrix.

Euclidean

Algorithm 1 Euclidean Distance matrix Calculator

```
function EUCLIDEAN(a, b)
    eucDist ← (a.x − b.x)^2 + (a.y − b.y)^2
    return √eucDist
end function
```

$eucDist \leftarrow (a.x - b.x)^2 + (a.y - b.y)^2$; **return** $\sqrt{eucDist}$

Haversine

Algorithm 2 Haversine Distance matrix Calculator

function HAVERSINE(a, b)

$R \leftarrow 6372.8$ ▷ Radius of Earth in Kilometres

$dLat \leftarrow radians(b.lat - a.lat)$

$dLon \leftarrow radians(b.lon - a.lon)$

$lat1 \leftarrow radians(a.lat)$

$lat2 \leftarrow radians(b.lat)$

$a \leftarrow \sin^2\left(\frac{dLat}{2}\right) + \cos\left(lat1\right) \cdot \cos\left(lat2\right) \cdot \sin^2\left(\frac{dLon}{2}\right)$

$c \leftarrow 2 \cdot \arcsin\left(\sqrt{a}\right)$

return $R \cdot c$

end function

Road distance

This is obtained with the help of *OSRM*'s *distances* module, which when given a list of latitudes and longitudes, returns their corresponding distance matrix with Road distance between each point. The python code snippet used to obtain the distance matrix is given below.

```python
import requests

def osrm_distance_matrix(coords):
    # calculates distance matrix for the given array of coords
    url = 'http://router.project-osrm.org/table/v1/driving/'
    for c in range(len(coords)):
        lng = str(coords[c].lon)
        lat = str(coords[c].lat)
        if c == 0:
            url = url+lng+','+lat
        else:
            url = url + ';'+lng + ',' + lat

    url = url+'?annotations=distance'

    dis_mat = requests.get(url).json()['distances']
    return dis_mat
```

4.2.2 Analysis

The purpose of this study is to use the D-embedding approach to predict the output cost of Road distance by utilizing the cost associated with euclidean distance. In this study, we have plotted a graph showing the predicted bounds for the output cost of Road distance based on the D-embedding of euclidean distance. The predicted output values fall within the upper and lower bounds, indicating that our model has effectively captured the relationship between Road distance and euclidean distance.

However, it is important to note that this result may be specific to this particular dataset and may not be generalizable to all datasets. To address this, we will further refine our mathematical model to better predict the greatest lower bound (r) and distortion (D) in the approximate embedding.

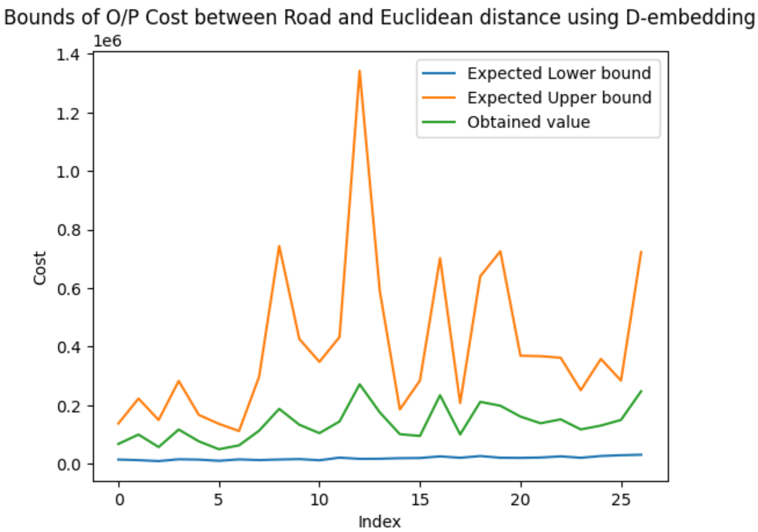

Figure 4.3: Approximate Embedding Bounds Analysis

CHAPTER 5

ANALYSIS AND CONCLUSION

5.1 City Network Analysis

5.1.1 Underlying Network as a standard metric

The objective of this study is to analyze the road network of a city by mapping it to the closest standard metric such as Euclidean, Manhattan, and Haversine distances. This analysis aims to provide insights into the road structure, travel distances etc. of the city's roads. The findings of this research could prove valuable for optimizing logistics operations in different areas based on their connectivity and structural characteristics.

Analysis

To analyze the road network of these cities mathematically, we can use graph theory. In graph theory, a network is represented as a graph, which consists of nodes (also called vertices) and edges. In the case of a road network, the nodes represent the intersections or junctions, and the edges represent the roads that connect them.

To measure the efficiency of a road network, we can use various metrics, such as the average shortest path length, the diameter (the longest shortest path between any two nodes), the clustering coefficient (a measure of how well-connected the nodes are), and the degree distribution (a measure of how many edges are connected to each node).

F-statistic

The code applies the **k-means clustering algorithm** to the distance matrix with a specified number of clusters $\boldsymbol{k}$. The labels of each node are then assigned based on the cluster they belong to.

Next, the code calculates the centroids of each cluster and uses them to calculate the within-cluster sum of squares (**WSS**) and between-cluster sum of squares (**BSS**). The WSS is the sum of the squared distances between each node and its assigned centroid within each cluster. The BSS is the sum of the squared distances between each centroid and the mean of all nodes within each cluster.

Finally, the code calculates the F-statistic using the WSS and BSS values, and prints out the result. A high F-statistic indicates that the clusters are significantly different from each other, which means that the distance metric used is ineffective in capturing the underlying structure of the road network. For example Haryana city's F-statistic value is shown below in 5.1. The city's F-statistic value is least for Manhattan distance among Manhattan, Haversine and Euclidean distance. Further details on this is discussed in the next section 5.1.1.

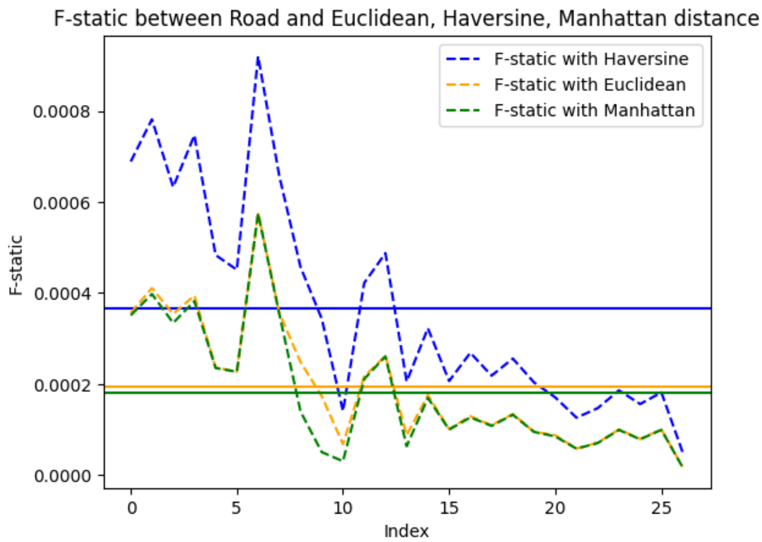

Figure 5.1: Haryana city's F-statistic value comparison

Simple Efficiency Metric

The efficiency of a road network calculated as the average of reciprocals of each value in the distance matrix is known as the **harmonic mean efficiency**.

This metric considers all the distances between pairs of nodes in the network and provides a comprehensive evaluation of the network's efficiency.

One of the advantages of using the harmonic mean efficiency is that it gives more weight to smaller distances in the matrix, which may be more relevant for local travel and transportation planning. However, this metric may not be suitable for larger networks or long-distance travel, as it does not consider the effects of traffic congestion or other factors that can impact travel time.

Overall, the harmonic mean efficiency provides a useful tool for evaluating the efficiency of a road network based on the distance matrix, and can be used to compare different networks or assess the impact of changes in the network structure.

For example Chennai city's efficiency is shown below in 5.2. The city's road network efficiency is close to Haversine distance, details of which will be discussed in the next section 5.1.1. Additionally as expected Eucldiean distance metric has the maximum efficiency, considering that it evaluates the shortest distacne between two points.

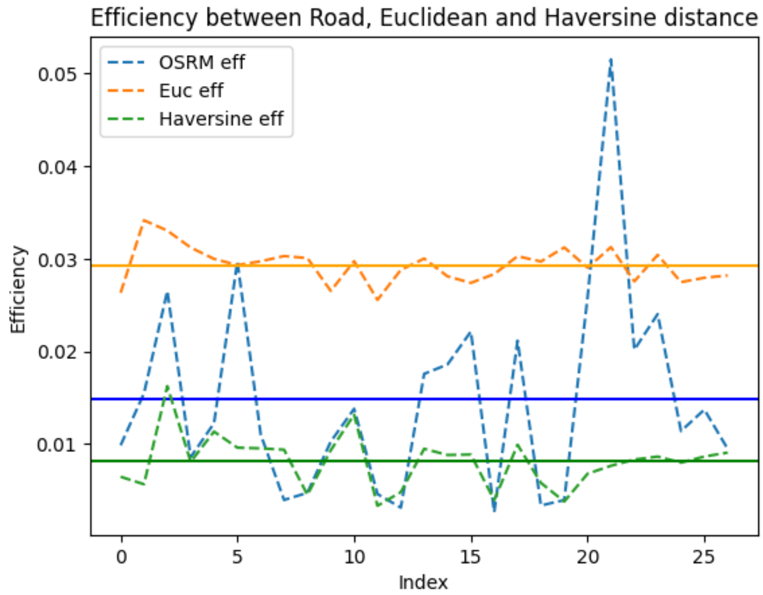

Figure 5.2: Chennai city's efficiency comparison

Discussion

Analyzing the road network of a city or a state, in terms of a distance metric, requires considering various factors such as the geography of the region, the quality of the roads, the population density, the presence of traffic congestion, and the types of transportation modes available.

Chennai

In the case of Chennai's road network, there are several reasons why the **Haversine** distance could be a reasonable choice for comparing it. First, Chennai is located on the southeastern coast of India, and the city's road network is primarily concentrated in and around the coastal region. The Haversine distance takes into account the curvature of the Earth, which makes it an appropriate metric for measuring distances on a spherical surface.

Second, Chennai is one of the most populous cities in India, with a population density of approximately 26,000 people per square kilometer. This high population density can lead to traffic congestion, which in turn can make it more difficult to travel directly between two points. The Haversine distance is useful in such cases because it considers the shortest path between two points on the surface of the Earth, rather than the straight-line distance between two points.

Finally, Chennai has a well-developed transportation network, including several major highways, rail networks, and a port. The Haversine distance can be used to measure the distances between various transportation hubs in the city, which can help in the planning and optimization of transportation routes.

In conclusion, while there may be other distance metrics that could be used to analyze Chennai's road network, the Haversine distance appears to be a reasonable choice due to its ability to take into account the spherical nature of the Earth's surface, its suitability for measuring distances in high population density regions with traffic congestion, and its applicability to the transportation network of the city.

Haryana

Firstly, much of Haryana's road network follows a grid-like pattern, especially in urban areas such as Gurugram and Faridabad. This means that the Manhattan distance can accurately reflect the Road distances between two points on the grid-like road network.

Second, Haryana is one of the most densely populated states in India, with a population density of approximately 600 people per square kilometer. The Manhattan distance is useful in such cases because it takes into account the length of the road segments that a vehicle must travel, rather than the straight-line distance between two points.

Finally, Haryana is a relatively small state, with a total area of approximately 44,000 square kilometers. As a result, distances between any two points on the road network are unlikely to be very large. The Manhattan distance is particularly well-suited for measuring distances in small regions, as it is computationally efficient and relatively easy to calculate.

In conclusion the Manhattan distance appears to be a reasonable choice due to its grid-like pattern, its ability to take into account traffic congestion, and its suitability for measuring distances in small regions, to analyze Haryana's road network.

Applications

1. **City planning**: The information obtained from the road network analysis can be used by city planners to make informed decisions about where to build new roads, how to optimize existing road networks, and how to allocate resources for public transportation.

2. **Logistics and delivery**: Companies that rely on transportation to move goods and services can use road network analysis to optimize their delivery routes, reducing costs and improving efficiency.

3. **Environmental impact**: Analyzing the road network can help identify areas where traffic congestion and idling vehicles contribute to air pollution. This information can be used to implement policies and initiatives aimed at reducing the environmental impact of transportation.

5.2 Results

1. **Inverse Map Projection**
   One of the major findings of this thesis is the identification of appropriate map projection methods and the development of a structured process for the inverse projection of 2D coordinates onto a closed urban area of interest. This process can be useful in various applications that require accurate spatial data analysis within the specified urban area.

2. **City's Road Network**
   Another result of this study is the identification of a few techniques to determine the most suitable standard distance metric for a city's road network. This finding can be applied in logistics and delivery planning to optimize transportation routes and reduce costs.

3. **Geometric Embedding**
   Overall this study demonstrates the effectiveness of the D-embedding approach in predicting the output cost of Road distance and provides a framework for optimizing the model to achieve better predictions. The results of this study will contribute to the field of transportation planning and provide valuable insights for decision-makers in the transportation industry.

5.3 Limitations and Future Work

1. **Determining a *centre***
   One variation to this study is that in the determination of the old_center (3.1.2) by computing the average of the minimum and maximum x and y values is a commonly used method, other methods such as using the median or the centroid of the smallest polygon encompassing all these points may yield different results. Further research can be done to investigate the impact of using different methods to determine the old_center on the overall results of the study.

2. **Compression to fit**
   A possible direction for future work is to explore changes to the compression method used in this study. For example, modifying the step factor or optimizing the process by finding the largest polygon that lies within the city and compressing the data points into that polygon may lead to different and potentially better results.

3. **Statistics on road network**
   Future work could also include exploring an in-depth analysis of various road network statistics such as connectivity, accessibility, centrality, and density in Indian cities. This can provide a more comprehensive understanding of the road network in these cities and their implications on various applications such as urban planning and transportation management.

4. **Exhaustive Testing**
   One of the limitations of this study is the relatively small number of datasets (27) used. Future work could involve expanding the dataset to a more exhaustive set to provide greater confidence in the obtained results. For example, extending the study to larger datasets could involve predicting bounds of the output cost in a Road distance based dataset using the Euclidean dataset, using the distortion values of the embedding. This can help to validate and reinforce the findings of the current study while providing insights into the scalability and reliability of the proposed methods.

APPENDIX A

A SAMPLE APPENDIX

A.1 Haversine Distance

$$\begin{aligned}
\text{hav}(\theta) &= \text{hav}(\varphi_2 - \varphi_1) + cos(\varphi_1) \cdot cos(\varphi_2) \cdot \text{hav}(\lambda_2 - \lambda_1) \\
\text{hav}(\theta) &= \sin^2(\tfrac{\theta}{2}) \\
\text{hav}(\theta) &= \tfrac{1-\cos(\theta)}{2}
\end{aligned}$$

The haversine distance **d** is given by:

$$\begin{aligned}
\mathbf{d} &= r \cdot \text{archav}(h) \\
&= 2r \cdot \arcsin(\sqrt{h}) \\
&= 2r \cdot \arcsin(\sqrt{\text{hav}(\varphi_2 - \varphi_1) + (1 - \text{hav}(\varphi_1 - \varphi_2) - \text{hav}(\varphi_1 + \varphi_2) \cdot \text{hav}(\lambda_2 - \lambda_1)}) \\
\implies \mathbf{d} &= 2r \cdot \arcsin(\sqrt{\sin^2(\tfrac{\varphi_2-\varphi_1}{2}) + (1 - \sin^2(\tfrac{\varphi_1-\varphi_2}{2}) - \sin^2(\tfrac{\varphi_1+\varphi_2}{2}) \cdot \sin^2(\tfrac{\lambda_2-\lambda_1}{2})})
\end{aligned}$$

A.2 Haversine Proof and Derivation

To prove that the Haversine distance follows the triangle inequality, we need to show that for any three points A, B, and C on the surface of a sphere, the distance from A to C is less than or equal to the sum of the distances from A to B and from B to C.

Let $\mathbf{d}(x, y)$ denote the Haversine distances between points x and y. We want to show that $\mathbf{d}(A, C) \leq \mathbf{d}(A, B) + \mathbf{d}(B, C)$

Using the formula for Haversine distance, we have:

$$\begin{aligned}
\mathbf{d}(A, B) &= 2r \cdot \arcsin(\sqrt{\sin^2(\tfrac{latB-latA}{2}) + \cos(latA)\cos(latB)\sin^2(\tfrac{lonB-lonA}{2})}) \\
\mathbf{d}(B, C) &= 2r \cdot \arcsin(\sqrt{\sin^2(\tfrac{latC-latB}{2}) + \cos(latB)\cos(latC)\sin^2(\tfrac{lonC-lonB}{2})}) \\
\mathbf{d}(A, C) &= 2r \cdot \arcsin(\sqrt{\sin^2(\tfrac{latC-latA}{2}) + \cos(latA)\cos(latC)\sin^2(\tfrac{lonC-lonA}{2})})
\end{aligned}$$

We can rewrite the formula for $\mathbf{d}(A,C)$ as:

$$
\begin{aligned}
\mathbf{d}(A,C) =& \mathbf{d}(A,B) + \mathbf{d}(B,C) \\
& - 2r \cdot \arcsin\left(\sqrt{\sin^2\left(\tfrac{latB - latA}{2}\right) + \cos(latA)\cos(latB)\sin^2\left(\tfrac{lonB - lonA}{2}\right)}\right) \\
& - 2r \cdot \arcsin\left(\sqrt{\sin^2\left(\tfrac{latC - latB}{2}\right) + \cos(latB)\cos(latC)\sin^2\left(\tfrac{lonC - lonB}{2}\right)}\right) \\
& + 2r \cdot \arcsin\left(\sqrt{\sin^2\left(\tfrac{latC - latA}{2}\right) + \cos(latA)\cos(latC)\sin^2\left(\tfrac{lonC - lonA}{2}\right)}\right)
\end{aligned}
$$

We can prove that $\mathbf{d}(A,C) \leq \mathbf{d}(A,B) + \mathbf{d}(B,C)$ by showing that the two terms on the right-hand side of the above equation are non-negative.

For the first term, we have:

$$
2r \cdot \arcsin\left(\sqrt{\sin^2\left(\tfrac{latB - latA}{2}\right) + \cos(latA)\cos(latB)\sin^2\left(\tfrac{lonB - lonA}{2}\right)}\right) \leq \mathbf{d}(A,B)
$$

This inequality follows from the fact that the Haversine distance is always the shortest distance between two points on a sphere, and therefore the distance between A and B along any other path must be greater than or equal to the Haversine distance.

Similarly, for the second term, we have:

$$
2r \cdot \arcsin\left(\sqrt{\sin^2\left(\tfrac{latC - latB}{2}\right) + \cos(latB)\cos(latC)\sin^2\left(\tfrac{lonC - lonB}{2}\right)}\right) \leq \mathbf{d}(B,C)
$$

Again, this inequality follows from the fact that the Haversine distance is always the shortest distance between two points on a sphere.

Therefore, we have:

$$
\begin{aligned}
\mathbf{d}(A,C) = & \mathbf{d}(A,B) + \mathbf{d}(B,C) \\
& - 2r \cdot \arcsin\left(\sqrt{\sin^2\left(\tfrac{latB - latA}{2}\right) + \cos(latA)\cos(latB)\sin^2\left(\tfrac{lonB - lonA}{2}\right)}\right) \\
& - 2r \cdot \arcsin\left(\sqrt{\sin^2\left(\tfrac{latC - latB}{2}\right) + \cos(latB)\cos(latC)\sin^2\left(\tfrac{lonC - lonB}{2}\right)}\right) \\
& + 2r \cdot \arcsin\left(\sqrt{\sin^2\left(\tfrac{latC - latA}{2}\right) + \cos(latA)\cos(latC)\sin^2\left(\tfrac{lonC - lonA}{2}\right)}\right) \\
& \leq \mathbf{d}(A,B) + \mathbf{d}(B,C) \\
\implies & \mathbf{d}(A,C) \leq \mathbf{d}(A,B) + \mathbf{d}(B,C) \qquad \text{(Hence Proved)}
\end{aligned}
$$

This completes the proof that the Haversine distance follows the triangle inequality.

REFERENCES

1. **Augerat, P.** (1995). Approche polyédrale du problème de tournées de véhicules (polyhedral approach of the vehicle routing problem). *Institut National Polytechnique de Grenoble - INPG*.

2. **Burning Compass** (2023). Where is india. URL https://www.burningcompass.com/on-world-map/india-on-world-map.html.

3. **CVRPLIB** (2023). Plotted instances and solution. URL http://vrp.galgos.inf.puc-rio.br/index.php/en/plotted-instances?data=A-n32-k5.

4. **Daumiller, G.** (2018). Map projection of indian subcontinent. URL https://www.quora.com/What-is-the-Map-Projection-of-Indian-Subcontinent.

5. **G.Clarke, J. W. W.** (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*.

6. **Jan Van Sickle** (). Lambert conformal conic projection. URL https://www.e-education.psu.edu/geog862/node/1812.

7. **Martin**, **R. Straka**, **P. Lenort**, and **Besta** (2015). Clarke and wright saving algorithm as a means of distribution streamlining in the environment of a concrete company. *Internation Scientific Journal about Logistics*. URL https://pdfs.semanticscholar.org/1de5/601827b32ac10e37ab274a501ec20a0fd077.pdf.

8. **Matousek, J.**, *Lecture Notes on Metric Embeddings*. Institute of Theoretical Computer Science, ETH Zurich, Switzerland, 2013.

9. **Mittal, M.** (2022). Vehicle routing problem - practical implementations using osrm.

10. **National_Grid_System** (2023). National grid - surveying. URL https://mineportal.in/blog/national-grid/1#:~:text=India%20used%20generally%20two%20types,Spheroid%20as%20a%20reference%20surface.

11. **Uchoa, A. M. T. A., Diego** (2017). New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research*.

12. **University of Minnesota Libraries** (). Scales and projections. URL https://open.lib.umn.edu/app/uploads/sites/178/2017/07/Image044.jpg.

13. **USGS** (2000). Map projections. URL https://www.icsm.gov.au/sites/default/files/projections.pdf.

14. **Wikipedia** (2023). Mercator map projection. URL https://en.wikipedia.org/wiki/Mercator_projection.