Report

Assignment 0: Review Unit
Chess Analyst
CSC 501

University
of Victoria

Submitted to:

Prof. Sean Chester

Submitted by:

Anushka Halder (V00961967)

Sanjana Arora (V00966221)

Tavanpreet S. Oberoi (V00963163)

1. Introduction

2. Data Modelling

2.1 Conceptual Data Model

It can be seen from the given dataset that it has both spatial and temporal attributes; hence it is required to model data that supports both spatial-temporal modelling.

The Spatial attributes include 'Latitude', 'Longitude', 'Altitude', 'Angle', 'Bearing', 'Heading' etc. while temporal attributes include 'Date' and 'Time'.

Apart from Spatial-temporal attributes (out.csv), the dataset also contains some weather and temperature related attributes (Dataset File 2), both are connected using 'Metar' attribute.

Problem Statement: It is a situation of finding the nearest neighbor. It is important to identify the spatial location of multiple aircrafts based on different weather conditions to minimize the collision percentage and for the smoother air traffic movement. It is also important to analyze at what time of the day, more aircrafts can be allowed to be flown from the airport.

There are possibly two types of data structures which can be implemented on Spatial dataset (GIS). The foremost is Vector and the other is Raster. The vector representation of the model is shown in Figure 1. In our scenario vectors are preferred over raster for following reasons:

- Vectors are composed of points, lines and polygons and the given dataset can be model based on these three attributes. We considered 'Polygon' as Airport Area, 'Lines' as Airplane Trajectory and 'Points' are the co-ordinates (Spatial attributes) of the Airplane at one instance of time.
- Vector data can easily render geographic features with great decision, but it also cost high processing time and complex data structures.
- Each node, label, vertex, line is stored with its own attributes, hence makes model complex, but best suitable when dealing with geospatial data.
- Raster datasets are composed of rectangular arrays of regularly spaced square grid cells. It is usually preferred when dealing with 'Continuous' or 'Image' dataset. Since our dataset is neither of them, so is not suitable to use Raster Modelling.
- In Raster data structures, grids are formed which aren't required in our data modelling.
- The goal of our modelling is to find the nearest neighbor point (another airplane) to reduce the chance of collision while allowing maximum traffic to flow, hence raster data structure won't yield better results as those works well with continuous data.
- In Raster datasets, cell size remains constant, hence it does not have absolute co-ordinate's locations, requiring more computation and storage to calculate any decisions (nearest neighbor in this case).

In the conceptual model for airport dataset – Figure 2., it can be seen that trajectory of an aircraft can be found using *'Lines'*, spatial attributes like longitude, latitude, angle, heading etc. are represented by *'Points'* which helps in identifying the position of an aircraft at particular instance of a day, and the runway region can be denoted by *'Polygon'*.

The model has been designed such that it can easily be expanded if more spatial attributes be added to the given dataset. The modelling is not done using raster data structure, hence worrying about size of the grid when new data is added is not the matter of concern.

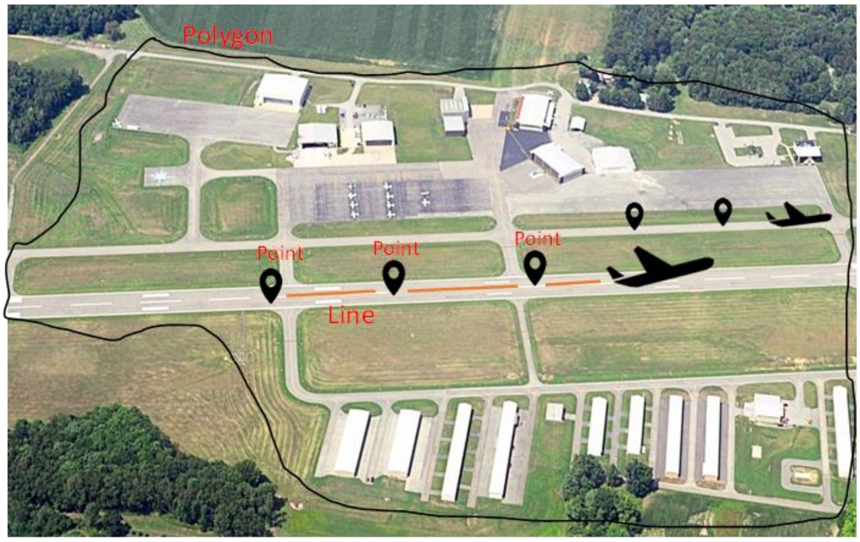


Fig. 1. Vector Representation of the model

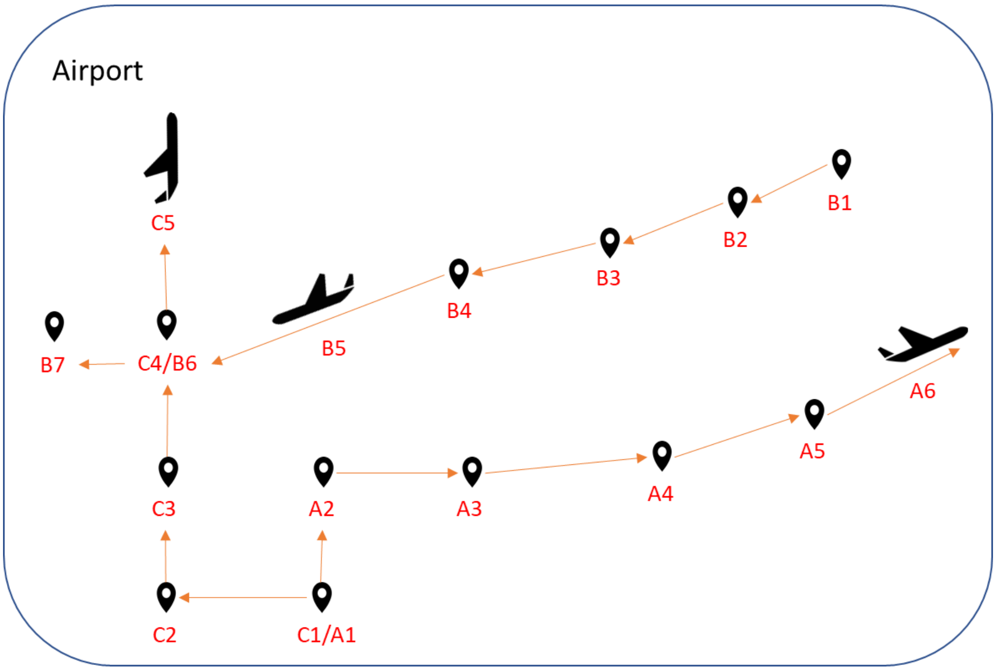


Fig. 2. Conceptual model for Airport Dataset

2.2 Logical Data Model

Logical Data Model adds details about data structures being used when implementing the model. The three common data structures used with spatial data, each having its own advantages over other are:

- R-tree
- Quad-Tree
- KD-Tree

KD-trees are space portioning data structures for representing points in K-Dimensions. They are a type of data structure used to efficiently describe our data. KD-trees aid in the organization and partitioning of data points based on specified parameters. For our dataset, we have represented the data using KD-Tress and used it for query retrieval. It is preferred over R-Tree and Quad Tree for following reasons:

- R-Tree partition the data into rectangles (Area), while KD-Tree do binary split (Points). Binary Split points are also disjoint while R-Tree could have overlapped area.
- R-trees can hold rectangles and polygons, whereas k-d-trees can only hold point vectors (as overlap is needed for polygons) which is required in our case.
- Quad-tree always split data in all the dimensions, which increases the computation and empty cells in the data structure. Since our spatial dataset has 6 dimensions, so it will take lot of time to form a tree with many empty child nodes.
- The problem is to find nearest neighboring co-ordinate to specific co-ordinates, which is most efficient in KD-Trees as it doesn’t have any empty cells as in quad tree.
- Searching in KD-Tree is easier as compared to quad tree as the closest object might be placed right on the other side of a division between nodes. KD-trees on the other hand, allows implementation of very efficient nearest-neighbor search.

While implementing KD-Tree, temporal dimension has been converted to numerical dimension, so that it can be considered as attribute. For converting the temporal data, differential time in ns (nanoseconds) are considered as numerical dimension. It also helps in saving the space in the model.

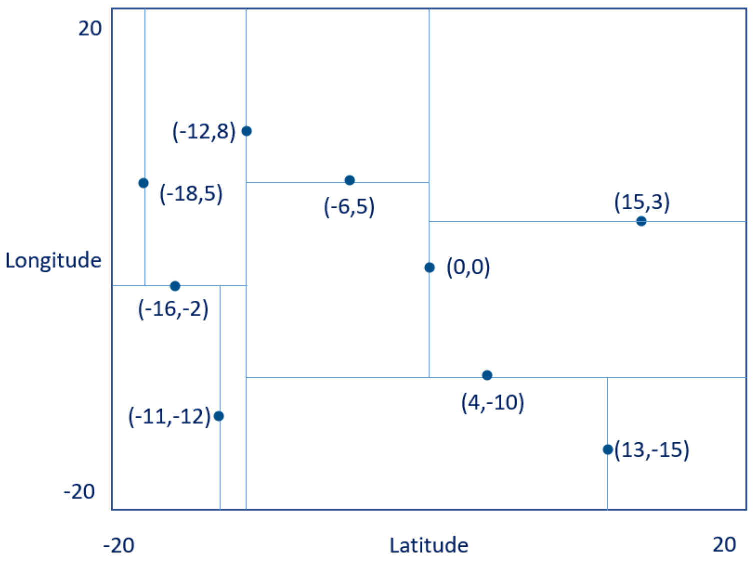


Fig. 3. Logical Data Model (considering 2-D) for Airport Dataset (KD-Trees)

The above diagram in Fig. 4. gives an overview of Logical Data Model for the given dataset. It has been displayed in 2-D, but when implementing 6 spatial dimensions are considered. In KD-Tree, alternatively tree split on each dimension as it can be seen in Fig. 5 below. Using binary search in 2-D case and exhaustive search in N-D case, KD-trees are useful to find the nearest neighbor, also allowing to limit the tree based on ranges. Therefore, below logical model has been implemented in our analysis.

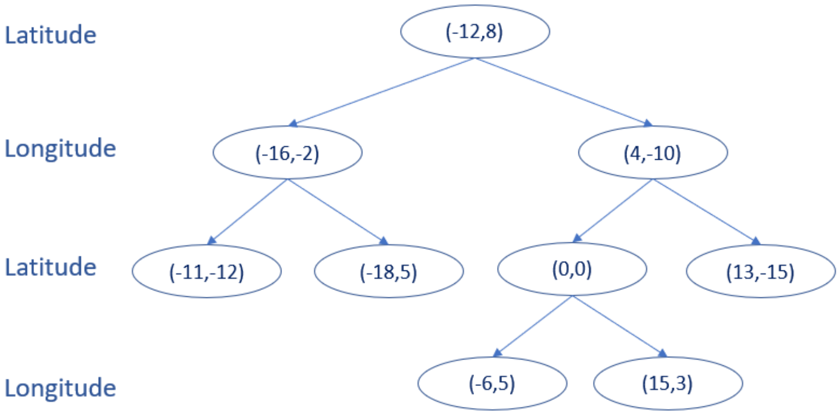


Fig. 4. KD-Tree Representation for 2-D (Latitude and Longitude)

3. Exploratory Data Analysis

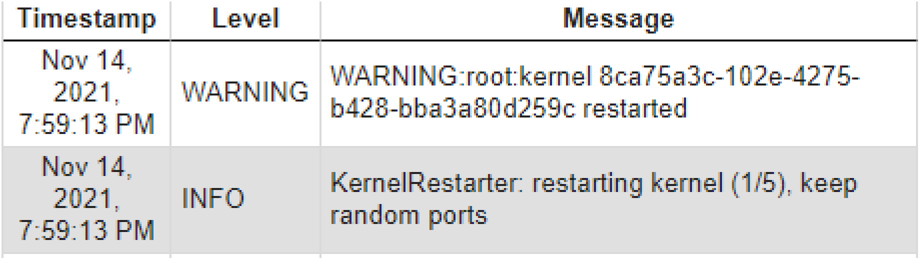
4. Scalability

Processor – Intel(R) Core(TM) i7-4510U CPU @ 2.00GHz   2.60 GHz

RAM – 8GB

Tool/software – Jupyter Notebook and Google Collab

Language – Python 3.7



Timestamp	Level	Message
Nov 14, 2021, 7:59:13 PM	WARNING	WARNING:root:kernel 8ca75a3c-102e-4275-b428-bba3a80d259c restarted
Nov 14, 2021, 7:59:13 PM	INFO	KernelRestarter: restarting kernel (1/5), keep random ports

Number of Games	**Time Taken by Dictionary**	**Time Taken by Array**
100	0.89 seconds	0.92 seconds
1,000	9.73 seconds	9.34 seconds
10,000	109 seconds	106.59 seconds
50,000	586 seconds	525.98 seconds

Table 1:Time taken to parse n records(games)

Observations:

- It has also been observed that when other task is open in the background, along with the execution, then the parsing time is more by almost 60%.