Realtime Visualization

Clima Canves

Emil Daub (12143524), Yannik Gaebel (12208157)

January 17, 2024

Contents

1 Introduction

The Clima Canvas application addresses the visualization of the Dataset, Earth Surface Temperature Data dataset, covering 3000+ cities from 1743 to 2013. Due to the dataset's size, we employ the Quadtree technique for efficient rendering. This report outlines the design and functionality of our application, emphasizing the integration of the extensive dataset and the use of Quadtree for optimal visualization.

2 Implementation

The implementation is divided into two key steps: preprocessing, involving the initial data preparation, and the development of the web application.

2.1 Preprocessing

For preprocessing we performed some data aggregation steps to reduce the data size as well as seasonal data imputation to fill missing values. The preprocessing was done in the notebook data_preprocessing.ipynb. We used the GlobalLandTemperaturesByCity.csv data which has the size 532 MB.
Since there was a significant proportion of missing values we implemented a strategy for data imputation. Table 1 shows the temperature values for a city in denmark that contain missing values. There is a clear seasonal trend visable in the data.
For this reason we used seasonal decomposition from statsmodels to fill the missing data. The results can be seen in table 2
The next transformation entailed aggregating all temperature values for each city into an array. This compressed the amount of data to under 100 MB which simplified the further use.
Finally, we added the Latitude and Longitude values with a geocoder because they did not seem to be correct in the dataset. The data was saved in processed_climate_data.csv.

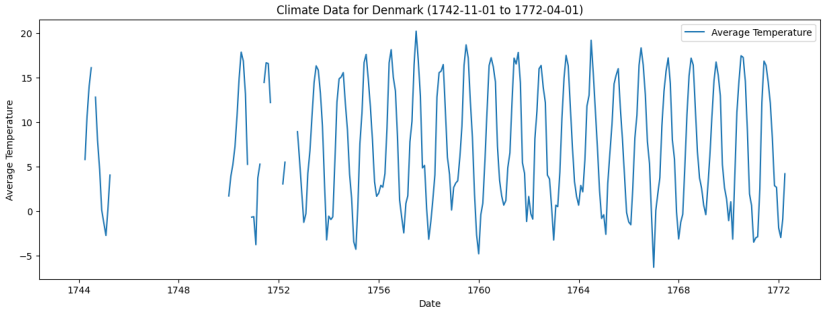

Figure 1: Plot of temperature values of Arhus

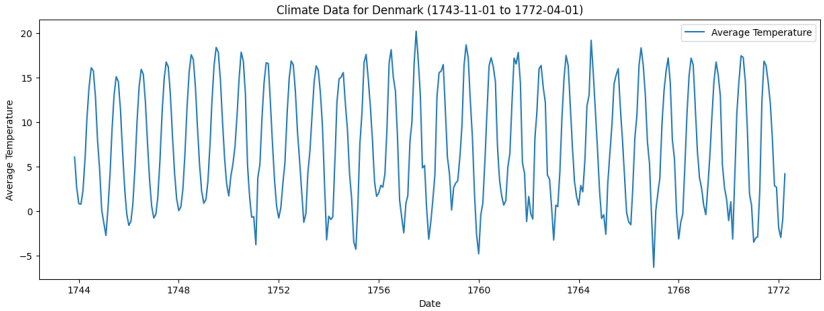

Figure 2: Plot of imputed temperature values of Arhus

2.2 Web Application

Our web application is developed using Node.js, with HTML for information display and JavaScript for data manipulation. Upon launching the app, our preprocessed dataset is loaded, populating the Quadtree and displaying initial information. The application features points on the map, where the color reflects the temperature, also displayed in the popup on hover, for the selected year/month using an adjustable slider, the slider can also be controlled by the arrow keys for finer adjustment. As the map is moved or zoomed, new data is efficiently loaded thanks to the Quadtree, ensuring a seamless experience without user wait times. The Quadtree utilizes latitude and longitude for spatial indexing, and the Leaflet library aids in determining the displayed viewport from the upper-left to the lower-right corners.

2.2.1 Quadtree

The Quadtree employed in our application is encapsulated within the QuadtreeNode class, serving as an efficient spatial data structure for organizing and querying geographical points. This implementation follows a recursive subdivision approach, dynamically dividing the geographical space into quadrants as necessary. Insertion of Points

Insertion of Points
The insert method handles the addition of points to the Quadtree. If a node has not reached its maximum capacity (maxPoints), the point is simply added. However, upon exceeding this capacity and having no subnodes, the node undergoes subdivision (subdivide method), redistributing existing points into the newly created quadrants.

Subdivision
The subdivide method partitions the node into four quadrants, each represented by a child node. This recursive division process continues as the Quadtree grows, maintaining an organized hierarchy of spatially related data.

Index Calculation

The getIndex method determines the quadrant index for a given point based on its latitude and longitude. This index is crucial for correctly placing the point within the appropriate quadrant during insertion and querying operations.

Querying Range
The queryRange method retrieves points within a specified geographical range. It efficiently traverses the Quadtree, checking for overlaps between the query range and each node's bounds, thus optimizing the search for relevant data points.

Bounds Overlap and Point Inside Range
The boundsOverlap method checks if the bounds of a node overlap with a specified range, ensuring that only relevant nodes are considered during querying. Additionally, the pointInsideRange method determines whether a point lies within a given range, aiding in the accurate identification of points to include in the query results.

This Quadtree implementation enhances the speed and efficiency of our Clima Canvas application by providing a scalable and organized structure for handling the extensive Climate Change dataset.

3 User Guide

Map Navigation
Upon launching Clima Canvas, a map is displayed with a slider above it. The map operates like any standard mapping application. You can drag the map to explore different regions and use the zoom functionality to get a closer or broader view. These familiar controls provide a seamless and user-friendly experience.

Data Visualization
Points are plotted on the map to represent various cities, each carrying valuable temperature information. The color of these points corresponds to the temperature of the selected year and month, indicated by a slider above the map. This dynamic visualization allows you to observe temperature variations across different geographical locations.

Interactivity
Hover over a plotted point to reveal a pop-up with detailed information. This includes the city name and the exact temperature value, offering a precise understanding of the climate data. The interactive pop-up enhances your exploration by providing instant insights into specific data points.

Time Selection
Use the slider above the map to select the desired year and month. As you adjust the slider, the temperature points on the map update in real-time to reflect the chosen timeframe. This feature enables you to observe how temperatures have evolved over different periods, adding a temporal dimension to your exploration.