

TNM048—Information Visualization

Spatio-temporal Exploration

January 31, 2018

Kahin Akram Hassan
kahin.akram.hassan@liu.se

1 Introduction

Spatio-temporal data can describe a position of an object or an event in space over a period of time. Traffic, weather or environmental data can be expressed in space and time. In order to make sense of the data, interactive exploration techniques can be used for filtering over space, time and other dimensions. Furthermore, due to the size of the data, data mining methods are often needed to extract further knowledge from large databases.

2 The Setup

The lab will be done using D3.js version 4 JavaScript library. Download the file lab3.zip from the course homepage, the code and file structure is similar to the previous labs.

3 The Data

For this lab the dataset used is a collection of earthquake events across Europe for the period 1987-2016. The data set has been retrieved from <http://earthquake.usgs.gov/earthquakes/search/> and is stored in a comma separated value file (lab3/static/data/data.csv). Each data point represents an earthquake and is described by a unique id, time-stamp, latitude, longitude, depth, magnitude and the name of the place where the earthquake occurred. The data are ordered by their time of occurrence.

Task 1:

Once downloaded, extract the compressed folder "lab3" and familiarize yourself with the file structure and the data.

4 Temporal focus+context via brushing

The focus-plus-context technique is often used to focus on some parts of the data, meanwhile still having the rest of the data showed as context for orientation.

The following example ([Link here](#)) demonstrates how to use D3's brush component to implement focus + context zooming.

Task 2:

Modify the code in area.js to visualize the earthquake data using F+C paradigm.

When done with this task you should see the graph.

5 Geo-Spatial visualization

The most direct way to represent geo-referenced data is to plot the data points on a map. For geo-spatial visualizations, D3 provides components for displaying and manipulating geographic data. These components use the GeoJSON format for representing geographic features..

Task 3:

In the draw function add `.enter().insert(path)` on the variable `countries` and then add `class country, path d, id` to be `d.id`, `title` to the name of the country and `fill`. You should now see the world map.

Task 4:

Now complete the code in function `geoFormat()` so that it reads in the data in GeoJSON format in order to see the points on the map

5.1 Filtering

Task 5:

Filtering by time - write the code in the public method `filterTime()` so that when the brushing in the F+C component is used the points on the map are filtered according to the specified time window. Don't forget to send `x.domain()` to the filter function from the brush function in `area.js`.

Task 6:

Filtering by attribute - write the code in the function `filterMag()` so that it filters the points according to the slider value. Don't forget to call in it `d3.select` on line 97

5.2 Clustering

Task 7:

Write the code in the public method `cluster()` so that it calls the k-means function written in the previous lab,(Remember, we are working with depth and mag here) and change the color of the data points according to the clustering classification. The clustering algorithm should to be applied on the filtered data.