WebPlotViz: Browser Visualization of High Dimensional Streaming Data with HTML5

Supun Kamburugamuve, Pulasthi Wickramasinghe, Saliya Ekanayake, Chathuri Wimalasena and Geoffrey Fox
School of Informatics and Computing, Indiana University
Bloomington, IN, USA

***Abstract*— Large volumes of high dimensional streaming data are increasingly becoming common place and the ability to project these data into three dimensional space to visually inspect them is an important capability. Algorithms such as Multidimensional Scaling (MDS) and Principal Component analysis (PCA) can be used to reduce high dimensional data into a lower dimensional space. In this paper we present 1. MDS based approach to project high dimensional time series data to 3D with automatic rotation to align successive data segments 2. Open source commodity visualization of three dimensional time series in web browser based on Three.js, and 3. An example based on stock market data.**

I. Introduction

WebPlotViz[1] is a software toolkit for generating, analyzing and visualizing high dimensional time series data in 3D space as a sequence of 3D point plots. WebPlotViz consists of a web portal for viewing the points in 3D, a highly efficient parallel implementation of Multidimensional Scaling (MDS) [1] for generating 3D points from large data sets and a generic workflow and tools to prepare data for processing and visualization. The input to the system is a set of items with different values at different time steps. The timestamp is used to segment the data into time windows. The time windows can be calculated by various methods such as sliding windows or accumulating times. A distance metric between each pair of data point in a data segment is chosen and these distances are projected to 3D using the MDS algorithm. Before running the MDS program, the data needs to be cleansed and prepared to suite the algorithms' requirements.

Dimension reduction is ambiguous up to rotations, translations and reflections and so when MDS is applied to a series of data segments in a time series data set, the resulting 3D points in consecutive plots are not aligned by default. We introduce a simple approach (termed rotations) to find the transformation to best align mappings that are neighbors in time. The MDS and rotation technology is well established but it is non trivial to use in this particular instance with many choices as to the details of application. Optimization is involved in both MDS and rotation stages; we weight the terms in objective functions by a user defined function. The result of MDS and rotations is a time series consisting of a set of vectors (one for each data entity) that are defined at each time value in time series. We developed an HTML5 (based on Three.js) viewer for such 3D time series which greatly helps understanding the behavior of time series data.

This paper describes the initial results of a study of the structure of financial markets viewed as collections of securities generating a time series of values using the WebPlotViz. This study should be viewed as an examination of technologies and approaches and not a definitive study of structure. For example, we look at only one set of US securities over a 13 year time period with daily values defining the time series.

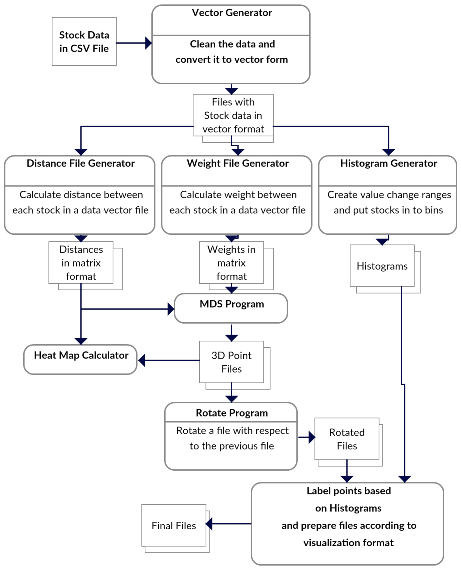
Stock Data in CSV File
Vector Generator
Clean the data and convert it to vector form
Files with Stock data in vector format
Distance File Generator
Calculate distance between each stock in a data vector file
Weight File Generator
Calculate weight between each stock in a data vector file
Histogram Generator
Create value change ranges and put stocks in to bins
Distances in matrix format
Weights in matrix format
Histograms
MDS Program
Heat Map Calculator
3D Point Files
Rotate Program
Rotate a file with respect to the previous file
Rotated Files
Label points based on Histograms and prepare files according to visualization format
Final Files

Fig. 1. Data processing workflow with stock data

II. Data Processing Workflow

We run all the steps including pre-processing, data analytics and post processing steps in a scripted workflow as shown in Figure 1. We started with MPI as the key technology to implement the data processing and a HPC cluster to run the workflow. Later we implemented the same data processing steps in Hadoop for better scalability and data management. All the programs in the workflow are written using Java and the integration is achieved using bash scripts. The MDS and Levenberg-Marquardt Rotation algorithms are Java based MPI applications, running efficiently in parallel and are not implemented in Hadoop. In the future we would like to use Apache Beam as an unifying platform to execute the workflow on different big data systems.

Pre-processing steps mainly focus on data cleansing and preparing the data to suite the inputs of the MDS algorithm. These data are run through the MDS algorithm to produce the 3D points. There can be high numbers of data segments for a data set depending on the time window and shift in time chosen. For each of these data segments we first put the data into a vector form where i^{th} element represents the

[1] https://spidal-gw.dsc.soic.indiana.edu/

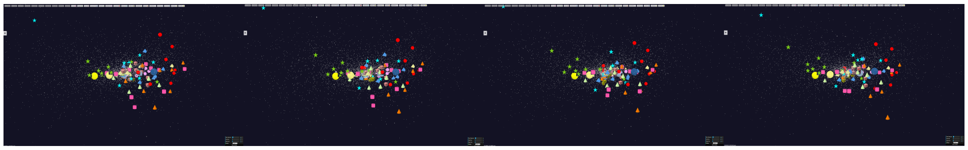

Fig. 2. Four consecutive plots from stock visualization

value at i^{th} time in that window. For each vector file we created, a distance matrix file and weight file is calculated according to user defined functions. The $(i,j)^{th}$ entry of these files contain the pairwise distance between i and j vectors. At the next step MDS is run on each data segment. To avoid communication overheads, we dont use the full HPC cluster for individual MDS runs as each MDS only invovles up to 7000 points (our code scales up to millions which run efficiently on large clusters). Instead here, multiple MDS instances are executed in parallel for different data segments.

Each MDS Projection is ambiguous to rotations, reflections and translation, which was addressed by a least squares fit to find the best transformation between all consecutive pairs of projections. This transformation minimizes the distances between the same data points in the two plots. Because two consecutive data segments can have different set of points, we find the transformation with respect to the common points between the two consecutive segments and apply this to the rest of the points. Next the plots are prepared for visualization. The points in the plots are assigned to clusters according to user defined classifications. Colors are assigned to clusters and finally the files are converted to the input format of the visualizer.

III. WebPlotViz

WebPlotViz is a HTML5 based viewer for large scale 3D point plot visualizations. It uses three.js [2] JavaScript library for rendering 3D plots in the browser. It is designed to visualize sequence of time series 3D data frame by frame as a moving plot. The 3D point files along with the metadata are stored in a NoSQL databases and allow scalable plot data processing in the back end. The user can use the mouse to interact with the plots displayed, i.e. zoom, rotate and pan. Also user can edit and save the loaded plots by assigning colors and special shapes to points for better visualization. WebPlotViz also supports single 3D plots including point plots and trees. The online version has many such example plots preloaded.

WebPlotViz is also a data repository for storing the plots along with their metadata and includes functions to search and categorize plots stored. The source code of WebPlotViz, data analysis workflow and MDS algorithm are available in DSC-SPIDAL github repository[3].

IV. Stock market data analysis

The data is obtained through the The Center for Research in Security Prices (CRSP) database through the Wharton Research Data Services (WRDS) web interface, which makes daily security prices available to the Indiana University students for research. The data can be downloaded as a csv file containing records of each day for each security for a given time period. We have chosen the CRSP data because it is being actively used by the research community and readily available free for research. The data we obtained includes roughly about 6000 7000 securities for each year. The number is not an exact one for all data segments because securities are added/removed from the stock exchanges and because we do data cleansing for each segment separately. The Pearson Correlation between the stock vectors is primarily used to calculate distances between securities in the Security Position Space.

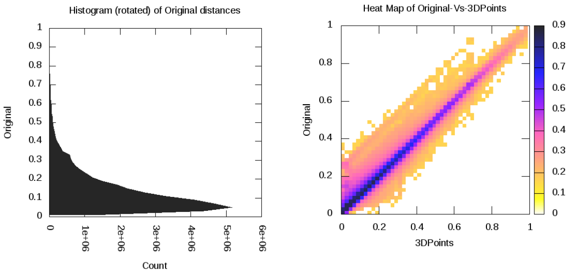

Fig. 3. Distance distribution and heat map

We considered daily stock prices from 2004 Jan 01 to 2015 Dec 31. We examine especially changes over one year windows (the velocity of the stock) and the change over the full time period (the position of the stock). The data can be considered as high dimensional vectors, in a space – the Security Position Space – with roughly 250 times the number of years of components. We map this space to a new three dimensional space – the 3D Mapped Space – (dimension reduction) for visualization. With a year period and a 7 day shift sliding window approach there are about 570 data segments each generating a separate 3D plot. Figure 2 shows four consecutive 3D plots of the stock data and Figure 3 shows the accuracy of the MDS mapping as a heat map. The dark values along the diagonal, indicate that most of the original distances are mapped to 3D distances accurately.

Acknowledgment

This work is partially supported by NSF 1443054 grant as it used data analytics from the SPIDAL library. Further we would like to extend our gratitude to FutureSystems team for their support with the infrastructure. We would like to thank Milinda Pathirage for his contributions to WebPlotViz.

References

[1] Ekanayake, S. , Kamburugamuve, S., Fox, G.: SPIDAL Java: High Performance Data Analytics with Java and MPI on Large Multicore HPC Clusters. http://dsc.soic.indiana.edu/publications/hpc2016-spidal-high-performance-submit-18-public.pdf (2016), Technical Report

[2] http://threejs.org

[3] https://github.com/DSC-SPIDAL