International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 02 | Feb-2016 www.irjet.net p-ISSN: 2395-0072

Data Visualization and Communication by Big Data

T. Gnana Prakash[1]

[1]*Assistant Professor, CSE Department, VNR VJIET, Hyderabad, TS, India*

Abstract - *Data visualization is viewed by many disciplines as a modern equivalent of visual communication. It is not owned by any one field, but rather finds interpretation across many (e.g. it is viewed as a modern branch of descriptive statistics by some, but also as a grounded theory development tool by others). It involves the creation and study of the visual representation of data, meaning "information that has been abstracted in some schematic form, including attributes or variables for the units of information". Data visualization is both an art and a science. The rate at which data is generated has increased, driven by an increasingly information-based economy. Data created by internet activity and an expanding number of sensors in the environment, such as satellites and traffic cameras, are referred to as "Big Data". Processing, analyzing and communicating this data present a variety of ethical and analytical challenges for data visualization. The field of data science and its practitioners called data scientists have emerged to help address this challenge. In this project real time data is considered for the analysis.*

Key Words: **Data visualization, Big Data, Python, Real time Data and visual data**

1. INTRODUCTION

Due to the technical progress of the last decades, which enables the production of small sized micro processors and sensors at a low cost, a new research area has developed, dealing with wireless sensor networks. These networks mostly consist of a large amount of tiny sensor nodes, which combine sensing, data processing and communicating components. To ensure these three functions the nodes feature a number of sensors, a micro computer and a wireless communication device.

The general purpose of such a sensor network lies in its deployment in or very close to a phenomenon a user wants to observe. After a network is deployed the mere act of sensing includes the following three working steps. Step one is the measurement of a physical property by one of the sensors. The second step involves the micro computer which computes the data delivered from the sensor depending on the desired result. In a third step the computed data has to be transmitted from the sensor node to its destination, where it is often stored in a database. Fig 1 shows how data from sensor node A could get to the user.

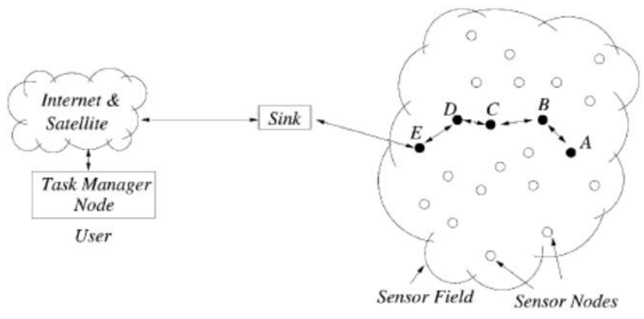

Fig. 1: Sensor nodes Scattered in Sensor Sink

After several hops inside the sensor field the sent information reaches a so-called sink, which communicates with a task manager node via internet or satellite. The sensor network itself is thereby a self-organizing network with a certain protocol stack used by the nodes and the sink. After these three steps, which have been enquired rather widely follows a fourth step, which is rather poorly explored, namely the visualization of the sensor data. Just seeing the raw data of a sensor network stored in a database mostly does not fulfill the needs of the users. So the data need to be analyzed and shown to the user in a way, where information can easily be gained from them. Which kind of information can be gathered from a sensor network, depends on the application area the network is used in.

2. VISUALIZING SENSOR DATA

A sensor can be defined as "a device that receives a stimulus and responds with an electric signal whereby stimulus is the quantity, property or condition, that is sensed". Sensor networks consist of a large amount of tiny sensor nodes. The following section therefore will deal with sensors by looking at several of their aspects. It will be shown different classifications of sensors, ways to gather data from sensors and the relevance of the position of a sensor and the time of its measurement. At the end of the section an existing sensor network will be introduced.

All the different kinds of sensors share the same purpose. They ought to respond to a physical property by converting it into an electric signal, which can be operated on to produce an output. How this physical conversion is managed inside the sensor node is of no importance for our goal of visualization. The importance lies in the delivered data such as for example the momentary temperature at the sensors position.

Depending on the application there are different ways to gather data from the sensors. By having a look at the applications themselves, they can roughly be separated into two parts: Analysis and event detection. In an analytical application the user wants to access data at a certain time, to watch the status of a phenomenon. Beneath the data measured by sensors two other values have to be considered as important. On the one hand a user of a sensor network wants to know, where the sensor, that delivers the data, is to be found.

After a closer look at sensors and some principles of information visualization both topics have to be combined to achieve a visualization of sensor data. This section therefore shows how to extract useful data out of the sensor data and how to simplify the choice of a suitable visualization. At the end the visualization of the existing sensor network will be shown. When it comes to the visualization of sensor data the first question to be asked is: Which data shall be shown? In a sensor network that delivers multiple continuous data, it is nearly impossible to show all the data. At this point the needs of the user have to be considered. For example when monitoring a factory process a user is interested in abnormal data like a pressure value that is too high.

For the extraction of useful knowledge out of raw sensor data there can be used data mining techniques. But before this technique can be applied, the data need to be prepared to increase efficiency. After choosing the relevant data there has to be found a visualization to present these data efficiently.

Due to the enormous variety of application areas it is hard to assign a special visualization to a certain kind of sensed data. Instead the data are classified to narrow the range of possible visualizations and thereby simplify the choice. The sensed data always have more than one dimension. As mentioned in position and time of a measurement always play a certain role when analyzing the data. So the columns of the table represent the spatial-temporal dimensions of sensor data. The temporal aspect is thereby divided into momentary, which means an instantaneous on demand value, and continuous, which means values of a certain time period, while the spatial aspect is separated into relative and absolute values. The columns are therefore split into four parts, as they are divided twice. The rows of the table stand for the dimensions of a sensor, which are based on the number of different sensing components a sensor can own. They can differ from 1- to multidimensional. The entries of the table consist of the data types of Shneidermans data type taxonomy. This means, that a certain n-dimensional sensor with regard to the temporal and spatial aspect belongs to one of these data types. Examples for visualizations of the different data types can be found in Shneidermans paper. The temporal data type is not included, as the temporal aspect is encoded as a further dimension of the data.

As can be seen, the dimension of sensor data grows by one, when regarding the temporal progression of the value, and grows by two, when regarding the absolute position of a value, as the absolute position is thought to be given by a x- and a y-value not regarding the z-axis of space. Depending on the final dimension of the sensor data, there has to be designed a suitable visualization that fits the needs of the user? For example a temperature sensor in a factory, that monitors the temperature of a machine to ensure its functionality, would be a 1-dimensional sensor (= temperature) with a relative position (= machine x) and a continuous measurement. Its data are therefore, 2-dimensional and could be shown in a line-graph.

3. PYTHON

Python is a multi-paradigm programming language: object-oriented programming and structured programming are fully supported, and there are a number of language features which support functional programming and aspect-oriented programming (including by met programming and by magic methods). Many other paradigms are supported using extensions, including design by contract and logic programming. Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. An important feature of Python is dynamic name resolution (late binding), which binds method and variable names during program execution.

The design of Python offers only limited support for functional programming in the Lisp tradition. The language has map (), reduce () and filter () functions; comprehensions for lists, dictionaries, and sets; as well as generator expressions. The standard library has two modules (iter tools and fun tools) that implement functional tools borrowed from Haskell and Standard ML.

Python's developers strive to avoid premature optimization, and moreover, reject patches to non-critical parts of CPython which would offer a marginal increase in speed at the cost of clarity. When speed is important, Python programmers use PyPy, a just-in-time compiler, or move time-critical functions to extension modules written in languages such as C. Cython is also available which translates a Python script into C and makes direct C level API calls into the Python interpreter.

4. PLOTLY

Plotly is an online analytics and data visualization tool, headquartered in Montreal, Quebec. Plotly provides online graphing, analytics, and stats tools for individuals and collaboration, as well as scientific graphing libraries for Python, R, MATLAB, Perl, Julia, Arduino, and REST. Plotly was built using Python and the Django framework, with a front end using JavaScript and the visualization library D3.js, HTML and CSS. Files are hosted on Amazon S3.

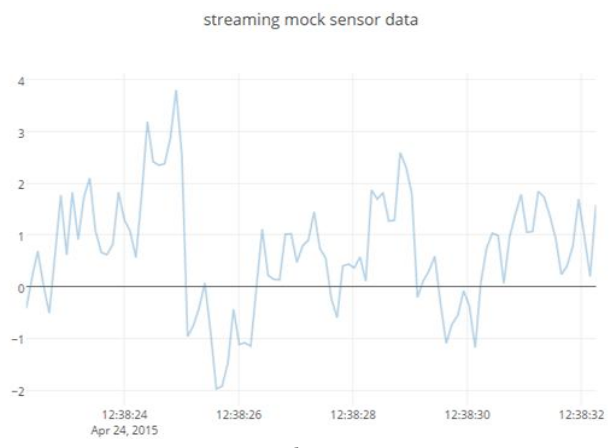

Fig 1: Sensor data projection

5. TEST CASES AND DISCUSSIONS

Table -1: Test Cases for Data

Test Cases	Expected Output	Actual Output	Priority
sensor data received consistently	clean flow of data without any discrepancies	data flow with missing values	high
cleaning of the data	data modified into a continuous steam	data cleaned successfully	low
storage of the data	storing data in an excel sheet	data copying errors	high
linking the data stream to plotly	successful integration in plotly	integrations issues and no proper data streaming	high
posting of the data on to the website	successful graph on the website	continuous streaming	-----
providing data dynamically to the program	the data is accepted and the corresponding graph is generated	the graph is generated successfully	-------
other plots using the same data	different plots like bar graph or scatter plot	various kinds of graphs are generated	-----
plotting with extreme values	an error or an out of the bounds graph	the graph is generated but not visible	-----
Adding values to the generated plot	An error or modified graph	An error that says the generated plot cannot be further modified	----

6. CONCLUSIONS

Through this interactive and enjoyable project on data visualization, we could learn the real time issues in data cleaning as well as visualizing. This project provided deep insights into the real world of data visualization and how effective a piece of junk data also can be made something very interesting for analysis and understanding.

ACKNOWLEDGEMENT

We are very much thankful to our Principal and Management for providing all facilities like research labs and software's to carry out this work in college.

REFERENCES

Usama M. Fayyad, Andreas Wierse, Georges G. Grinstein "Information Visualization in Data Mining And Knowledge Discovery" Morgan Kauffman publishing, 2002

Jason W. Osborne "Best Practices In Data Cleaning" John Wiley & Sons, 2012

Nathan Yau "Visualize This" John Wiley & Sons, 2011.

Martin C. Brown “Python: The Complete Reference” International Journal on Osborne, 2001, pp: 67.

Buja, Andreas “Interactive Data Visualization” Journal on Volume On Visualization, Vol. 15, 2014, pp: 153-163.

Chi, Ed H. “A taxonomy of Visualization techniques” National Journal on Information Visualization, Vol: 3, 2010, pp: 69-75.

Ganti, V. Kaushik, R., Chaudhuri,S.” A primitive operator For Data Cleaning And Visualization” Journal on Data Engineering, Vol: 22, 2009, pp: 5.

https://plot.ly/python/streaming-tutorial/

https://docs.python.org/2/tutorial/

http://www.learnpython.org/

BIOGRAPHY

Gnana Prakash.T received his Bachelors of Technology degree in 2006 and Masters in 2010 from Jawaharlal Nehru Technological University, Hyderabad. He has more than 8 years of teaching experience Presently working as Assistant Professor in the department of CSE, in VNR VJIET, Hyderabad and currently pursuing Ph. D. from JNTU, Kakinada. His Research interest areas are Mobile Ad Hoc & Sensor Networks, Image Processing, Computer Vision, etc.