VisDiscovery

Visualization recommendation engine

Computer and Communications Department

Team members:

1. Ali Rashad
2. Esraa Hefny
3. Nada Essam
4. Passant Ibrahim
5. Omar Rostom
6. Yehia Arafa

Team leader:

Yehia Arafa, yehiaarafa11@gamil.com

Supervisors:

- Dr. Mohamed Bassyouny.
- Dr. Mohamed Saad, msaad@vt.edu

VISDISCOVERY

Visualization recommendation engine

○ Project description

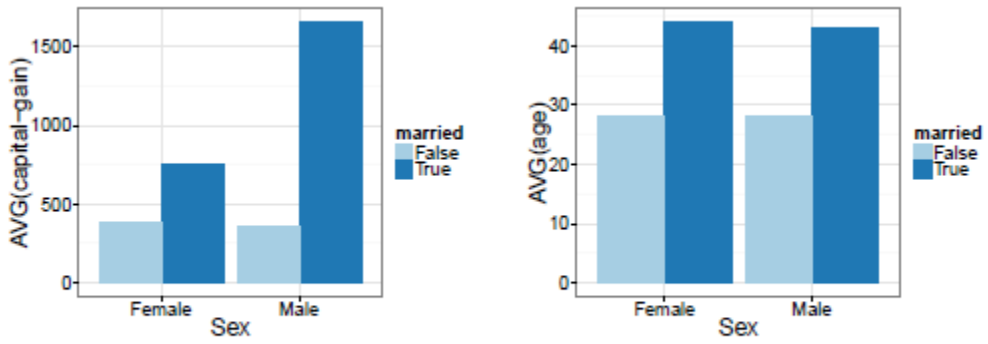
Abstract:

Data analysts often build visualizations as the first step in their analytical workflow. However, when working with high-dimensional datasets, identifying visualizations that show relevant or desired trends in data can be laborious. Our graduation project introduces VisDiscovery, a visualization recommendation engine, that recommends interesting visualizations to facilitate fast visual analysis: given a subset of data to be studied, VisDiscovery intelligently explores the space of visualizations, evaluates promising visualizations for trends and recommends those it deems most "useful" or "interesting", helping data scientists and analysts connecting things in interesting ways and look at data from different angles.

Project details:

Data visualization is about conveying data as efficiently as possible, data scientists and analysts always use visualization techniques to explore patterns in their data and show their result effectively. Given a new dataset or a new question about an existing dataset, an analyst builds various visualizations to get a feel for the data, to find anomalies and outliers, and to identify patterns that might merit further investigation. However, when working with high-dimensional datasets, identifying visualizations that show interesting variations and trends in data is non-trivial: the analyst must manually specify a large number of visualizations, explore relationships between various attributes and combinations, and examine

different subsets of data before finally arriving at visualizations that are interesting or insightful. This means that he needs to manually specify and examine every visualization that could possibly be done on his data.
Our engine - VisDiscovery - tackle the problem of automatically identifying and recommending visualizations which the data analyst will find it interesting for visual analysis.
Recommending visualizations is the fact that whether a visualization is interesting or not depends on a host of factors. VisDiscovery adopt a simple criterion for judging whether the visualization is interesting or not: a visualization is likely to be interesting if it displays large deviations from some reference, that deviation can often guide users towards visualizations they find interesting. And so we used deviation as our utility metric to decide whether a certain visualization is interesting or not.
For example;
If a data analyst wants to know how marital-status impacts socio-economic indicators like education and income, among others. He uses the US Census data to conduct his analysis comparing unmarried US adults to married US adults.
The analyst came with these two charts:



(a) Interesting Visualization *(b) Uninteresting Visualization*

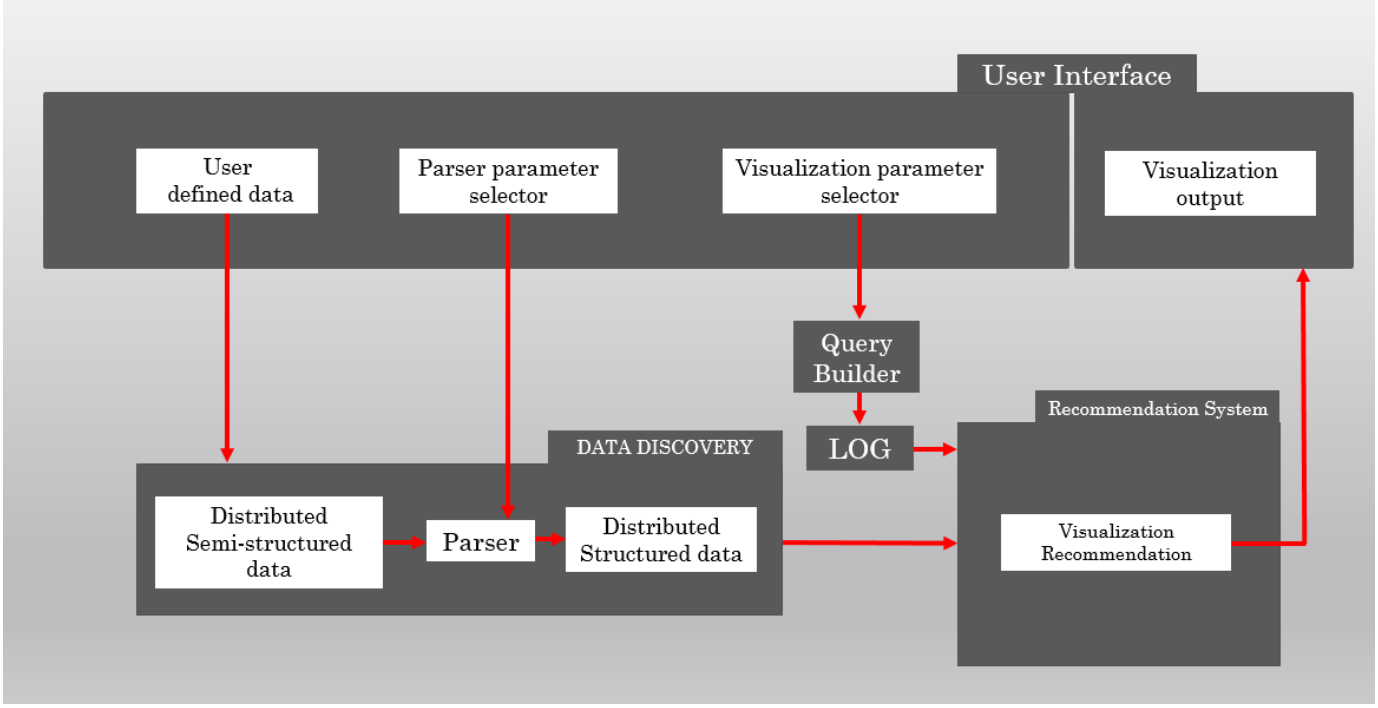
we know that for this particular task, analysts find the visualization in (a) interesting since it depicts a property of the data that is different for unmarried adults compared to married adults. Specifically, the chart depicts that although capital gain for female and male unmarried adults is approximately equal, capital gain for female, married adults is only half that of male, married adults. This also

shows that (b) is a visualization that the analyst will not find interesting. Note that this chart depicts that age does not show different trends between unmarried and married adults.

What we mean by deviation bases utility metric that; visualizations which show trends in the target data (unmarried adults) that deviate from trends in reference data (married adults) are potentially interesting to analysts.

Our system can be connected on any database system running SQL queries, but on addition to that our system have the feature to operate on semi-structured data [CSV, Jason, XML], as those type of data are the most common types any data analyst uses. Our system first converts the semi-structures data into structured data as those in any DBMS which resides in a fixed field to be simple and easy to use with SQL queries, later we run our many operations on top of this dataset.

Architecture:



○ Challenges

In the previous example the data analyst wanted to know how marital-status impacts socio-economic indicators like education and income, among others. For instance, he may build a chart showing average income as a function of marital status, visualize marital status as a function of education, plot the correlation with race and gender, visualize impact on hours worked per week, and so on. Depending on the types of visualizations created, the number of possible visualizations grows exponentially with the number of indicators in the dataset. As a result, creating and examining all possible visualizations quickly becomes intractable.
The two major obstacles we encountered while implementing our idea:

(1) Operating on a very large data.
(2) Responding within interactive time scales.

VisDiscovery operates on a large scale data, that we could not even store on a single machine. So we stored our data in a distributed manner using **HDFS-Hadoop file system.**

Usually any user will need the system to have a good interactive time and since we make a lot of computations on a very large data set to recommend those interesting visualizations, we have had to make optimizations to our system:

- First, we used parallel SQL queries execution on these distributed data by using **Cloudera Impala [SQL-on-Hadoop]** which boosted our performance in executing and running SQL queries.

- Secondly, we had to decrease the number of times we issue and fetch the data using the SQL queries, despite the fact that we used parallel SQL queries execution. So we used shared-based optimization, meaning that we intelligently merge and batch queries, combining multiple queries in a single one, reducing the number of queries issued to the database and in turn, minimizing the number of scans of the underlying data.

- Technologies used

Summary

We can summarize our project in these points:

- We build a system that recommends top k interesting visualizations to the data analyst, in order to help him discover more things about his data.
- Our system operates on top of any DBMS.
- Our system has the advantages of operating on semi-structured data by converting it to a structured one.
- Our system operates on a very large scale data distributed over a cluster of nodes to make computations faster.