# Data Visualization in Educational Datasets using a Rule-Based Inference System

Aniruddha Desai

Project Report for Course TCSS 702

Institute of Technology
University of Washington, Tacoma

Masters of Science in Computer Science and Systems

2014

Committee Chair
Dr. Ankur M. Teredesai, Institute of Technology

Advisors
David Hazel, Institute of Technology
Dr. Gregory Benner, Center for Strong Schools

# TABLE OF CONTENTS

Table of Contents ........ 1

Abstract ........ 1

1 Introduction ........ 2

2 Related Work ........ 2

3 Mathematical basis ........ 3

3.1 Rule-based Inference: ........ 4

3.2 Supervised Learning: ........ 4

4 Identifying Constraints ........ 4

5 Rule-based Inference ........ 9

6 Algorithm ........ 10

7 Implementation ........ 11

8 System Architecture ........ 12

9 Experimental Testing ........ 12

10 Results Of Testing ........ 13

11 Lessons Learned ........ 13

12 Future Work ........ 14

13 Conclusion ........ 14

14 Acknowledgements ........ 14

15 References ........ 14

## ABSTRACT

**Dynamic data visualization can be a very useful analytical tool for discovering insights in complex real-world data sets with high dimensionality and large variety of data types. We leverage publicly available data sets from Washington State's Public Education System to demonstrate usefulness of data visualization as a tool for analysis and effective decision making. We created "querybuilder", a web-based interface which allows the user to generate ad-hoc queries. Our online inference system efficiently generates dynamic visualizations for user-specified queries. We then address the question of how to select an appropriate visualization type that would be best suited for the result of a specific query on the given data set. The main motivation for this project is developing a rule based inference system to automatically select the appropriate visualization type.**

***Keywords*— data visualization; high-dimensionality; rule-based inference; web-based data analytics**

## 1 INTRODUCTION

Washington State's public education system publishes annual reports for student performance metrics, drop-out / graduation rates, data on instructors' credentials, demographic data among other educational statistics [1] spanning about ten years. There is a big interest among both educational policy makers, parents, teachers and the general public to analyze this dataset for the important factors that influence student achievement. For example, effectively delivered high-quality continuing professional development can be a major factor influencing student outcomes [2, 3]. The dataset we analyze and visualize in this project contains data attributes such as teachers' professional development, student learning outcomes, special education students' performance among others. While this is a very robust data set, with opportunities for providing insights to parents, educators, and administrators alike, the data is often not easily consumed or understood without significant manipulation of the data source with sophisticated data preprocessing systems. The data has a high degree of variety with diverse data types consisting of numerical data such as test scores, percentages; categorical data such as demographic groups, boolean values; and geo-spatial data of different granularities such as district, county, city etc. The motivation behind this project is to develop a visualization system which would allow users unfamiliar with the analyzing education data sets to get insights into this high-dimensional and high-variety dataset by easily interacting with it and getting quick, effective and automated visualizations to represent individual dimensions targeted with ad-hoc user-defined queries. We use a web-based drag and drop interface for efficiently generating visualizations which is easy to use for teachers, parents, educational policy makers or even the general public. We extend the "querybuilder", an easy to use GUI-based online tool for the creation of ad-hoc queries that we developed in an earlier phase of this project, for developing the data visualization system. This project focuses on solving the problem of selecting the appropriate visualization type for a specific ad-hoc query generated by the user for a particular dataset. Our approach to solve this problem uses a rule-based inference system.

## 2 RELATED WORK

After an extensive review of cutting edge visualization technologies that make interaction with and visualization of data quick and easy, we have identified several deficiencies in the current visualization landscape. Several open source libraries and API's such as d3.js (Data-Driven Documents) [4] and Open Street Maps [5] exist, but these need considerable development of software architecture and back-end integration before they can be used in the manner we envisioned for our system. Tableau Software [6] provides a robust and efficient system for generating quick visualizations but being a proprietary product it does not expose a lot of its API for customization. We were able to access Tableau desktop's free (with student account) functionality for manually generating visualization "dashboards" to improve our understanding of appropriate visualizations for query results within our educational datasets. As such, we found the current state of the art in data visualization did not have any open source customizable tools that were capable of changing visualization type in response to ad-hoc query results on complex, high variability real-world data sets [6, 7]. Overall, we found d3.js to be among the more useful open source APIs in terms of the relative ease of harnessing its powerful and expressive graphics by integrating it within our web-based data visualization system.

## 3 Mathematical basis

The problem of selecting the most appropriate visualization for a specific set of attributes is NP-Hard. We provide a proof to support this as follows:

*Proof:*

The problem of selecting the best visualization type involves two aspects – one is to search if a solution exists and the other is to decide on an optimal solution after we know that there exists at least one such solution. To analyze the hardness of the search problem we consider all the possible configurations of variables in our dataset that are of significance in the decision making process. Let us assume this is done in an initial step of setting up our theoretical problem. Let Q be the set of all the possible user generated queries. Let C be the set of boolean variables that correspond to some conditions or "constraints" being met or otherwise in a specific query with regards to the different clauses and the fields in the clauses. For instance, these variables could be "field in SELECT CLAUSE is numeric", "field in WHERE CLAUSE is geo-spatial" or "the field in GROUP BY aggregates results chronologically in ascending order" and so on and so forth. Let R be the set that denotes all the possible valid rules that can govern the selection of visualization types depending on the nature of a specific query. Finally, let V be the set of all possible visualization types that could be available to our inference engine. We can then define our problem "VIZ" as one of searching if an "appropriate" visualization exists for any arbitrary query submitted to the inference engine. We say that for any query that represents an instance of VIZ we return "yes" if a visualization exists and "no' if it does not.

*Reduction:*

We can now show that the SATISFIABILITY (abbreviated SAT) problem, which is known to be NP-complete [8] can be reduced in polynomial time to VIZ. The SAT problem is stated as follows:

SAT = {F | F is a boolean formula with a satisfying assignment}
Where boolean formula F comprises of an arrangement of boolean literals in a generalized CNF (conjunctive normal form) for example:
$\Phi(x_1, x_2, x_3, x_4) = (x_1 \vee \neg x_2) \wedge (x_2 \vee x_3 \vee \neg x_1) \wedge \ldots \wedge (x_4 \vee \neg x_3)$

An instance of SAT involves a specific assignment of the variables for which formula F would return true if it is satisfiable and false otherwise. We can now map every clause in the SAT to every query in set Q for VIZ. We can map individual literals within each clause in SAT to the three sets of variables in VIZ. We map literals to an arbitrary number of individual constraints in our set C in VIZ. We can map other literals to an arbitrary number of boolean variables corresponding to whether or not certain rules from our set of rules R in VIZ are obeyed by a given configuration of constraints' values. Another set of literals would be mapped to the types of visualizations we have available from our set V in VIZ. Since the number of units of work involved will be linear in the total number of literals in the GCNF formula such a mapping can be done in polynomial time. Now, we can state for an instance of SAT when a certain assignment of literals returns a true value with formula F we can come up with a new formula V(F) in polynomial time, that will return “yes” (a visualization exists) for the corresponding (mapped) instance of our VIZ problem.

This shows that the SAT problem is reducible in polynomial time to VIZ, which makes the basic search problem we considered initially NP hard. This analysis proves that the problem of selecting the appropriate visualization type for an ad-hoc query is an NP-Hard problem.
Since the problem is NP-hard and an exact algorithm cannot be found for it that is tractable for complex real-world data sets (assuming P ≠ NP), we need to use an approximation algorithm in our approach to solve this problem. Two well-known approaches for developing approximation algorithms [9, 10] are:

### *3.1 Rule-based Inference:*

In this approach a set of rules or constraints is developed. Based on these we infer the appropriate type of visualization for a particular query. These rules are derived from heuristics and patterns observed in the dataset. The set of rules can be extended and modified as needed to improve the quality of inferences generated.

### *3.2 Supervised Learning:*

This involves using as a “training set” a dataset that we know contains appropriate choices of visualization types as “ground truth” values correctly paired with the corresponding ad-hoc queries. We then build prediction models and train them on this ground truth data using established machine learning algorithms such as SVM [11]. Then, we can use these models on the “unknown” dataset to make inferences. The initial “training data” could be developed by first making some rule-based inferences and manually validating their correctness by checking if the visualization types are appropriate for the queries for which they were generated.

We focused on the rule-based approach in this project as the first stage of this project and intend to make the supervised learning approach the basis for future work.

## 4 IDENTIFYING CONSTRAINTS

We studied the effect of using different visualization types for a given query to identify constraints that could be used to decide on the most suitable of all visualization types available to us. At this stage, we have three types of visualizations implemented in the system – Bar Charts, Pie Charts and Maps. We now present some illustrations of observations we made.

*Type of Visualization Field:* If the visualization field is a numeric data type we try to visualize it by counting data points for each value as illustrated by Fig 1a. Interestingly, this does not work well to adequately visualize the data and we find ourselves in need of a method of aggregation as discussed further under “*Aggregation Field Choice”.* In contrast, if the visualization field is categorical we are able to directly visualize it by simply counting the data points in each category as Fig 1b illustrates.

*Query 1a: SELECT temp.PercentWhite, COUNT(0) as value FROM (SELECT alldemoschool.PercentWhite FROM alldemoschool WHERE alldemoschool.`District` = 'Tacoma School District' ) AS temp GROUP BY temp.PercentWhite ORDER BY value DESC;*

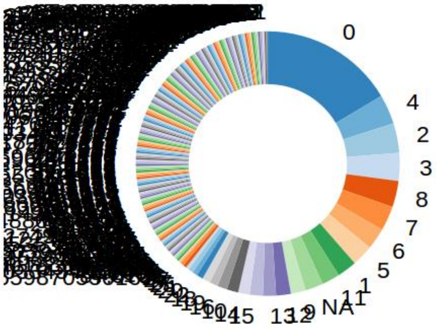


*Figure 1a – Pie Chart Visualization for Query 1a with numerical field – too many data points – requires aggregation.*

*Query 1b: SELECT temp.Amao3MetTarget, COUNT(0) as value FROM (SELECT amaobyschool.Amao3MetTarget FROM amaobyschool WHERE amaobyschool.`District` = 'Auburn School District' ) AS temp GROUP BY temp.Amao3MetTarget ORDER BY value DESC;*

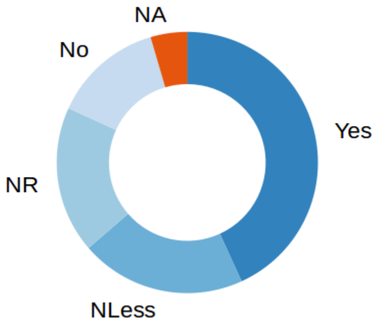


*Figure 1b – Pie Chart Visualization for Query 1b with a categorical field.*

*Number of Results – Pie Chart vs. Bar Chart:* As illustrated in Fig.2a & 2b, when the number of rows in the query result exceeds a threshold, pie charts were not as effective as bar charts for data visualization. The same query result is used to generate both visualizations for comparison.

*Query 2: SELECT temp.School, avg(temp.PercentTeachersWithAtLeastMasterDegree) AS value FROM (SELECT * FROM alldemoschool WHERE alldemoschool.`County` = 'Kitsap' ) AS temp GROUP BY temp.School;*

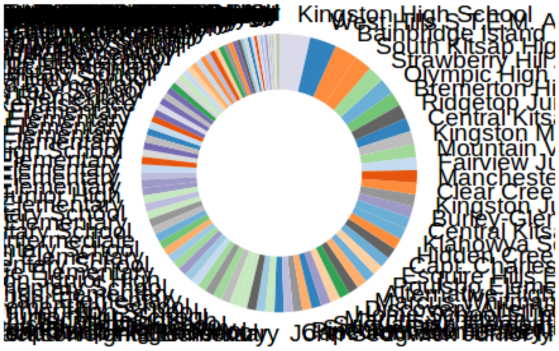


*Figure 2a – Pie Chart Visualization for Query 2 is too busy.*

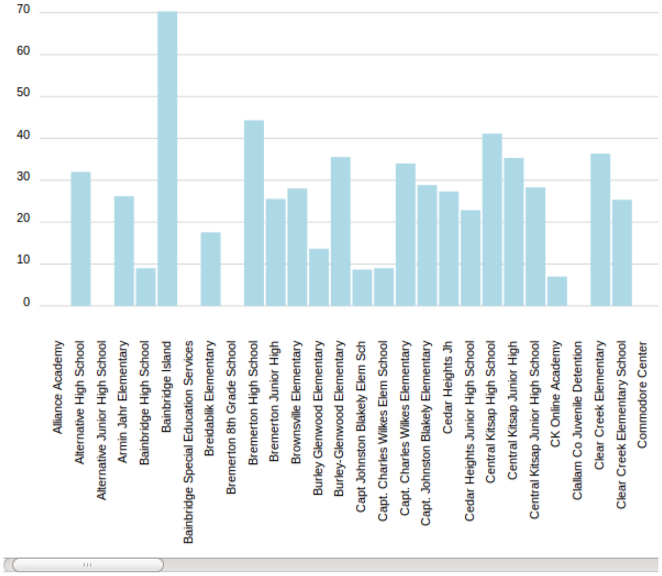


*Figure 2b – Bar Chart Visualization for Query 2 is more effective.*

*Geo-Spatial Fields – Map vs. Bar Chart:* As shown in Fig. 3a and 3b, when the group by field is identified as a geo-spatial type (such as “county” or “district”) a map could be a more effective visualization than a bar chart. Maps have a desirable property of making all the data fit into a compact view – where as a bar chart, in contrast, would require the user to scroll sideways to look at all the bars in the entire visualization.

*Query 3: SELECT temp.County, avg(temp.TotalEnrollment) as value FROM (SELECT * FROM alldemoschool WHERE alldemoschool.`SchoolYear` = '2002-03' ) AS temp GROUP BY temp.County;*

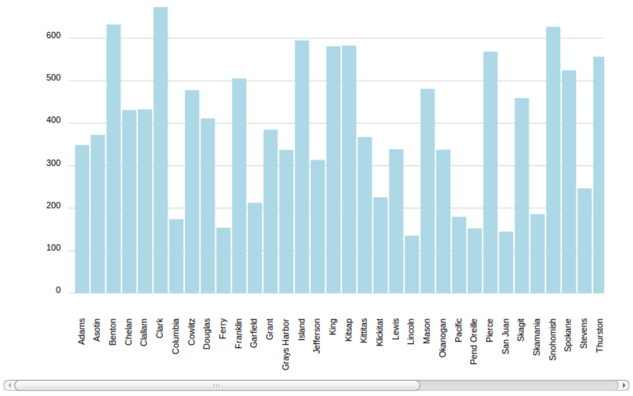


*Figure 3a – Bar chart for Query 3 requires scrolling to see all geo-spatial data points (counties) in the result.*

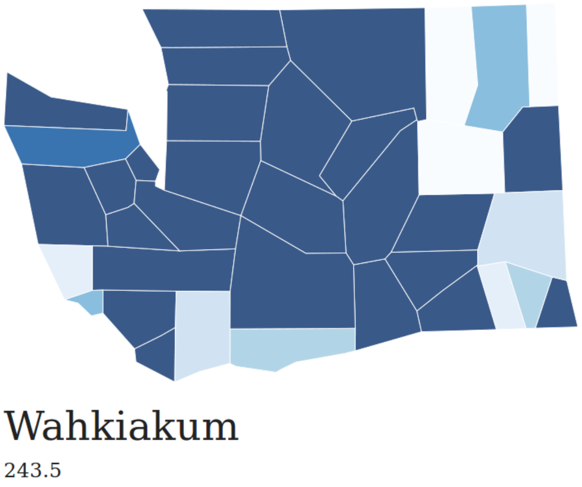
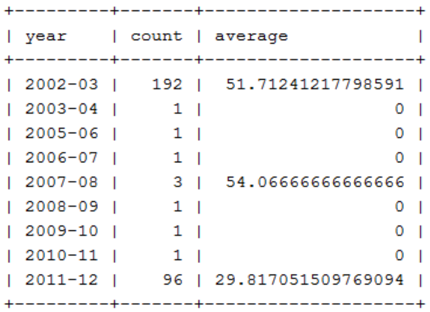


*Figure 3b – A Color-coded Map (chloropleth) for Query 3 is a more compact and effective data visualization choice (a compact legend, not in the picture, associates the darker tones to higher values).*

*Aggregation Field Choice:* When the selected visualization field is numeric we need a method of aggregating it to create a visualization showing data variability across meaningful groups. A simple strategy is to discretize the data and show a distribution of data points across intervals. We found a more interesting way to visualize the data is to automatically infer a suitable aggregation field for using as a "group by" dimension in the query. We use a constraint based approach for this. We identify all categorical dimensions in the tables we are referencing in our query. We compare them to find a field such that when it is used to aggregate the visualized field, the least missing data or nulls values are encountered in all the groups in the query result. We use Fig 4a and 4b for illustration. When we group by X field we find majority of data are concentrated in a few groups while the other groups are empty on account of high incidence of missing values or nulls. Instead, when we group by Y field the data is better distributed over all the groups.

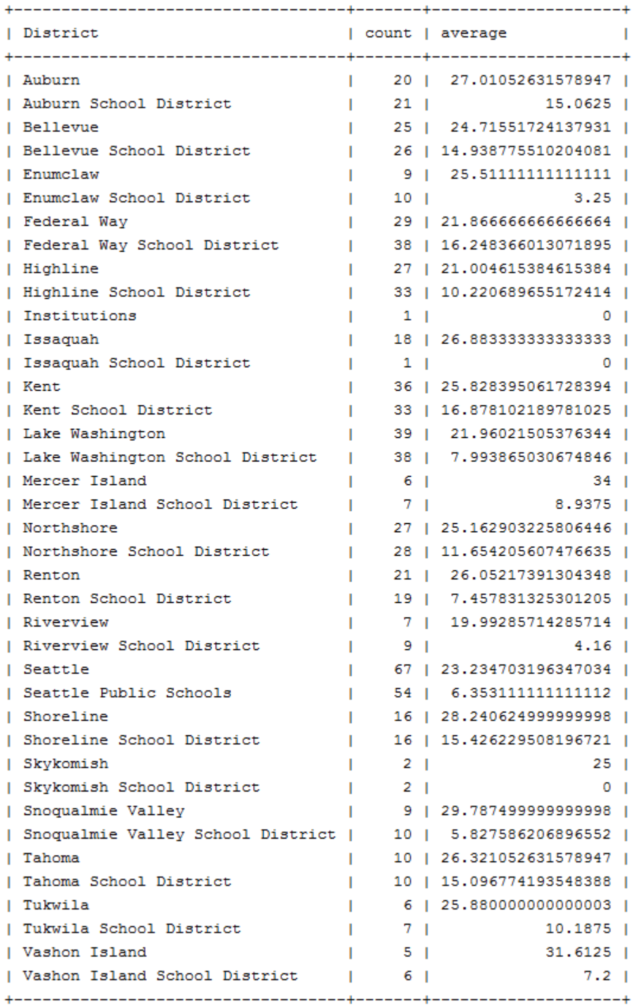
*Query 4a: SELECT SchoolYear AS year, count(distinct PercentTeachersWithAtLeastMasterDegree) AS count, avg(PercentTeachersWithAtLeastMasterDegree) AS average FROM alldemoschool WHERE County = 'King' GROUP BY SchoolYear;*



```
+---------+-------+--------------------+
| year    | count | average            |
+---------+-------+--------------------+
| 2002-03 |   192 |  51.71241217798591 |
| 2003-04 |     1 |                  0 |
| 2005-06 |     1 |                  0 |
| 2006-07 |     1 |                  0 |
| 2007-08 |     3 |  54.06666666666666 |
| 2008-09 |     1 |                  0 |
| 2009-10 |     1 |                  0 |
| 2010-11 |     1 |                  0 |
| 2011-12 |    96 | 29.817051509769094 |
+---------+-------+--------------------+
```

*Figure 4a – Query 4a result shows data for some groups are empty due to missing values or nulls.*

*Query 4b: SELECT District, count(distinct PercentTeachersWithAtLeastMasterDegree) AS count, avg(PercentTeachersWithAtLeastMasterDegree) AS average FROM alldemoschool WHERE County = 'King' GROUP BY District;*



```
+-----------------------------------+-------+--------------------+
| District                          | count | average            |
+-----------------------------------+-------+--------------------+
| Auburn                            |    20 |  27.01052631578947 |
| Auburn School District            |    21 |            15.0625 |
| Bellevue                          |    25 |  24.71551724137931 |
| Bellevue School District          |    26 | 14.938775510204081 |
| Enumclaw                          |     9 |  25.51111111111111 |
| Enumclaw School District          |    10 |               3.25 |
| Federal Way                       |    29 | 21.866666666666664 |
| Federal Way School District       |    38 | 16.248366013071895 |
| Highline                          |    27 | 21.004615384615384 |
| Highline School District          |    33 | 10.220689655172414 |
| Institutions                      |     1 |                  0 |
| Issaquah                          |    18 | 26.883333333333333 |
| Issaquah School District          |     1 |                  0 |
| Kent                              |    36 | 25.828395061728394 |
| Kent School District              |    33 | 16.878102189781025 |
| Lake Washington                   |    39 |  21.96021505376344 |
| Lake Washington School District   |    38 |  7.993865030674846 |
| Mercer Island                     |     6 |                 34 |
| Mercer Island School District     |     7 |             8.9375 |
| Northshore                        |    27 | 25.162903225806446 |
| Northshore School District        |    28 | 11.654205607476635 |
| Renton                            |    21 |  26.05217391304348 |
| Renton School District            |    19 |  7.457831325301205 |
| Riverview                         |     7 |  19.99285714285714 |
| Riverview School District         |     9 |               4.16 |
| Seattle                           |    67 | 23.234703196347034 |
| Seattle Public Schools            |    54 |  6.353111111111112 |
| Shoreline                         |    16 | 28.240624999999998 |
| Shoreline School District         |    16 | 15.426229508196721 |
| Skykomish                         |     2 |                 25 |
| Skykomish School District         |     2 |                  0 |
| Snoqualmie Valley                 |     9 | 29.787499999999998 |
| Snoqualmie Valley School District |    10 |  5.827586206896552 |
| Tahoma                            |    10 | 26.321052631578947 |
| Tahoma School District            |    10 | 15.096774193548388 |
| Tukwila                           |     6 | 25.880000000000003 |
| Tukwila School District           |     7 |            10.1875 |
| Vashon Island                     |     5 |            31.6125 |
| Vashon Island School District     |     6 |                7.2 |
+-----------------------------------+-------+--------------------+
```

*Figure 4b – Query 4b result has data evenly distributed.*

## 5 Rule-based Inference

Once we identify constraints and criteria that favorably impact the choice of visualization types in various situations we use them to generate rules. We formulate the following rules by combining constraints and check for applicable rules when every query result is processed by the system:

1. *Data Type of Visualization Field:* The first constraint we use is the data type of the primary visualization field. If the field chosen by the user is categorical we can easily aggregate the data by applying GROUP BY on the same field and use the counts of data points in each group. If the data type of the visualization field is numerical, we aggregate the data on a suitable field (which we discuss further in item 4 below).

2. *Number of Results:* If the number of results (number of groups when the data is aggregated) is less than 15, we select a pie chart and otherwise we select a bar chart. With some experimentation we notice that above this threshold the pie chart visualizations start to lose their effectiveness.

3. *Data Type of Aggregation Field:* We select a map if the aggregation field is geo-spatial (such as "County" or "City") and if it is non-spatial categorical field we fall back to selecting either pie chart or bar chart depending on the number of results criterion listed above.

4. *Choice of Aggregation Field:* If the visualization field is numerical we will need to select a suitable way to aggregate the data before generating the visualization. We select one of the categorical fields in the data that has the fewest null / missing values and provides an even distribution of the data points in the query result. Since choice of underlying aggregation field impacts the choice of resulting visualization type, it is an integral part of the inference engine.

## 6 ALGORITHM

The inference engine uses the following algorithm:

```
if Visualization Field is Categorical
  if field is geo-spatial
     generate a map
  else
     if number of results after group by ≤ 15
        generate a pie chart
     else
        generate a bar chart

else /* Visualization Field is Numeric */
  infer or select a method of aggregating data:
     if categorical field with the least missing  or null values (i.e. most
     populated field) is identified by inference engine
       group-by field is auto selected and used to generate visualization
     else
       discretize the data in the visualization field into ten intervals while
       allowing the user to change the number of intervals, and use intervals for
       aggregating data
     if user wants a different group-by field
       allow user to select a categorical "group by" field of their choice

   Use aggregation to compute averaged values of numeric data for each group; select
   the appropriate visualization type as follows:
     if group-by field is geographic
       generate a map
     else
       if number of results after group ≤ 15
          generate a pie chart
       else
          generate a bar chart
```

The algorithm can be further extended to incorporate more types of visualizations as well as a more diverse set of constraint-based rules designed to handle a wider range of scenarios for which we wish to infer ideal visualizations types.

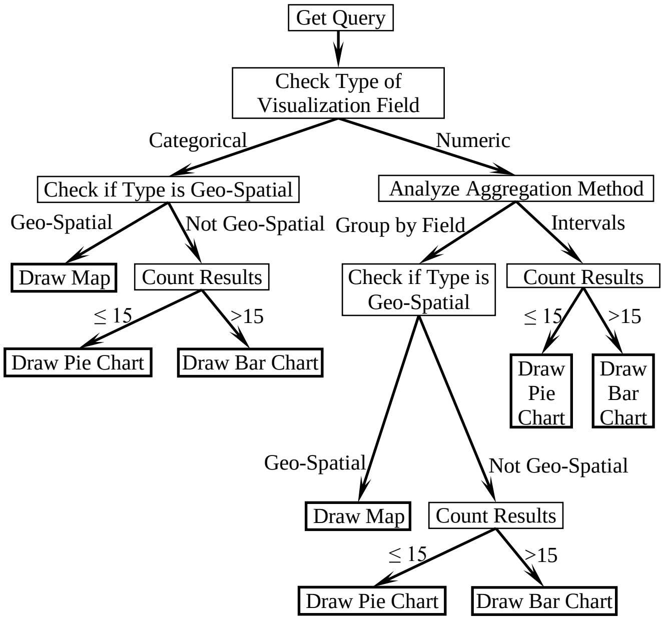


*Figure 5 – Graph of Rule-Based Inference Process*

Each path from root to leaf in this decision tree represents a sequence of constraints in the form of a rule. Therefore, currently eight rules are implemented in our inference engine. We intend to continue adding more constraints and developing more rules as we gain more insights into what factors contribute to better selection of visualization types.

## 7 IMPLEMENTATION

The main steps of system implementation are:

1. Pass the visualization field (SELECT clause of the resulting query), localization field (WHERE clause) and / or aggregation field (GROUP BY clause) obtained from the querybuilder user interface as parameters to the underlying rule-based inference engine.
2. Develop a rule-based inference engine that efficiently selects the appropriate data visualization type based on parameters supplied to it and automatically displays visualization.
3. Give the user the ability to change:
    - Localization field
    - Aggregation field or discretization option
    - Visualization type
4. Collect and record data on the correctness of the decisions made by the inference engine by tracking user feedback. Store results in a database table as the User ID, Query, Type Selection and a Boolean Flag for whether User considered the visualization type was appropriate or not. We intend to use this feedback data in the next phase of the project involving supervised learning techniques.

# 8 SYSTEM ARCHITECTURE

We developed a web-based data visualization system with an Apache 2.0 webserver, a MySQL database, and a server-side web application written in PHP. The system uses the following two JavaScript libraries:

- JQuery to generate U/I widgets that create the easy-to-use drag and drop functionality, and
- Data Driven Documents or “d3.js” [4] to create powerful visualizations with SVG (Scalable Vector Graphics) elements in HTML.

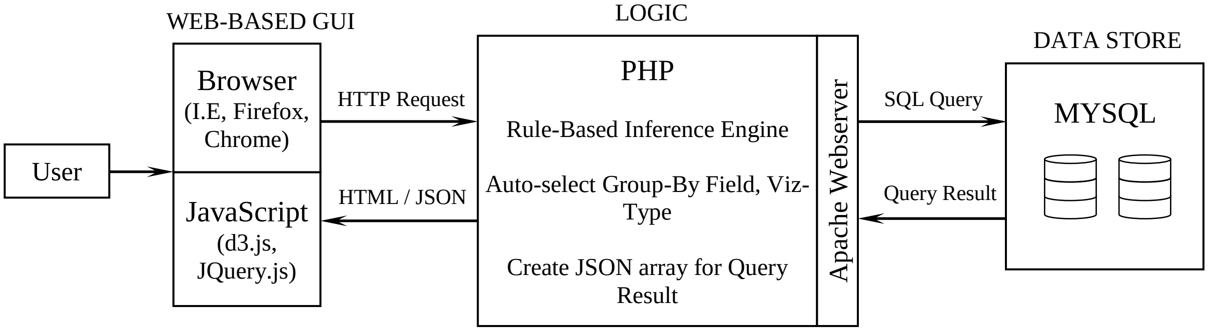


*Figure 6 – Schematic System Architecture*

# 9 EXPERIMENTAL TESTING

We designed a simple experiment to test the rules (below) which we implemented in the inference engine. The experiments are meant to measure how effective each rule was in inferring the most suitable visualization when the necessary set of constraints are satisfied to make the rule applicable.

Rule 1:
*{Vis. Field is Categorical, Vis. Field is Geo-spatial}:* ***Map***
Rule 2:
*{Vis. Field is Categorical, Vis. Field is Not Geo-spatial,*
*Result Row Count for Group By Vis. Field ≤ 15}:* ***Pie***
Rule 3:
*{Vis. Field is Categorical, Vis. Field is Not Geo-spatial,*
*Result Row Count for Group By Vis. Field > 15}:* ***Bar***
Rule 4:
*{Vis. Field is Numeric, Group By Field is Geo-spatial}:* ***Bar***
Rule 5:
*{Vis. Field is Numeric, Group By Field is Not Geo-spatial, Result Row Count for Group By Vis. Field ≤ 15}:* ***Pie***
Rule 6:
*{Vis. Field is Numeric, Group By Field is Not Geo-spatial, Result Row Count for Group By Vis. Field > 15}:* ***Bar***
Rule 7:

*{Vis. Field is Numeric, Group By Field is Not Geo-spatial, Result Row Count for Discretized Vis. Field ≤ 15}:* ***Pie***
Rule 8:
*{Vis. Field is Numeric, Group By Field is Not Geo-spatial, Result Row Count for Discretized Vis. Field > 15}:* ***Bar***

*Testing Procedure:* A group of eight testers performed testing on the system. Each rule was tested five times by every individual as they choose arbitrary fields from the dataset to form the queries they fed into the system. The testers recorded their findings, based on their personal preferences, as to whether the inference engine had made the most appropriate choice of visualization type or not. This yielded a set of forty readings per rule. In Table 1 we show the tabulated results of how each rule performed during these experiments. The results of our experimental testing were mostly positive.

## 10 Results Of Testing

TABLE I. PERCENTAGE OF TIMES AN OPTIMAL VISUALIZATIONTYPE WAS SELECTED BY INFERENCE ENGINE

| Rule No. | Readings | | |
|---|---|---|---|
| | *Total Inferences* | *Correctly Inferred* | *Percentage Correct* |
| 1 | 40 | 23 | 58% |
| 2 | 40 | 31 | 78% |
| 3 | 40 | 28 | 70% |
| 4 | 40 | 23 | 58% |
| 5 | 40 | 14 | 35% |
| 6 | 40 | 31 | 78% |
| 7 | 40 | 22 | 55% |
| 8 | 40 | 29 | 73% |
| **Average Percentage of Correct Inference** | | | 66% |

## 11 Lessons Learned

We found while implementing the system that visualizations could be much more effective if the underlying data were more reliable and consistent. Our dataset had a lot of noise and missing values. This is quite typical of a complex real-world data set and with high dimensionality and a varied set of data types we had to be careful about validity and veracity of the data. We learned that, in the future, it would be greatly beneficial to perform more data validation and cleaning before the visualization stage. Due to the high variety in the data types, we also encountered some challenges while preprocessing it to generate metadata from information schemas. We noted that in the future we could get around this problem by letting our system "guess" the data type as such and allowing the user to change it, if needed, via user feedback.

## 12 Future Work

We have access to another public data set in the education data domain that contains financial data for Washington's public education system [12]. In the future, we plan to connect our system with this dataset as well, to further explore potential applications.

At this stage we have developed a basic set of rules and our system is intended to be extendable by evolving a robust knowledge base through experimental analyses of our initial set of rules. In the future, a probabilistic approach to making inferences can be explored to address the same problem.

As a part of future work, we intend to add more rules based on new constraints to make the inference engine more robust. For example, we plan on implementing a new rule based on the type of user that is logged into an online session on the system to explore whether that user-type constraint can help fine-tune the decision making on visualization type (i.e. education policy makers may be more interested in bar charts for aggregation on a year-by-year basis whereas a parent maybe more interested in a map with information organized by county).

In the next phase of this project, we will follow the supervised learning approach by the using the experimental results of the accuracy of our preliminary rule-based inferences. We will perform more experimentation and testing to generate a more extensive set of "ground truth" data on which we can to train prediction models.

## 13 Conclusion

In this project we described a rule-based approach to the problem of selecting the most appropriate visualization type for a given user-defined query. The system we developed is designed to automatically generate data visualizations powered by a rule-based inference engine. The system is implemented by integrating open source APIs and makes the basis for an extendable platform for data visualization.

## 14 Acknowledgements

The author would like to thank his Professor Ankur Teredesai for envisioning and formulating the problem definition for this work. The author is also grateful for having continuous access to the staff and resources at the Center for Web and Data Science which was made possible by Dr. Teredesai. The author would also like to thank Professor Gregory Benner (of the Center for Strong Schools) for giving him an opportunity to explore educational data while working on the OSPI Project. The author also acknowledges David Hazel (of the Center for Web and Data Science) for extensive support on a range of topics from basic API / implementation-related issues to high level system architecture and design issues in the areas of data storage, pre-processing, analysis and visualization.

## 15 References

[1] (02/20/2014) Washington State OSPI Report Card. Available: http://reportcard.ospi.k12.wa.us/Summary.aspx?year=2012-13.

[2] G. J., Benner, J. R., Nelson, N. C., Ralston, & P. Mooney, “A meta-analysis of the effects of reading instruction on the reading skills of students with or at risk of behavioral disorders”, Behavioral Disorders, 2010, 35(2), 86–102.

[3] D.L. Ball, & D. K. Cohen, “Developing practice, developing practitioners: toward a practice-based theory of professional development”, Darling-Hammond & G. Skyes (Eds.), Teaching as the learning professional: Handbook of policy and practice, 1999, (pp. 3-32). San Francisco: Jossey-Bass.

[4] M. Bostock, V. Ogievetsky and J. Heer, “D3: Data-driven documents”, IEEE Trans. Visual. Comput. Graphics 17(12), pp. 2301-9. 2011. Available: http://dx.doi.org/10.1109/TVCG.2011.185. DOI: 10.1109/TVCG.2011.185.

[5] (02/22/2014) Open Street Maps Wiki. Available: http://wiki.openstreetmap.org/wiki/API.

[6] (02/22/2014) Tableau Business Intelligence Software. Available: http://www.tableausoftware.com/business-intelligence.

[7] V. Chandola, R. R. Vatsavai, and B. Bhaduri, “iGlobe: An interactive visualization and analysis framework for geospatial data”, Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications (COM.Geo '11). ACM, New York, 2011, Available:http://doi.acm.org.offcampus.lib.washington.edu/10.1145/1999320.1999341, Article 21, 6 pages. DOI: 10.1145/1999320.1999341.

[8] S. A. Cook, “The complexity of theorem-proving procedures”, Third Annual ACM Symposium on Theory of Computing, New York (1971), 151-158.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to Algorithms. The MIT Press, 2001.

[10] S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, Algorithms, Berkeley, McGraw-Hill, 2006.

[11] J. Han, M. Kamber and J. Pei, Data mining concepts and techniques. Morgan Kaufmann, 2012.

[12] (02/22/2014). Statewide Longitudinal Data System. Available: http://www.k12.wa.us/Data/.