Browser-based Categorization of Data Towards Automated Visualization

Steven Diviney

May 3, 2013

Abstract

This paper presents a system to automatically generate suitable visualizations given arbitrary data. Data Visualization is an increasingly common technique used to reinforce human cognition. In many areas of human activity the volume of data being generated is increasing rapidly. New methods must be employed to assist in the comprehension of this data. There has been a good amount of research performed to assess what factors contribute to the creation of an effective visualization. Additionally many new and novel visualizations have been created. However, automatic generation of visualizations has received little attention.

Such a tool would help to combine these two areas of research. An understanding of what factors contribute to an effective visualization are encoded into the system presented. Given an arbitrary dataset the system attempts to select the most appropriate visualization. This paper also discusses to what extent this process is viable.

Using the limited amount of information contained in a raw dataset it is possible to select a comprehensible visualization. As the dataset becomes increasingly complex the effectiveness of such a system diminishes. A number of datasets are input to the system and an evaluation is undertaken. The results presented show that such a system can be used to assist users in the creation of suitable visualizations while avoiding the creation of inappropriate or even misleading visualizations.

Contents

Chapter 1

Introduction

10/04/2013
changes in
onenote

This chapter introduces the dissertation topic and explains the motivation behind the work. A research question is purposed and a number of objectives are outlined in an attempt to satisfy it. This is followed by evaluation criteria and finally by a summary of the document structure.

1.1 On Visualization

There are three fields subfields of Data Visualization. The boundaries between them are not particularly distinct and they are referred to somewhat interchangeably in academic literature. There are many other types but these three are of particular interest as their primary concern is the visualization of large volumes of data almost always with the aid of a computer.

1.1.1 Information Visualization

Information visualization is perhaps the most broadly used and can be thought to encompass all of the fields of visualization. The term today is generally applied to the visual representation of large-scale collections of non-numerical information, such as text in a book or files on a hard-disk. The distinction between this definition and that of Data Visualization seems to be quite poor. The type of the data in question is used to distinguish the two. The terms “information” and “data” are not so easily distinguished. Information can be thought of as a level of abstraction above data. The mere fact that a dataset is non-numerical does not identify it as information. A set of ordinal labels or categories is just as meaningless as a set of

numbers. Information is created by organizing such data and presenting it with context.

Information visualization is concerned with representing more abstract topics. A good example is process visualization. Each element in a process visualization represents a complex topic.

1.1.2 Data Visualization

Data Visualization is the science of visual representation of data, defined as "facts and statistics collected together for reference or analysis" (oed31, 2013). As stated the distinction between data and Information Visualization is not very concrete. This distinction is generally not regarded as important but this paper is concerned with Data Visualization. The synthesis of new information through the creation of visual artifacts.

1.1.3 Scientific Visualization

Scientific Visualization is concerned primarily with the visualization of objects in three dimensional space with an emphasis on realistic rendering. This emphasis on realism is primarily what distinguishes it from other forms of visualization. This is not to say that the other forms may distort the data, rather that abstract data does not necessarily have a spatial dimension. How would one realistically visualize the lines of a book? Novel ways must be invented to accomplish this.

1.2 Motivation and Description

Data Visualization is defined as an internal construction in the mind. The field of Data Visualization is connected with creating visual artifacts in order to facilitate individuals in building an internal representation of a dataset (Spence, 2001). Data Visualization is not a new field but it has only become an established area of scientific research in recent years . The volume of data a typical computer can generate and process far exceeds what a human can comprehend. Data Visualization is becoming an increasingly popular technique to aid this comprehension.

Cite journal dates

Information visualization is a growing area of research. Presently there is a good amount of discussion surrounding new and novel visualization

techniques and how to create effective visualizations . The articles presented in these journals typically focus on the creation of specific visualizations, new and interesting ways to graphically represent specific types of data. There is also a suitable body of literature concerning the process of creating a visualization. There have been a number of books published on the subject . These works outline the steps needed to visualize different types of data but do not attempt to automate the process.

The majority of new techniques exists in some degree of isolation. Individual implementations typically exist in near complete isolation, often with the dataset hard coded into the software. There are a number of products that attempt to create a complete end to end process for visualizing data sets. These products are either highly specialized or lack complex visualizations and require user input throughout the process .

Highly specialized applications such as gretl can afford to make numerous assumptions about the input dataset. It is assumed they will be used as part of a specific suite of tools and as such are able to directly process the proprietary output of such tools. Such output is often rich with meta-data which is used to assist the visualization process .

General purpose tools such as Microsoft Excel contain a number of simple charts and graphs. They require a basic level of user training to create and offer no assistance in selecting the most appropriate chart for a given dataset. This often leads to unsatisfactory, confusing or even misleading results .

With a few exceptions, such as gretl, these tools are proprietary and lack documentation on the techniques they use .

This paper aims to address these deficiencies by providing a fully automated end to end visualization tool for arbitrary datasets. There have been some notable projects that accomplish such a goal. Emphasis has been placed on areas where previous works have relied on human actors to complete the process.

1.3 Research Question

The objective of this project is to determine to what extent can suitable visualizations be dynamically and automatically generated using browser based technologies. The Automatic Classifier and Data Visualizer, or AC↯DV is presented. The goals for the project are as follows:

Cite IEEE Visualization Journal, Sage and others

Mazza, Shneiderman, Approche Graphique, Haskell etc

How do you back up a claim like this? Perhaps get rid of it

Ref state of the art or just leave for SOA?

again, state of the art, gretl would be a good example, need more

Excel

Either cover this in SOA or get rid of it

Show a really bad example of excel output

This is another rather bold claim

Polaris, Mackinlay. Need to look back over these and pick out what was expanded upon.

- Design and develop software capable of accepting arbitrary data in a specific format and display it using suitable visualizations. This should be done without an intervention from the user.
- Access the level of benefit that the visualizations can bring to potential users from various fields.
- Investigate to what extent visualizations can be generated given only an input dataset.
- Elaborate on the potential of a more sophisticated version of the software using additional techniques to determine features of input datasets.

Key to this project is the notion of suitable visualizations. In order for a visualization to be considered useful it must meet several criteria. This will be outlined in detail in later sections .

Ref appropriate chapter

Browser based technologies have been selected due to their wide-scale support. The goal of the system is to simplify the process of visualization creation. Browser based technologies require no setup or configuration and thus simplify the process. . The use of browser based technologies also help to offload the vast majority of computation to the client. Were AC⚡DV to be developed further this approach would significantly reduce the cost needed to operate it as a service.

But it's not an end user app so is this sufficient justification?

1.4 Evaluation

The system will be evaluated by inputting a number of datasets and assessing the output against the previously stated goals. A number of use cases have been drawn up to determine how beneficial such a tool is in aiding various users in visualizing data. These use cases entail typical visualization tasks a user from any number of different fields may encounter, i.e. visualizing expenditure and returns for different products.

Switching between tenses a bit here

. AC⚡DV is not intended as an application suitable for direct use by end users. It is the potential core of an easy to use visualization tool. For this reason an evaluation of the user experience is not needed. The goal of the system is to quickly produce effective visualizations that represent the data accurately.

Visualizations are composed of many individual elements with different attributes such as colour, size and spatial location. These elements have been the subject of previous research and a number of guidelines exists outlining their usage . However these guidelines are not hard rules and their exists no formal way to evaluate a complete visualization. Methods from the field of Human Computer Interaction are typically used, specifically user evaluations and trails. AC↯DV generates visualizations that are well understood, thus eliminating the need for lengthy user evaluations concerning the effectiveness of the individual output visualizations. The guidelines stated above are used by the system to pick appropriate visualizations. These guidelines will be used to benchmark the effectiveness of visualizations produced by the system.

Cite Mazza, Jock etc. CALL them retinal variables

1.5 Overview

This chapter is followed by a survey of the State of the Art in the areas of automated visualization and the visualization process. chapter 3 examines the design of the visualization tool and gives an overview of the process of Information Visualization. chapter 4 gives a complete description of the projects implementation and any problems encountered. chapter 5 is an evaluation of the project and a discussion of its merits and failures. The extent to which such a system is viable is also discussed in this chapter. The paper concludes with a look at potential future work that could extend AC↯DV.

Chapter 2

State Of The Art

This chapter presents work currently being undertaken in the two main areas of research discussed in this paper; automated visualization and the visualization process. An analysis of these projects conclude this chapter.

2.1 Visualization Process

An understanding of how to create a visualization is key to this thesis. This process has been the subject of a good amount of discussion and seems to be fairly well understood but there are still discrepancies and differences of opinion. This thesis uses a model based off of Bertin's retinal variables (Bertin & Barbut, 1973). This is a cognitive model that ranks the effectives of various visual attributes, i.e. size, colour, at conveying various types of information, i.e. quantity, frequency etc. This section presents an overview of two works detailing this model. The model will be detailed in full in a later section.

2.1.1 Introduction to Information Visualization

Introduction to Information Visualization (Mazza, 2009) is one of the most recent texts providing a state of the art in the field of information visualization. As well as listing many examples of newer more novel visualization techniques in the later chapters it also contains a substantial amount of information about the process of information visualization. This set of first principles has influenced the thesis presented in this paper quite heavily. The author acknowledges that given an arbitrary set of data there is no way

to determine, automatically or otherwise, what visualization is a suitable representation of that data. There is in fact no agreed criteria to evaluate the effectiveness of a finished visualization.

However there are certain characteristics of data that can be used to help select an appropriate type of visualization. These steps are outlined as follows:

- Define the problem: Is the goal of the visualization to communicate some meaning in the data, allow the user to explore the data or confirm a hypothesis?
- Examine the nature of the data to represent: What type of data is being dealt with; ordinal, quantitative or categorical.
- Dimensionality of the data: The number of attributes of the dataset. In a univariate dataset one attribute varies with respect to another. Bivariate and trivariate extend this with multivariate meaning four or more dimensions.
- Structure of the data: This can be more easily understood by defining what types of data structures can be used to store the data; linear data structures such as vectors and tables, temporal data, spatial data hierarchies and networks.
- Type of interaction will the user have with the data: can the user transform the data by modifying it, can the modify parameters used to display the data or is the visualization static.

This process is summarized in Table 2.1 (Mazza, 2009). It will be revisited in later sections.

Problem	Data type	Dimension	Data Structures	Interaction
Communicative	Quantitative	Univariate	Linear	Static
Explore	Ordinal	Bivariate	Temporal	Transformable
Confirm	Categorical	Trivariate	Spatial	Manipulable
		Multivariate	Hierarchical	
			Network	

Table 2.1: Variables to consider when designing visual representations.

This process eliminates visualizations that are not suitable for a given data set. Deciding on an individual visualization to use is a decidedly harder

task. The effectiveness of a visualization is dependent on the perceptual capabilities of the viewer. As there is no empirically verified model of human perceptual abilities one is purposed. This model is based off the work of Jacques Bertin (Bertin & Barbut, 1973). Bertin was the first person to define a set of "retinal variables"; how different properties of graphics can be used to convey various types of information. His work has been extended since it's original publication and the model presented here is perhaps one of the most up to date. A list of retinal variables is presented in order of effectiveness conveying different types of information. This idea will be revisited throughout this paper and a complete model will be detailed in a later chapter.

Mazza's work is a very complete introduction to information visualization and draws attention to all of the factors that need to be considered when creating a visualization. It is perhaps the most complete state of the art currently available and builds on a large amount of influential publications in the field. It has been very influential in the design of the system presented in this thesis.

2.1.2 The Structure of the Information Visualization Design Space

The Structure of the Information Visualization Design Space provides an overview of the field of data visualization as it was in 1997. An organization of information visualization literature is purposed and several examples are provided. The organization technique is then used as a framework for new designs. The paper builds on Bertins work (Bertin & Barbut, 1973) which has been expanded on by Mackinlay (Mackinlay, 1986). Visualization techniques are grouped based on similarities of their data to visualization mappings.

A series of steps that need to be preformed to create a visualization are given.

Data Data is defined as a set of values taken on my a set of variables. It is an obvious prerequisite of any visualization. Different types of data have their own characteristic operations and subcategories have yet more unique operations. For example patent text or a financial report have their own characteristics and unique operations. In order to do any useful processing

of data a more general model is used. Data is divided into three categories.

- Nominal, denoted by N
- Ordered, denoted by O
- Quantitative, denoted by Q

Data is further divided into the original dataset D and data D' that has been selected from this set and possibly transformed by some filter F.

Visualizations Visualizations are defined as being of composed marks, their graphical properties and elements requiring human controlled processing. Human visual processing works on two levels.

- Automatic Processing: Works on visual properties such as position and colour. These are Bertins retinal variables (Bertin & Barbut, 1973). This processing occurs at a subconscious level and as a result is very fast but limited in power.
- Controlled processing: This can be easily thought of as conscious thought. It is very powerful but limited in capacity. These limitations are fairly well understood (Miller, 1956).

Visual presentations consist of a set of marks (points, lines etc), a position in space (X and Y) and a set of retinal properties (colour and size). This distinction is a somewhat different grouping to the ones purposed by Bertin and Mackinlay. One can only assume the reason for this grouping is to limit the number of attributes or combination of attributes of a visualization. Connectedness and enclosure are also given as attributes.

These attributes are then arranged in a table to categorize various visualizations. An example is shown in Table 2.2. This example has been simplified to two dimensions.

Name	D	F	D'	X	Y	R	$--$ []	CP
Year	Q	$>$	Q	P				
Quality	Q	$>$	Q		P			
Type	N	$>$	N			C		

Table 2.2: Bar chart classifications. R, retinal properties. C, colour. $--$ Connection. [], Enclosure. CP, Control Processing.

Structure of the Information Visualization Design Space presents a good overview of the state of visualization research. The classification method is useful but is continuously amended to account for different types of visualizations.

2.2 Automated Visualization

The focus of this paper is automated visualization, taking a dataset and generating a visualization from it without any user input. There are a few projects that incorporate some kind of automated data visualizer. Generally such a visualizer is part of a larger project and requires user interaction to function. This chapter presents two such projects with varying degrees of automation.

2.2.1 Polaris

Polaris extends the well known pivot table interface to display information visually. citation Multiple visualizations are displayed on a pivot table which the user can interact with by selecting or "brushing" data-points to filter the displayed data. The visualizations act as interactive query builder that allow user to explore large datasets quite rapidly. First published in 2000 it has evolved into Tableau, a commercial visualizations product.

Polaris consists of two main components. A graphic generator and a database query generator. The database query generator will be omitted in this discussion as it is not relevant this thesis. The graphic generator makes several assumptions about the nature of the input data. Similar assumptions are made in this thesis.

- Data is characterized as either quantitative or ordinal.
- Nominal data is assigned an ordering and treated as ordinal.
- Nominal fields are dimensions, or independent variables.
- Quantitative fields are measures, or dependent variables.

Three different types of graphics can be generated based on the input data. User input is then used to refine this to a single generated graphic.

The user interacts with a relatively complex UI. The UI contains a number of "shelves" onto which the user places data-sources and data records.

The user also selects what type of mark to associate with each record. The selected mark and nature of the data is used by the system to determine what type of visualization to render. For example, if the data-source contains an ordinal and quantitative field and the user selects a bar as the mark to use a bar chart is generated. If the user selects a dot as the mark a dot plot is generated. An understanding of Bertins retinal variables (Bertin & Barbut, 1973) is encoded into the system to ensure the visualizations are comprehensible. It is not clear how Polaris handles input that cannot be effectively displayed using these rules, e.g. when the input data set is too large to be displayed.

Polaris allows multiple series of data to be overlaid on the same visualization but it does not appear to allow more than two dimensions of data to be displayed at the same time. As it is a data exploration tool it requires user input too function. While it does contain an understanding of retinal variables these are used to automate the generation of mappings between data and graphical marks. It does not contain any rules governing how the various retinal variables are combined so it may be possible for a user to create ineffective or incomprehensible visualizations .

Go into a bit more and cite

2.2.2 A Presentation Tool

A Presentation Tool, henceforth referred to as APT, is an application-independent presentation tool capable of automatically designing effective visualizations of relational information. It attempts to create a wide variety of designs and encode graphic design criteria in a useful form. It achieves this by defining graphical presentations as sentences in a graphical language. An expressiveness and effectiveness criteria are used to codify the graphic design criteria. The "sentences" produced by these two systems are then combined using artificial intelligence techniques to create designs.

APT is divided into three parts, expressiveness, effectiveness and composition. Expressiveness determines how the input information can be expressed in a graphical language. It is encoded into a formal language. This languages main purpose is to ensure that any graphic encodes all of the input data and only the input data. It is not used to select entire graphics, rather it is used to test individual graphical mappings, i.e. to a single axis or type of mark.

Effectiveness is is used to determine which of the sentences generated by

the expressiveness phase is most effective at displaying the data in a useful manner It is dependent on the capabilities of the user. There is a difficulty in that no verified theory of human perceptual capabilities exists so one is purposed. It is based off Bertins retinal variables (Bertin & Barbut, 1973) which have been ranked in order of effectiveness by Cleveland and McGill (Cleveland & McGill, 1984). This ranking has been extended to include rankings of ordinal and nominal types of data although this extension has not been empirically verified .

Double check expressiveness and effectiveness synchronous w.r.t each other.

Does Mazza match up with this ranking?

The system generates multi-dimensional graphics by combining the effectively one dimensional or single data series sentences generated in the expressiveness and effectiveness stages. Designs are merged on common data. The types of data to merge on are ordered in terms of effectiveness.

- Mark composition.
- Double axis composition.
- Single axis composition.

Replace the system with relevant name

APT combines these three stages using artificial intelligence techniques. The algorithm has three steps: partition, selection and composition. Each step contains various choices. If these choices do not lead to design backtracking is used to consider others.

- Partitioning: Each set of relations, or columns in the database, are partitioned to match the criteria of of one of the sentences defined in the expressiveness stage.
- Selection: A list of candidate designs for each partition is ordered in terms of effectiveness.
- Composition: The individual designs are composed into unified presentations of all the input data using the composition algebra.

Example in notes of bar-chart rule.

Overall APT is a very robust and complex system. However it is perhaps somewhat dated. It is only capable of generating static graphics. It can also factor in the output media into the design choices, i.e. if the screen is monochrome colour is omitted for the effectiveness stage. The use of A.I. techniques is a novel and somewhat unique approach. The parameters of

APT are not made completely clear, it is not clear what the limits of its capabilities are.

2.3 Data Visualization Tools

Thus far a reasonably complete examination of various visualization systems and processes have been discussed. However in order to get a more complete sense of the state of the art of automated visualization a broader approach is needed. This section presents brief summaries of a number of visualization design tools. These tools are then used to create a feature matrix presented in Table 2.3.

Blurb here about general groupings of tools

Gretl Gretl is a software package for econometric analysis. It provides a wide variety of financial calculations and ,notably , visualizations. It is a highly specialized application and generally expects data formatted in a certain way. The user must pick the type of chart to render. No further user input is needed. Admittedly very few intermediary processing steps are needed to generate a specific visualization given a pre-defined data format. However Gretl can handle some variation in this format so the process is not a straight mapping. The types of visualizations generated include line graphs, box blots, scatter plots and other graphics typical of econometric analysis. Gretl automatically ranges the dataset so the output visualizations are always comprehensible.

Citation

Microsoft Excel Microsoft Excel is a substantial data analysis tool that centers around a pivot table display . Users can visualize the data they are working with by selecting the relevant table entries and selecting a chart type. Excel then automatically generates the visualization. Some user training is required in order to produce effective and correct visualizations. It is very easy for a novice user to create nonsensical visualizations, or in the worst case misleading ones. An new version of the software attempts to remedy this by providing functionality to automatically recommend the most effective visualization. A number of visualizations are available but are generally limited to no more than three dimensions. Any exceptions expect data of a specific format, i.e. Candle stick graphs expect specific labels.

Citation for excel and pivot table

Google Charts API Google Charts API is a browser based charts library . It is an API aimed at developers and is not suitable for end users. User friendly interfaces do exist but it is debatable how easy they are to use without training. The API provides a wide number of visualizations. These visualizations also contain elements of interactivity, leveraging the features of modern browsers to do so. Although it is an API the software is not completely manual. Given an input dataset it is capable of automatically generating visual mappings. However it should again be noted that this process is greatly simplified by having the user select the output visualization to generate.

citation

Cite Hohi Charts

Tableau Tableau was born out of the research done by Mackinlay(Mackinlay, 1986). Like APT it appears to be a very robust and complex system. Interestingly it makes use of the visual shelving system presented by Polaris . The user inputs a data source and then select the attributes to visualize. Again, as in Polaris, the data is divided into dimensions and measures. As the user selects attributes to visualize various types of applicable charts are made available. It is not made clear which one of the available visualizations is most effective so the user is left to decide. As it is a progression of the work presented by Mackinlay it is assumed an encoding retinal variables is present.

citation

ManyEyes Many eyes is a web based collaborative visualization authoring tool. Multiple users can work to create a visualization. It offers a wide variety of visualizations but leaves the task of selection completely to the user. If the system cannot generate the selected visualization it notifies the user. Aside from this the user can generate any visualization they see fit. Without an understanding of the various visualizations a user can easily generate ineffective graphics. While the focus of this tool is collaborative authoring it does contain features that automate various aspects of the visualization process. Data can be input in a variety of of formats and the system appears to do a good job of mapping them to appropriate visual features. The visualization engine is Java based

Expand or is that enough for anyone to know?

OpenHeatMap OpenHeatMap is a web based geographic visualization tool. It allows users to overlay data across various maps. It is very easy

to use. A dataset is uploaded and the visualization is presented. It is not clear what the formatting restrictions of the input data is but it appears to be relatively robust if occasionally fickle. The software is flash based and struggles noticeably with even moderately sized datasets. The user can zoom and pan the visualization interactively.

D3 D3 is a browser based charts library. It offers much more control and flexibility then Google Charts API and as a result requires a substantially greater amount of developer effort. Whereas Google Charts offers complete visualizations D3 sits at a much lower level of abstraction. It is a tool set developers can use to create their own visualizations from scratch.

Published in 1986, perhaps too old?

2.4 Analysis

Presently there is a good amount of literature surrounding individual visualization as demonstrated by (Mazza, 2009), and large degree of consensus on how to go about creating a visualization. However there have only been a few attempts to automate such a system. Such an application would not only allow users to quickly produce useful and accurate visualizations but would also server to validate the process that has become agreed upon. Such an evaluation is frequently omitted from these publications. Mazza outlines a set of criteria to evaluate individual visualizations (Mazza, 2009) but the focus of this thesis is testing if the entire process is valid.

Cite Structure of information

Works such as Polaris and APT suggest that this process is indeed valid. However Polaris does rely on the user for some tasks. APT appears to be a very robust and complex system but is somewhat dated and cannot generate visualizations with any interactivity. Given the ubiquity of powerful computing platforms this seams to be quite a large deficiency. The project presented in this thesis aims to create an automated version of the visualization process. The interesting aspects and possible deficiencies of the various systems examined in this chapter are summarized in the table below. The terms used to evaluate each system are qualified below the table.

citation

citation

Native Browser Based. The systems runs in a web-browser environment using native web technologies. These technologies, i.e. HTML5, SVG, EM-

	Native Browser Based	Automatic Visualization	No User Training Required	Generic Application	Large Number of Visualizations	Interactive	Automatic Feature Determination	No User Input Required	Encoding of Retinal Variables
Gretl		x	x			x			
Excel				x	x		x		
Google Charts API	x			x	x	x	x		
Tableau		x	x	x	x	x	x		x
ManyEyes		x	x	x	x	x	x		
OpenHea Map		x	x			x		x	
D3	x			x	x	x			
Polaris		x	x	x	x	x	x		x
APT		x	x	x	x		x		x

Table 2.3: Summary of systems examined

CAScript etc, arguably compose the most widely adopted software platform in history. These technologies are described via specifications and implemented by multiple vendors. This approach has several benefits compared to older container based web-technologies such as Java or Flash.

Citations everywhere

Go into this more back in the intro and ref from here

Automatic Visualization. Given a dataset the system should be capable of automatically generating visualizations with no user input during the visualization process. Tasks such as selecting which entries to visualize or what visualization to output are not included as user tasks in this context, only the tasks involved in mapping data entries to graphical marks.

No User Training Required. This is somewhat hard to quantify meaningfully as it depends on the capabilities of each user. Low level technologies such as APIs are not intended for end users and require training to use. Complex applications with many features may also require user training. Any system that allows the user to generate incomprehensible visualization requires training to be used successfully.

Generic Application. A wide array of datasets can be input to generic systems. Such systems do not make assumptions about the structure of the input data, i.e. Each record must contain a data, certain number of variables etc.

Large Number Of Visualizations. The systems can output a wide array of visualizations. Of particular importance is the systems ability to output different types of visualization, i.e. Two dimensional, three dimensional etc.

Interactive. The system should provide some sort of interaction. This can be as simple as scroll over text or as complex as multiple nested visualizations than can be panned and enlarged.

Automatic Feature Determination. The system should be able to automatically determine the features of the dataset. These include attributes such as dimensionality and types of data. Systems with a limited number of uses typically have this information hard coded in based on data labels and do not determine it at runtime.

This is weak, maybe take it out

No User Input Required. At no point should the user have input information into the system. This extends automatic visualization requirement by including such actions as selecting an output visualization.

Encoding of Retinal Variables. The work started by Bertin (Bertin & Barbut, 1973) has been incredibly influential. Most visualization systems and visualizations incorporate it to some degree. Furthermore Bertin's original work has evolved substantially and can be found under different names in many publications outlining the visualization process. For our purposes it is assumed that some knowledge of retinal variables are encoded in a system if there is some mention of them or if it is not possible for the user to generate visualizations that clearly violate the rules outlined by Bertin and others (Card, Mackinlay, & Shneiderman, 1999), (Mazza, 2009), (Spence, 2001), (Cleveland & McGill, 1984).

This thesis will attempt to determine to what extent can suitable visualization be dynamically and automatically be generated from arbitrary data using browser based technologies. At no point should the user have to provide input. Browser based technologies have in recent years become very powerful platforms and are perhaps the most ubiquitous software platforms in existence.

2.5 Conclusion

Replace section with chapter in the text

Chapter 2 introduced a state of the art automated and manual visualization processes and how they can be used to simplify the creation of visualizations from arbitrary data. By making the process of creating a visualization easier more data can visualized and thus comprehended more readily. Many existing systems are limited by requiring human intervention. If the user is untrained they can introduce errors which may lead to confusing or even mis-representative visualizations.

Chapter 3

Design

This chapter introduces the design principles and core technologies used in this project. The architecture for the Automatic Classifier and Data Visualizer, henceforth referred to as AC↯DV, is presented. The visualization process is also described. The chapter concludes with .

Take ACDV out of intro?

SOMETHING

3.1 Technologies used

This section describes the technologies used in this project and the motivation for choosing said technologies. One of the main aims of the project was to quickly generate visualizations. This extends to any system setup the user may have to perform before inputting data. Extendability was another key consideration. The system should allow new visualizations to be added with relative ease.

Browser based technologies have been selected for use in this project. The core technologies used are SVG, Javascript and HTML. Modern web-browsers allow for near seamless interoperability between these technologies. Such technologies offer many advantages over more traditional standalone executable binaries. If new visualizations need to be added they need only be appended to the existing software. They are then instantly available. The web browser if perhaps the most ubiquitous software platform in history making cross compatibility for applications within it almost a non-issue. The technologies used are freely available and exists as standards, not just implementations. This can lead to unexpected behavior in separate implementations but this is becoming less common as these standards and their

implementations mature. As these technologies are not maintained by a single vendor they are far more resilient to an organization discontinuing their support. A short description of all the technologies used follows:

Javascript is a loosely typed interpreted programming language. It is defined in ECMA-262 standard. It was originally implemented as a client side scripting language. Javascript functions can be embedded or included in HTML pages and they can interact with the DOM. It is a dynamic language, a feature that was utilized extensively.

Document Object Model (DOM) is a convention used to represent and interact with objects in HTML and XML. The objects are stored in a tree structure that can be addressed and manipulated. This project creates graphics by generating and placing multiple SVG objects into the DOM using Javascript.

Scalable Vector Graphics is an XML based vector image standard created by the World Wide Web Consortium. SVG images support interaction and animation. The images and their behavior are are defined in XML which allows them to be manipulated and generated like any DOM object. The ability to generate interactive graphics at runtime makes them ideal for this project.

HyperText Markup Language (HTML) is a markup language that allows the creation of web pages that can embed objects such as SVG images. AC⚡DV uses HTML as a simple container to link together its various components. Google Chrome was used to render the HTML pages.

D3 is an open source Javascript library for manipulating DOM based on Data. It allows data to be bound to a DOM and then apply data-driven transformations to the document. A simple example would be generating a HTML table from an array of numbers. It greatly simplifies the modification of DOM compared to the W3C DOM API. The key difference is D3 uses a declarative approach instead of the W3C imperative approach. D3 was used extensively in the creation of AC⚡DV.

Cascading Style Sheets (CSS) is a style sheet language that expresses the presentation of structured documents. The key benefit offered by CSS is the separation of the presentation of DOM element. It is not used extensively in AC↯DV.

Web Server. Although AC↯DV runs entirely on the clients computer almost every web browser does not support cross-origin resource sharing from local files. Such requests must be sent using HTTP. This means that data stored in an external file can not be loaded into AC↯DV without the use of a web server. A very simple python server was used in this project but any HTTP server is sufficient.

One area of concern was performance. Standalone executables have greater access to a machines resources and can take advantages of optimization techniques such as multi-threading or specialized instruction sets such as SSE . This is not usually a problem for browser-based applications but as AC↯DV may have to process thousands of records some effort has been made to evaluate the its performance and keep it within a reasonable bounds.

Need citation?

Citation

3.2 Overview of Data Visualization Process

A variety of different techniques have been purposed to crate date visualizations. The technique purposed by Mazza (Mazza, 2009) is one of the few constructed from a sequence of complete steps. These steps do not vary based on the nature of the data or the desired output making the process one of the most suitable for automation. The process was outlined in Table 2.1 and will now be expanded on.

1. **Define the problem.** This step depends on the goal the user is trying to accomplish. It is also necessary to make note of the perceptual capabilities of the user as these can be used to inform decisions later in the process. There are three main problems data visualization can help to solve.

    - Communicate. Visualizations can be used to quickly communicate information present in a data set. This is the most general

task and can be thought of as presenting the entire dataset for easy consumption by the user.

- Explore. Visualizations can be used to explore datasets. Exploratory analysis is often used to form new hypothesis that may not have been considered with more formal techniques . Such visualizations summarize the main characteristics of the data set and then allow the user to filter the data and focus on subsets.
- Confirm. Here the data is being analyzed to confirm a specific hypothesis.

Cite Tucky

2. **Nature of the data.** Here the data is classified into one of three types. This classification is later used to inform the visual mapping.
   - Quantitative or numeric data, e.g. real numbers.
   - Ordinal or non-numeric data which does have an intrinsic ordering, e.g. days of the week.
   - Categorical or non-numeric data with no ordering, e.g. names.
3. **Dimensionality.** The number of dimensions, or attributes, of a dataset determine what representation to use. These attributes can be dependent or independent. The dependent attributes are generally the ones of interest and are analyzed with respect to the independent attributes. The number of dependent attributes determine the dimensionality.
   - Univariate. One dependent attribute varies with respect to an independent variable. Example visualizations include Bar and pie charts.
   - Bivariate. Two dependent attributes vary with respect to an independent variable. Example visualizations include Histograms.
   - Trivariate. Three dependent attributes vary with respect to an independent variable. Example visualizations include Bubble charts and Tree Maps.
   - Multivariate. Four or more dependent attributes vary with respect to an independent variable. This is where visualization becomes much more difficult because people are not used to working

in four or more dimensions. Example visualization include Parallel Coordinates and Scatter-plot Matrices.

citation

Citation

Generally multiple instances of the same type of visualization can be combined to create a visualization of greater dimensionality, i.e. Clustered bar charts can be used to show data with many dimensions. There are limits on the combination of visualizations. These limitations vary depending on the specific type of visualization. These limitations will be addressed in Section 3.3.

4. **Data structures.** These are used to contain the data and are derived from the nature of said data.

   - Linear.
   - Temporal.
   - Spatial.
   - Hierarchical.
   - Network.

   This is somewhat difficult to automate completely and generally involves pattern matching the data against known templates. For example there are a myriad of ways to represent a valid date.

5. **Type of interaction.** This depends largely on the initial needs of the user but is also decided by factors such as the scale of the dataset (large set may necessitate interaction as they cannot be displayed completely) and the medium used. A visualization can be:

   - Static. Not modifiable.
   - Transformable. The user can modify and transform the data by filtering entries, choosing different visual mappings etc.
   - Manipulable. The user manipulate the view by zooming or rotating the image.

Each of the options described here help to point a specific visualization technique. The architecture of AC↯DV is a simplified version.

3.3 Visual Mapping

Visual Mapping is the most important aspect of creating a visualization. If the elements of a dataset are mapped to visual elements effectively any patterns in that dataset are easily perceivable. A poor mapping results in these patterns becoming almost imperceptible. Creating an effective mapping is done by understanding and exploiting the way humans perceive what they see. Certain visual properties will more readily draw greater focus. There is a model detailing the effectiveness of various visual properties although it is not concrete. This model of human perception began with the work of Bertin (Bertin & Barbut, 1973) and has been expanded upon by many others .

Lots of citations here

3.3.1 Memory

Human perception is composed of human vision and human memory. What we see is stored in memory and then processed. There are three basic types of memory:

- **Sensory Memory.** All input from sense organs is stored in memory for a very short time. Such memory is independent of conscious control and is processed automatically. For this reason the processing that takes place is called *preattentive processing*. During preattentive processing only a limited set of basic features are considered. These are the *retinal variables* and includ attributes such as colour, size and position. An understanding of these attributes and how they can be used to convey different types of information is absolutely key to creating an effective visualization. Pre-attentive processing is extremely fast and allows a detailed mental picture of a scene to be built very quickly.

- **Short-term memory.** Short term memory is where some of the information from sensory memory is transfered. Information can persists here for up to a minute but the capacity is extremely limited compared to sensory memory. The general rule is that 5 to 9 items can be stored in short term memory although this depends on the type of information being stored (Miller, 1956).

- **Long-term memory.** Items in short-term memory can be committed to long term memory by repeated rehearsal.

3.3.2 Retinal Variables

A number of preventative visual properties have been identified. These visual properties have been ranked in order of effectiveness at conveying different types of information although such rankings do not seem to have been completely verified (Mackinlay, 1986) (Mazza, 2009).

Ths visual properties can be grouped into four categories: *colour, form, movement and position.*

Colours can be expressed in different ways. Red, Green and Blue values are typically used when expressing the colours on a computer screen. For the purpose of data visualization it is more efficient to talk about colour in terms Hue, Saturation and Lightness as they map to preventative attributes far more readily. Hue is the aspect of colour we typically refer to by name and is usefully for labeling different sets. Saturation and Lightness describe the intensity of the Hue. A colour with a high intensity is more distinguishable from its surroundings so intensity is used to give certain elements a kind of visual precedence.

Motion is the most effective way for an attribute to gain focus. It should be used sparingly else the visualization quickly descends into a distracting mass of flickering objects.

Position is the most accurate attribute for encoding quantitative data in graphics.

Form is the most comprehensive of the visual attributes. There are many attributes that make up the form of an object, all of which are capable of representing various types of data with various levels of effectiveness.

- Orientation
- Length
- Width

- Size
- Collinearity
- Curvature
- Spatial Grouping
- Added Marks
- Shape
- Numerosity

There are a number of different rules governing the use of these attributes. These rules are primarily concerned with the number of attributes to use and which attributes are the most suitable to use for different types of data.

The number of different pre-attentive attributes to use in a visualization is quite important. If it is too great an incomprehensible visualization is produced. If the number is too low then the full capabilities of the medium have not been exploited. The latter is the more desirable so a conservative approach is best adopted.. Millers work on human information processing capacity (Miller, 1956) is the most widely known put does not deal with specifically with preventative loading. Similar studies specifically with visualization in mind have also been conducted (Few, 2004), (Ware, 2012). These studies have not come to the same conclusions so the limits they purpose are not hard rules. Ware suggests no more than ten distinct values for pre-attentive attributes, although there are several exceptions, while Few limits this number to four.

The combination of attributes is a far harder problem and has not been the subject of much research. However different attributes are better than others at conveying different types of information. This has been the subject of a relatively large amount of work. Cleveland and McGill purposed and verified a ranking of of attributes in terms of their ability to convey quantitative information. Mackinlay expanded, although he did not verify, this ranking to include ordinal and categorical information (Mackinlay, 1986). More recent research has shown that the issue is more complicated (Spence, 2001) so a definitive raking does not exists. Mazza extends the work done by

Few (Few, 2004) to purpose an "indicative rule of thumb" for the three data types (Mazza, 2009). Mazzas work is a simplified ranking the groups the pre-attentive attributes into one of three groups for each data type: suitable, limited suitability and not suitable. It is shown in Table 3.1.

Attribute	Quantitative	Ordinal	Categorical
Hue	×	×	✓
Intensity	–	✓	×
Orientation	–	–	×
Length	✓	–	×
Width	–	–	×
Size	–	–	×
Collinearity	×	×	×
Curvature	–	–	×
Spatial Grouping	×	×	×
Added Marks	×	×	✓
Shape	×	×	✓
Numerosity	✓	✓	×
Position	✓	✓	–
Flicker	×	×	–
Motion	–	–	×

Table 3.1: Encoding the three data types with pre-attentive attributes (Mazza, 2009). ✓ indicates the attribute is suitable for the given data type. – indicates the attribute has limited suitability. × indicates the attribute is not suitable.

Short term memory and pre-attentive processing play en extremely important role in the design of effective visualizations. An understanding of these pre-attentive attributes can be used to make the most important information draw the focus of the viewer. If the rules surrounding these are ignored it can result in incomprehensible visualizations.

3.4 Architecture

AC↯DV consists of two main components. A chart API and a classifier. Data is input to the classifier where an appropriate charts is determined and displayed in the view. The two components are distinct. The chart API can be used completely independently. Similar chart APIs do exist but are relatively complex. The goal of AC↯DV was to create a chart API that used an minimum of parameters.

reference DC.js

Diagrams

The system has been designed in this way to reflect the visualization process described above. The classifier encodes a paired down version of the process outlined by Mazza detailed in section 3.2. This approach also has the benefit of yielding a chart API that can be used by developers who wish to easily create a specific type of visualization.

3.4.1 Classifier

As mentioned the classifier based on the process outlined by Mazza. It is summarized in Table 3.2.

Problem	Data type	Dimension	Data Structures	Interaction
Communicative	Quantitative	Univariate	Linear	Static
Explore	Ordinal	Bivariate	Temporal	Transformable
Confirm	Categorical	Trivariate	Spatial	Manipulable
		Multivariate	Hierarchical	
			Network	

Table 3.2: Variables to consider when designing visual representations. Grayed out entries have been omitted from AC↯DV.

The first and last stages of the pipeline are fixed as they depend on the needs of the user. The most generic choices have instead been selected. A communicative visualization gives an overview of the dataset while a manipulable view allows the system to create an interactive visualization. AC↯DV is not a data processing tool so any transformations on the dataset must be done using other tools. The middle three stages of the pipeline have been the focus of the project.

AC↯DV reduces the categorization of data to ordinal and quantitative by assigning an implicit ordering to all nominal fields and subsequently treating them as ordinal. The assigned ordering is not important, indeed with many visualizations preserving the ordering of ordinal variables is not important for the final output. All nominal fields are treated as independent variables and quantitative fields treated as dependent variables. These are the same assumptions made by Polaris .

Citation

The current implementation of AC↯DV is limited to producing three visualizations with three or less dimensions. There is no technical reason for this limitation. The classification engine is designed to be extended easily.

AC↯DV uses a single data structure for all processing. The system can

test if a quantitative variable represents a date, hence the inclusion of temporal data structures. The design could be extended to detect the remaining types of data as such a task can be accomplished, to a high degree of reliability, with pattern matching, but this was out of scope.

The classifier determines these attributes at run time from the dataset. Each chart in the API has a similar set of properties. The system then attempts to match the characteristics of the dataset to the characteristics of a chart. AC↯DV returns a confidence score along with the visualization to give an indication of how suitable the visualization is. This score is determined from how closely the two sets of characteristics match. Some characteristics must match for the visualization to be considered such as the dimensionality and types of information to be displayed, i.e. if no chart can display two independent variables then no chart will be returned.

The encoding of the charts properties is a manual step that must be done by an expert. It is here that the retinal variables are considered. The mapping from data to marks is encoded into each visualization and it is up to the expert to decided how to bound them. As mentioned there are no hard rules to go by but a conservative approach has been adopted.

The system also contains a notion of "soft" and "hard" size for each visualization. These are the limits on how many records may be displayed in each visualization. Like the mapping to graphical marks and their associated retinal variables this notion of size is fixed. Deciding on the maximum number of entries for each visualization is quite tricky. It depends on

- The pre-attentive attributes used in the visualization.
- The number of items the user can commit to short term memory, which again is somewhat dependent on the pre-attentive attributes used (Miller, 1956).
- The task of the user. If they are investigating specific entries they will have to memorize them while the compare them to other entries. If they want an overview of an patterns in the data retaining specific entries is less important.

This is a very hard problem to automate and was ultimately abandoned. The last item is attempted with the notion of "soft" and "hard" size. If the dataset is within the soft size limit then every mark will be unique. If

it is within the “hard” limit AC↯DV will modify the output visualization. An example is the bar chart. Many unique entries quickly become hard to distinguish but if there are sufficient entries (between the two size limits) the system considers them part of a single series. The colour for every bar is then set to the same value and the visualization resembles a time series plot. However this may not always produce the desired result so exceeding the soft size limit is reflected in the confidence score.

3.4.2 API

The API consists of a collection of visualizations with shared declaration attributes. The design is based of off Mike Bostock’s, the creator of D3, convention for reusable charts . All visualization are designed as Javascript closures. Each chart only expects input data and a mapping of each data dimension to visualization “axis” (the notion of axis can be unique to each visualization) , which the classification engine can provide automatically. Attributes such as size and position can be adjusted after declaration if desired. Charts could be extended to included customizable colours and marks like similar libraries . However such customization would allow interference with the decisions made in encoding the various retinal attributes.

Citation

Cite Dc.js again

This will probably be broken down

Chapter 4

Implementation

Chapter 5

Evaluation and Discussion

Chapter 6

Future Work and Conclusions

References

Bertin, J., & Barbut, M. (1973). Sémiologie graphique. Mouton; Paris: Gauthier-Villars.

Card, S., Mackinlay, J., & Shneiderman, B. (1999). Readings in information visualization: using vision to think. Morgan Kaufmann.

Cleveland, W., & McGill, R. (1984). Graphical perception: Theory, experimentation, and application to the development of graphical methods. Journal of the American Statistical Association, 79(387), 531–554.

Few, S. (2004). Show me the numbers: Designing tables and graphs to enlighten.

Mackinlay, J. (1986). Automating the design of graphical presentations of relational information. ACM Transactions on Graphics (TOG), 5(2), 110–141.

Mazza, R. (2009). Introduction to information visualization. Springer.

Miller, G. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. The Psychological Review, 63, 81-97.

Oxford English Dictionary Online, 2nd edition. (2013, July). `http://www.oed.com/`.

Spence, R. (2001). Information visualization. Addison-Wesley.

Ware, C. (2012). Information visualization: perception for design. Morgan Kaufmann Pub.

Appendix A

Appendix A