Outlier Visualization in N-Dimensional Categorical Data Sets

A report submitted in partial fulfillment of the requirements for the award of the degree of

B.Sc (hons)

in

Computers (Software Systems)

By

John Rogers (X12105872)

Declaration Cover Sheet for Project Submission

Name: John Rogers
Student ID: X12105872
Supervisor: Simon Caton

SECTION 2 Confirmation of Authorship

The acceptance of your work is subject to your signature on the following declaration:

I confirm that I have read the College statement on plagiarism (summarised overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature: _____

Date: _____

Abstract

Humans can more easily and quickly interpret visual images than they can interpret the same data in text form. Knowledge contained in Big Data sets would be nearly inaccessible to the casual, or even moderately interested viewer, if it was not visualized. The primary use case of this project is server logs provided by IBM from their SameTime test systems.

Providing a meaningful visualization for a high-dimensional categorical case such as the primary use case, is particularly challenging for outlier detection. This is because, in high dimensionality, the data becomes sparse, and all pairs of data points become almost equidistant from one another. By using open source R and cutting edge technologies such as Tableau and IBM Watson, this project addresses the challenge of displaying high dimensional data in a meaningful and informative format.

Keywords: Visualization; Outlier; Sparse Data; Categorical Data;

Contents

Abstract 2

Introduction 6

Motivation for this project *6*

Domain challenges *7*

Background & Literature Review 9

Preliminaries *9*

Outliers *10*

Evaluation of Outliers *11*

High Dimensionality *11*

Visualization *13*

Technologies *15*

Alternative Technologies explored *16*

Methodology 18

Initial planning *18*

Requirements Gathering *18*

Context of use *19*

Scenario of use *19*

Data requirements *19*

User requirements *19*

Environmental requirements *19*

Requirements for effectiveness, efficiency and satisfaction *19*

Analysis & Design *20*

Use Case *20*

Logical Architectural View *21*

Implementation 22

Set up environment *23*

Cleaning the Data *24*

Building a Data Table *25*

Convert to Term Document Matrix *26*

Thresholds & Associations *27*
Convert to term-term Adjacency matrix *27*
Convert to Graph *28*
Formatting the graph *28*
Shiny (work in progress as of 08/06/16) *29*
Evaluation and Testing 31
Client evaluation and Feedback *32*
Testing Methodologies *34*
White Box Testing *36*
Black Box Testing *39*
Usability testing *43*
Conclusion and Further Work 45
Appendices 46
i. Research Interview with Mick Cooney & Jamie O'Leary *46*
ii. Curse of dimensionality code *52*
iii. Requirements Elicitation Interview *53*
Bibliography 56

TABLE OF FIGURES

Figure 1 Examples of Node and Edge Outliers (Aggarwal, 2013) 10

Figure 2 Adjacency matrix 11

Figure 3 Curse of Dimensionality R code (available in appendix ii) 12

Figure 4 Curse of Dimensionality output 13

Figure 5 Galton's smoothed correlation diagram for the data on heights of parents and children, showing one ellipse of equal frequency circa 1886. (Friendly, 2006) 14

Figure 6 Tableau, a modern frequency distribution plot 14

Figure 7 Categorization based on transformation steps within the information visualization pipeline (Liu, et al., 2015, p. 3) 15

Figure 8 Watson Output 1 16

Figure 9 Watson Clustering on cleaned data. 17

Figure 10 Iterative Incremental model of software development 18

Figure 11 Overall Use Case Diagram 20

Figure 12 Logical Architecture View 21

Figure 13 About R Studio 22

Figure 14 IBM Connections Sametime log (test file) 22

Figure 15 Setting up R environment 23

Figure 16 conversion to Term Document Matrix 26

Figure 17 Shiny UI script with inputs 29

Figure 18 Shiny Server Code 30

Figure 19 Raw Data 31

Figure 20 Successful visualization 31

Introduction

Outliers are any data points which show significant statistical or behavioural differences from the rest of the data. (Hawkins, 1980) formally defined an outlier as follows:

"An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism."

(Aggarwal, 2013, p. 1), further explains that when the generating processes behave in unusual ways it can result in outliers which can contain useful information. The outlier itself may be the event that caused the server crash or flagged the intrusion event.

"*Logs are event streams that are constantly spewed from every application, server instance, mobile and IOT device. They contain valuable information pertaining to application errors, system performance, security, feature usage and more*." (Parsons, 2015)

In a log file, the outlier events can be hidden in a vast amount of data and the idea of simply reading the file to find the issues becomes impracticable. It is at this point in time that informative visualization techniques have become of high importance.

There has been increasing interest in having organizations use visualization techniques to analyze and capture the complexity of many domains. This interest has arisen for many reasons: increased appreciation of the inherent complexity of many Big Data challenges; analysts and managers growing awareness of alternative ways to convey and consume information; and enterprise level software companies such as Logentries.com entering the market with off the shelf and/or bespoke software offerings which provide easy insight.

Motivation for this project

Motivation for this project comes from Log Files, which are a source of abundant data. The task of visualising this data into a form from which outliers can easily be observed is a challenging one, which is far from solved.

Automated analysis of log files is a growing area and one of the foremost companies operating in this field is logentries.com. Logentries was founded in 2010 in Dublin, Ireland by Viliam Holub and Trevor Parsons. Holub and Parsons worked together for over ten years at University College Dublin's Performance Engineering Laboratory, where they partnered with IBM to focus their research on root cause analysis of common performance and IT issues.

IBM assisted during this project acting as both client and mentor during the entire process. They also provided me with test logs on which to experiment and provided invaluable feedback during the requirements gathering and evaluation processes.

The commercial motivation for this project can be summed up as follows - *"The use case we are targeting is: Can we make log analysis easier for the 24x7 ops teams? It is in our interests for help the ops teams in order to reduce the amount of calls made to technical support and dev teams. Let's make the logs smaller by producing a filtered view through John's system. Reduce the size of the haystack rather than try discover the needle.".* (O'Connor, 2015)

From the requirements gathering document: "*The customer requires that a server log file can be quickly analysed for anomalous behaviour. The application should take a log file, parse it into usable information and from that information return a visualization of all events in the file*".

The personal motivation for this specific project came about during 3rd year. My work placement was with IBM Ireland working on test automation for their Sametime product. While there I asked some of the managers and engineers for their thoughts on final year projects. After fielding a few different ideas, I decided that working on automating log file analysis would be of interest to me and would be applicable specifically to the Applied Artificial Intelligence module of semester 8 and applicable tangentially to other modules this year.

Domain challenges

The domain challenges of this project have been discussed in Appendix i (Cooney & O'Leary, 2016) in more detail and are as follows:

1. The bulk of work on outlier detection and clustering that currently exists is numerical in data. My task was to find a means to express a distance between categorical data sets. For example, in the sequence of values (1a, 3b, 7c, 9e, 11f, 45e) it is relatively simple to see that 45e is a considerably further on one axis than the other data points. This is our outlier on a simple x-y plot.
   However, in the sequence of messages ("User x logged in", "User y failed to log in", "server message m", … "server message n", etc.) how do we define meaningful distances?

2. Data for projects such as this will tend to arrive in unspecified formats, the bulk of time will be spent cleaning and formatting the data into some manageable format.

3. The data must be reduced in size or else it is likely to be unworkable even with modern computers. Was there enough existing knowledge of the data to be able to filter out the bulk of the noise in an efficient manner?

4. Defining the outlier is non-trivial. The message may never have been seen before and as such evaluation of clusters becomes more about intuition than actual maths. Finding the needle in the haystack is great but what if the haystack also contained a pile of gold and diamonds?

5. The project must provide a meaningful visualization that can be easily understood by a lay person.

6. Dimensionality reduction is a huge and highly complicated topic that is still undergoing a great deal of research. Successful dimensional reduction will be of vital importance to the success of the project.

Background & Literature Review

"Doing statistics is like doing crosswords except that one cannot know for sure whether one has found the solution" – John W. Tukey

In 1977, John W. Tukey published *Exploratory Data Analysis* and argued that more emphasis needed to be placed on use of data to test and confirm hypotheses. Also that year the International Association for Statistical Computing was established with a mission statement as follows:" *It is the mission of the IASC to link traditional statistical methodology, modern computer technology, and the knowledge of domain experts in order to convert data into information and knowledge*." (Press, 2013) Since these initial milestones in the field of Data Analytics the concepts of outliers and visualization have undergone extensive study.

Preliminaries

Formal definitions below are based on the definitions of Databases from (Akoglu, et al., 2012)

In this paper we consider server Logs. A Log L is a bag of n terms over a set of m categorical features $F = \{f1..., fm\}$. Each feature $f \in F$ has a domain dom (f) of possible values {v1, v2...}. The number of values v ∈ dom(f) is the arity of f, i.e. arity (f) = |dom (f)| ∈ $\mathbb{N}$.

The domains are distinct between features. That is, dom $(fi) \cap$ dom $(f$j) = $\emptyset$, $\forall i \neq j$. The domain of a feature set $S \subseteq F$ is the Cartesian product of the domains of the individual features f ∈ S, i.e., dom $(S) = \prod$ f ∈ F dom (f).

A Log L is simply a collection of n terms, where each term t is a vector of length m containing a value for each feature in F. As such, L can also be regarded as a n-by-m matrix, where the possible values in a column i are determined by dom(fi).

Silva & Zhao (Silva & Zhao, 2016) define an adjacency matrix for us as follows: Mathematically, a non-weighted graph $G = (V, E)$ or weighted graph $G = (V, E, W)$ are frequently represented by an adjacency matrix A that is constructed from the vertex and edge sets. Therefore, the adjacency matrix A is defined as follows: The number of vertices $|V|$ serves to establish the dimension of the adjacency matrix which is always $V * V$. The edge set contributes to defining the entry values of the adjacency matrix in the following manner: The (i, j)-th entry of A is denoted as $A_{ij} = a_{ij} = W_{ij,}$ where W_{ij} is the weight of the edge linking i to j.

Outliers

There have been several proposals for the identification of outliers including (Aggarwal & Yu, 2001) (Arning, et al., 1996) (Breunig, et al., 2000) (Chaudhary, et al., 2002) & (Ghoting, et al., 2008) which all typically exploit the ordered domains of the attributes to give valid distance functions between tuples and as such, these methods cannot be easily applied on categorical data. These methods also need a distance metric for finding the k-nns of the data points, which can cause them to suffer from the curse of dimensionality in high dimensions.

(Aggarwal, 2013, p. 23) tells us *"Many data sets in real applications may contain categorical attributes, which take on discrete unordered values. ... Many of the techniques for nearest neighbour and density-based classification can be extended to the case of such attributes, because the concept of proximity can be extended to such cases. The major challenge is to construct a distance function, which remains semantically meaningful for the case of discrete data."*

This project will be pursuing a strategy of transforming the data into a graph and as such the explanation of edge and node outliers provided by (Aggarwal, 2013, p. 28) is of great value.

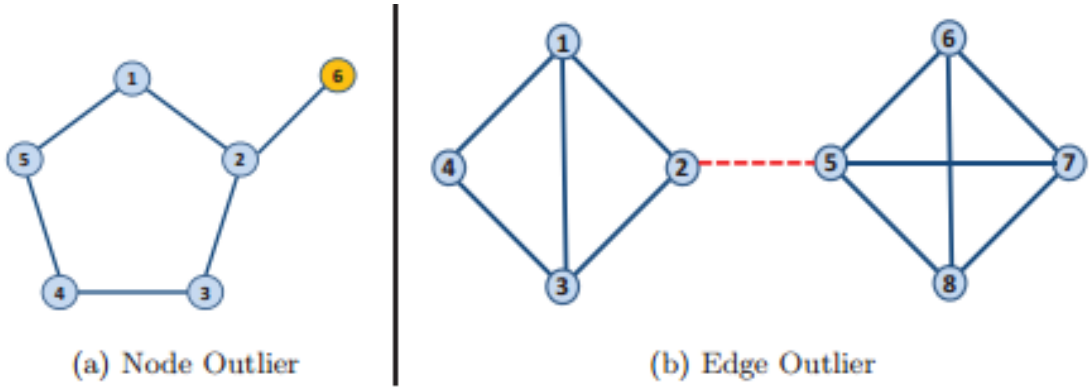


Figure 1Examples of Node and Edge Outliers (Aggarwal, 2013)

In Figure 1(a) node 6 is an outlier as it has an unusual locality structure. Alternately edge 2-5 in (b) could be considered an outlier as it connects two otherwise unconnected communities of nodes. This highlights the idea of "intuition" as per (Cooney & O'Leary, 2016)

(Aggarwal, 2013, p. 200) points out that proximity- and density-based techniques are the most promising methods of performing outlier detection. He makes the observation that "…categorical data can be transformed to binary data, by treating each value of the categorical attribute as a binary attribute.". This is the technique I will be using during implementation.

To further expand on this, it is important to understand the idea of an adjacency matrix.

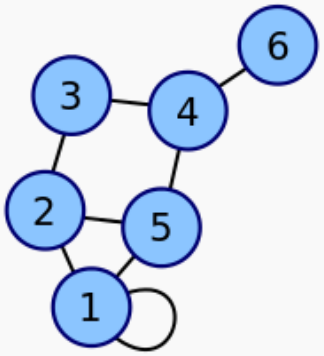
Labeled graph	Adjacency matrix
	$\begin{pmatrix} 2 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$ Coordinates are 1–6.

Figure 2 Adjacency matrix

For a simple graph with vertex set *V*, the adjacency matrix is a square $|V|^2$ matrix *A* such that its element $A_{i,j}$ is one when there is an edge from vertex *i* to vertex *j*, and zero when there is no edge. (Biggs, 1993). Figure 2 illustrates the relationship between graph and matrix.

Evaluation of Outliers

(Boriah, et al., 2008) document several experiments on the evaluation of outliers and they highlight that the overlap measure has become the most commonly used similarity measure for categorical data. They highlight the fact that computing similarity between categorical data sets is not straightforward as there is no notion of ordering. This applies specifically to the main use case.

(Aggarwal, 2013, p. 32) also highlights the issue with evaluation of outliers thus:" *In the unsupervised scenario (without ground-truth), it is often the case, that no realistic quantitative methods can be used in order to judge the effectiveness of the underlying algorithms in a rigorous way*".

High Dimensionality

The "curse of high dimensionality" is well explained by Aggarwal (Aggarwal & Yu, 2001) and by Houle et al. in (Houle, et al., 2010). The curse of dimensionality is the term used to describe the fact that as the number of dimensions in a state space increases, the fraction of volume occupied by the unit sphere becomes smaller and smaller.

As a result, points in phase space tend to become further and further apart when measured by a Euclidean distance metric and this can make the finding of true outliers harder to detect. Furthermore, as the dimensionality of the phase space increases, it often is not matched by the 'true' dimensionality of the data – the data often lies in a lower dimension subspace. Thus, dimensionality reduction techniques such as Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) can often be very useful both in terms of computation and modelling.

Put another way, in real-valued feature spaces, L_p norms or the cosine of the angle between the pair of vectors are commonly used to express similarities between vectors. For L_p norms in high dimensions, the work of Beyer et al. in (Beyer, et al., 1999) ,questions whether the concept of the nearest neighbour is meaningful.

We can use the following snippet of R code to illustrate the issue:

```r
library(pbapply)

Niter <- 1000000
Ndim <- 1:25

vol_frac <- pbsapply(Ndim, function(iterdim) {
    hits <- replicate(Niter, {
        space_samp <- runif(iterdim, 0, 1)
        sum(space_samp^2) < 1
    })
    sum(hits) / Niter
})

frac_plot <- ggplot() +
    geom_line(aes(x = Ndim, y = vol_frac)) +
    xlab("Dimensions") +
    ylab("Fraction of Volume") +
    ggtitle("Fraction of Volume Occupied by Hypersphere in Hypercube vs Dimensionality of Space")

ggsave(frac_plot, file = "volfrac_plot.pn", height = 10, width = 14)

```

Figure 3 Curse of Dimensionality R code (available in appendix ii)

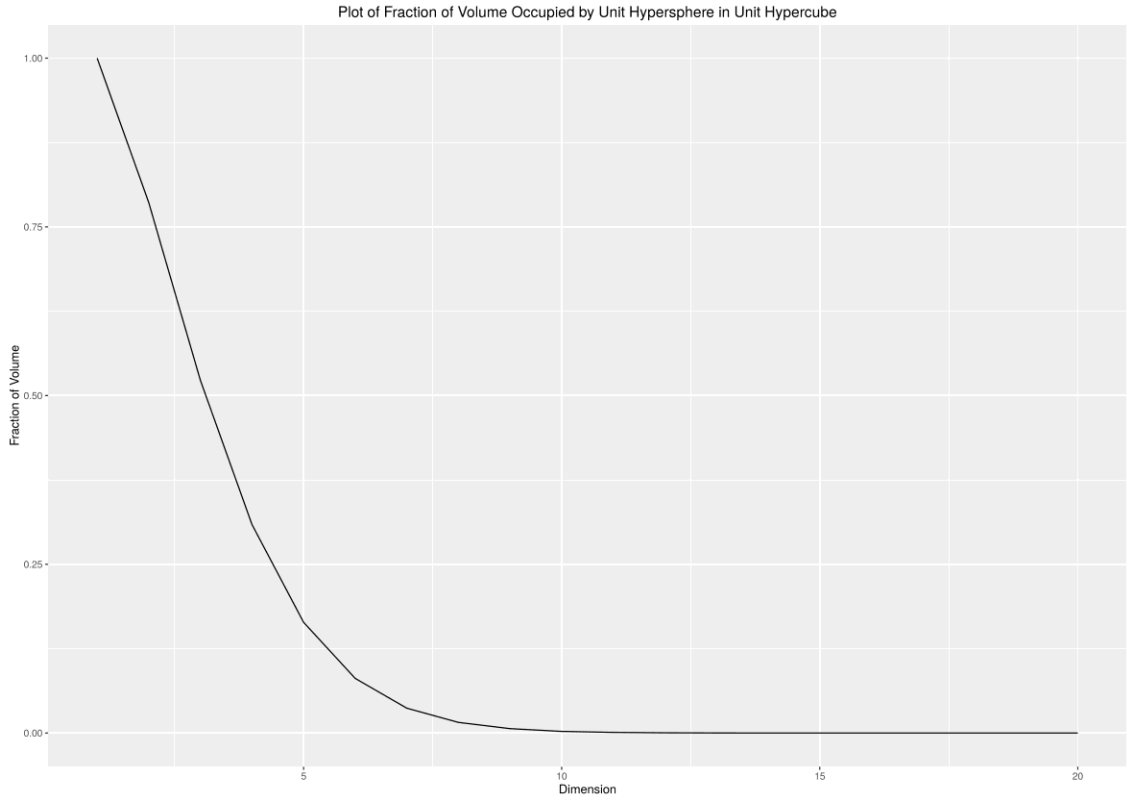


Figure 4 Curse of Dimensionality output

Two proposals for outlier detection in categorical data include (Akoglu, et al., 2012) and (Das & Schneider, 2007) which address the problem of finding anomaly patterns by building a Bayes net that represents a baseline distribution. Because of the high computational costs, they restrict their methods to only working with one or two component rules.

Visualization

Friendly in (Friendly, 2006), gives us a brief history of visualization from the time of cave drawings right up to our modern age highly complex diagrams and plots. In his work, Friendly makes special note of the contributions of Francis Galton [1822-1911] to data visualization and statistical graphics. It becomes obvious from this work that many of the graphs and plots we still use to this day are essentially the same but just computationally more complex. This is illustrated to effect in Figure 5 & Figure 6.

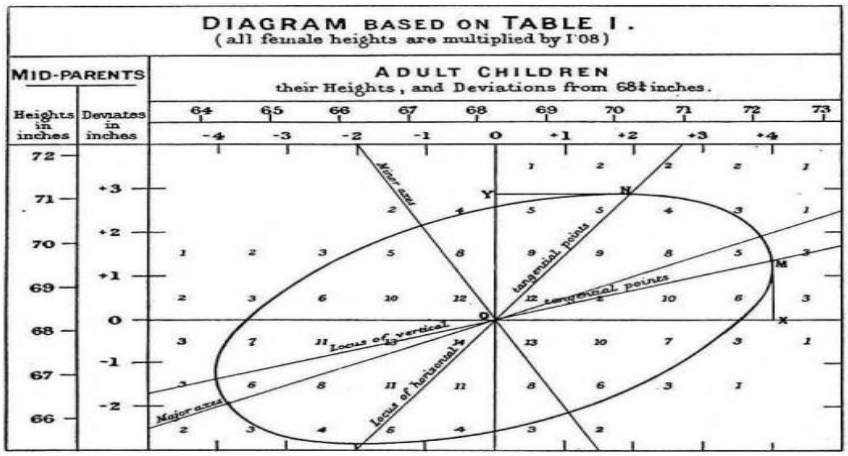


Figure 5 Galton's smoothed correlation diagram for the data on heights of parents and children, showing one ellipse of equal frequency circa 1886. (Friendly, 2006)

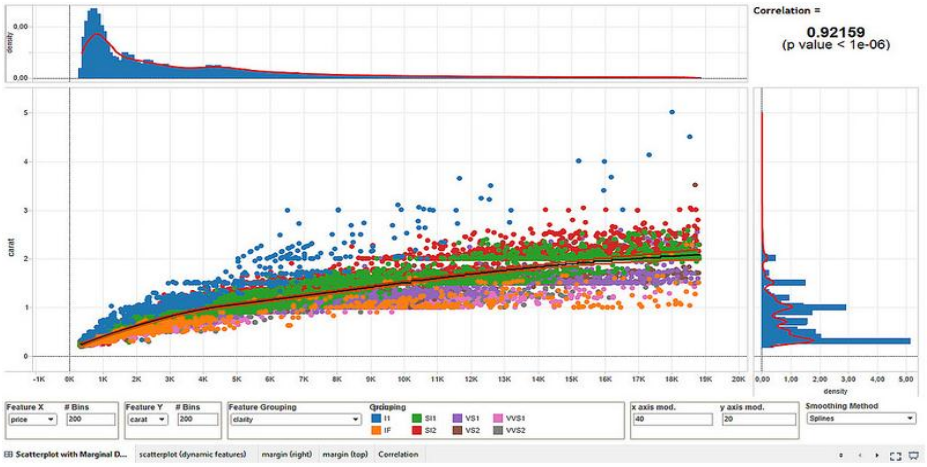


Figure 6 Tableau, a modern frequency distribution plot.

While we can see that for the last 130 years, data visualization has been viewed as an important analytical tool for organizations, it is quickly being recognized as an essential aspect of effective communication. (Lindquist, 2011) tells us "*…visualization techniques loom as potentially important sense-making, analytic and communications tools for capturing and addressing complexity. The promise is that, if properly chosen and calibrated, they can show the breadth and evolutions of problems and interventions, permit more detailed explorations of facets and strands, as well as how these facets and strands link to the whole".*

(Liu, et al., 2015) Highlight the state of the art in visualization. They categorize visualization techniques based on the three transformation steps of the information visualization pipeline (Card, et al., 1999), namely, data transformation, visual mapping, and view transformation.

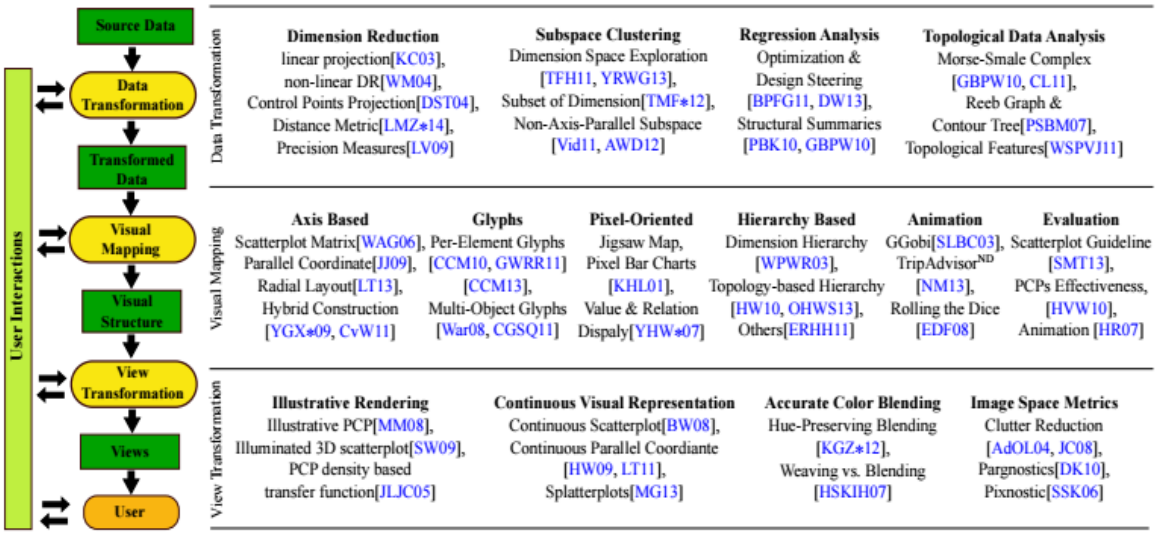


Figure 7 Categorization based on transformation steps within the information visualization pipeline (Liu, et al., 2015, p. 3)

Figure 7 gives us a comprehensive catalogue of tried and tested visualization techniques but for this project its main value is as a good roadmap as to the process that the final code should follow.

Technologies

The majority of work on this project has been completed using R. R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories by John Chambers and colleagues. The principal research materials and tutorials for R have come from (Lantz, 2013), (Chambers, 2008) and the R-Bloggers website (Assorted, 2016).

As per (R_Foundation, 2016), "R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes:

- an effective data handling and storage facility,
- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either on-screen or on hardcopy, and

- a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.”

The additional technology used was Tableau. Tableau Desktop is used to visualize and analyze data, create workbooks, visualizations and dashboards. All tutorials for Tableau came from their free online website.

Alternative Technologies explored

During the course of the academic year, for an Applied AI project I was asked to provide an alternative approach to the main project utilizing some form of Applied AI. I chose to use IBM Watson Analytics.

To sum up the findings of that project:

A lot of the explorations are fairly straightforward stuff that any spreadsheet can accomplish trivially such as “How many messages are there?”. But where Watson shines is in questions such as “What is the number of Timestamp for each Component?”. In a matter of seconds, we can easily visualise which component has had the most unique events during our data set as in figure 8 below.

As can be seen in figure 9 below, Watson starts to cluster data and has separated the injected outlier. Unfortunately, to further explore the data with Watson would require me learn Node.js so I could upload a corpus of the test data to Watson. This proved to be impractical due to time constraints but should be seen as a strong candidate for future work.

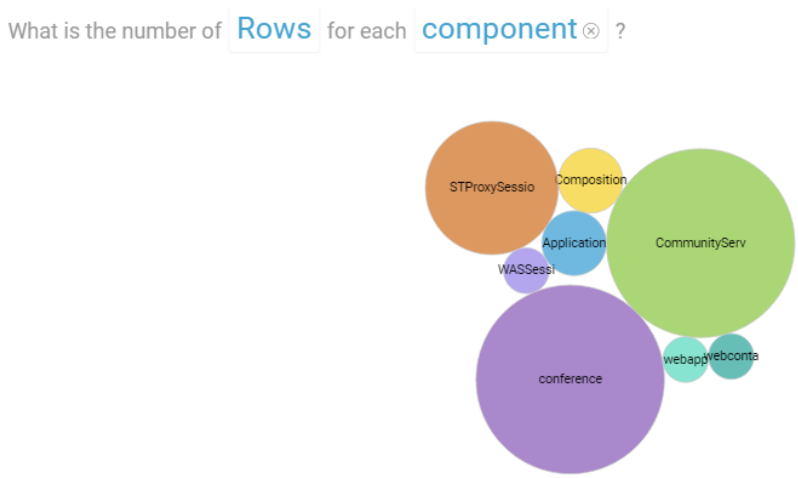


Figure 8 Watson Output 1

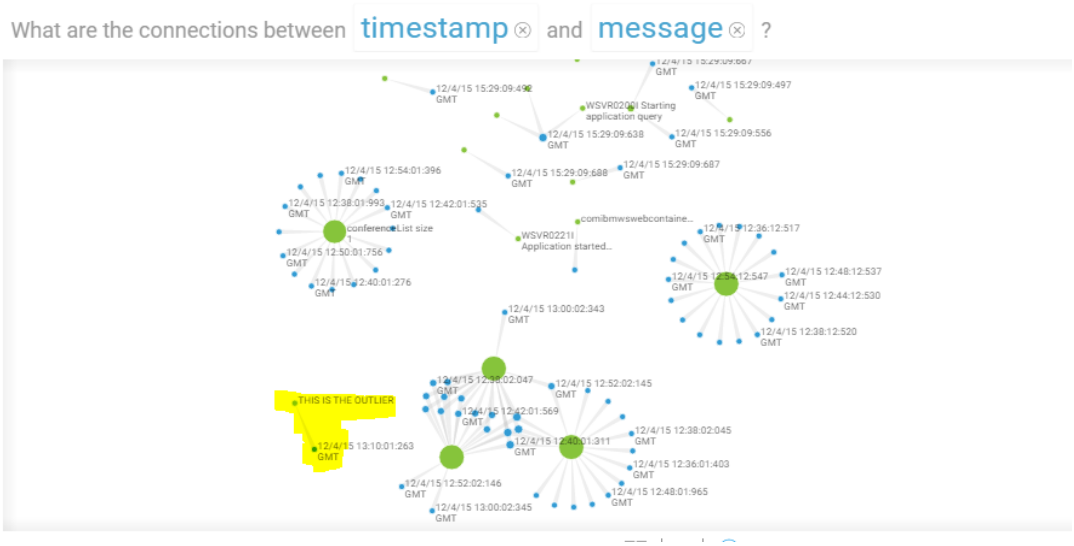


Figure 9 Watson Clustering on cleaned data.

Methodology

The methodology that best describes the development of this project is an Iterative Incremental Model. As most of the technologies and concepts were new to me this allowed for a developing understanding of the project.

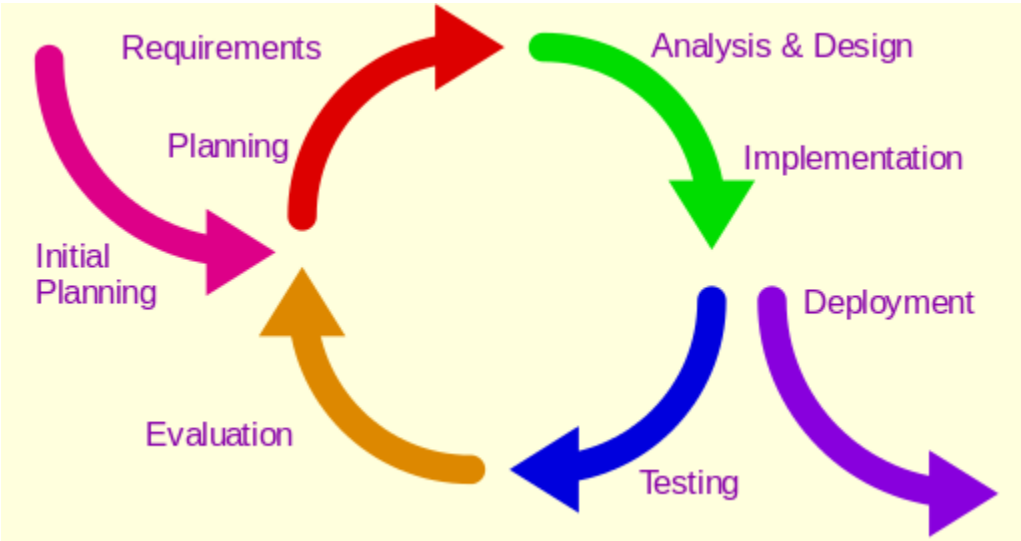


Figure 10 Iterative Incremental model of software development

As (Larman & Basili, 2003) explain, the basic idea behind this model is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental), allowing developers to take advantage of what was learned during development of earlier parts or versions of the system. Figure 10 illustrates that this is a continuous cycle until deployment. Learning comes from both the development and use of the system, where possible key steps in the process start with a simple implementation of a subset of the software requirements and iteratively enhance the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added.

Initial planning

Initial time planning for this project involved mostly guesswork due to the nebulous nature of the initial project scope. As subsequent iterations took place and the scope of the project took shape it became easier to get ideas as to how long each iteration should take

Requirements Gathering

Requirements gathering during the project was a continuous process involving bimonthly meetings with John O'Connor and other representatives of IBM. At first the requirements were quite vague but over the course of the iterative process they became clearly defined and are covered in more detail in (O'Connor, 2015)

Context of use

The application will be used to analyze issues that arise in the IBM Sametime servers. Users will be the IBM DevOps team and the IBM Connections Chat team.

Scenario of use

"When would this application be used? When ops specialist is analysing an issue and looking for the smoking guns." (O'Connor, 2015)

Based on the above statement, the application should be the first stop when a user needs to quickly analyze a large data set for potential issues.

Data requirements

For the purpose of this project, input data should be in the form of a Java Server Log File. The server should be running a live cloud system for enterprise level communication and collaboration (AKA IBM Sametime / Connections chat). It is foreseen that data sets could scale to 20+ servers operating 24/7 producing logs.

User requirements

End users would be Connections Chat team ops specialists who would be the owners and controllers of all related data.

Environmental requirements

The system should be able to run on Windows 7, the latest Windows iteration and on Linux Operating systems.

Requirements for effectiveness, efficiency and satisfaction

Due to the general intuitive nature of outlier detection as highlighted under Evaluation of Outliers section and the exploratory/academic nature of this project, there was no specific satisfaction metric placed on me by the client.

It was suggested by the client that when faced with extremely large data sets (1gb+) the algorithms should be allowed to run for up to 48 hours on an IBM® BigInsights™ on Cloud. This gives an idea of the resources the client would be willing to devote to the final algorithms if they proved to be useful.

Analysis & Design

Use Case

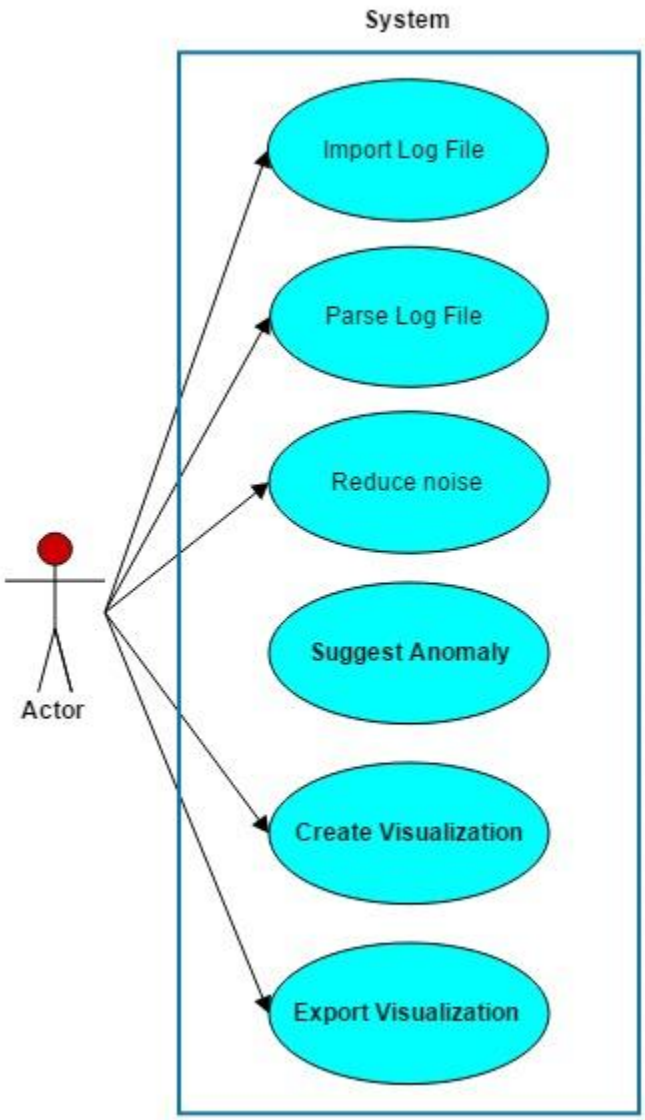


Figure 11 Overall Use Case Diagram

Logical Architectural View

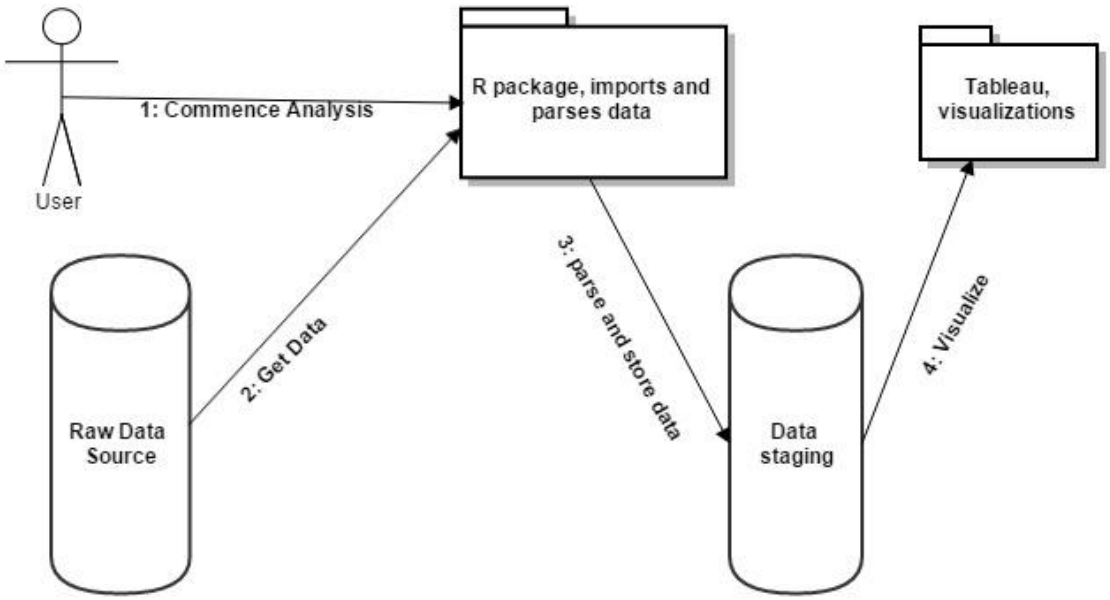


Figure 12 Logical Architecture View

Implementation

In this section, the main parts of the code will be explained using code snippets where appropriate. The majority of code was developed and tested using RStudio.

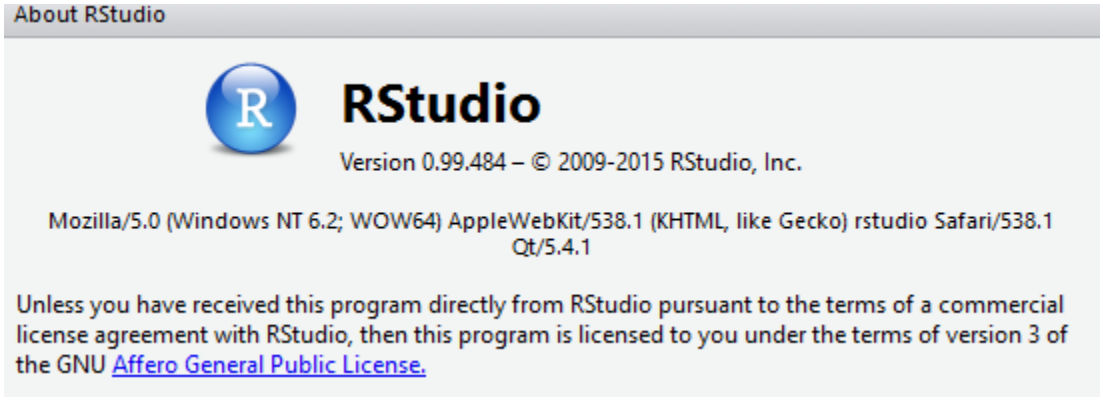


Figure 13 About R Studio

The primary use case for this project targets non-standard log files as in Figure 14. The initial block of code cleans this data into a more useable form for us to analyse. As can be seen the format changes at certain lines. For example, from line 181-192, the log is giving messages about the starting environment on the working server. These messages are not relevant to this exercise and so should be cleaned out.

```
174 BHT6A-stpnode1a logs_STProxyServer_SystemOut [12/4/15 13:17:05:875 GMT] 00000039 distSecurityC I   securityServiceStarted is false
175 BHT6A-stpnode1a logs_STProxyServer_SystemOut [12/4/15 13:17:05:888 GMT] 00000039 CGBridgeServi I   CWRCB0103I: The core group bridge service has stopped.
176 BHT6A-stpnode1a logs_STProxyServer_SystemOut [12/4/15 13:17:05:890 GMT] 00000039 DragDropDeplo I   CWLDD0004I: Stopping monitored directory application deploy
177 BHT6A-stpnode1a logs_STProxyServer_SystemOut [12/4/15 13:17:05:890 GMT] 00000039 DragDropDeplo I   CWLDD0005I: Monitored directory application deployment serv
178 BHT6A-stpnode1a logs_STProxyServer_SystemOut [12/4/15 13:17:05:903 GMT] 00000039 TCPChannel    I   TCPC0002I: TCP Channel TCPInboundChannel_ipcc.Default_IPC_C
179 BHT6A-stpnode1a logs_STProxyServer_SystemOut [12/4/15 13:17:06:085 GMT] 00000039 FailureScopeC A   WTRN0105I: The transaction service has shutdown successfull
180 BHT6A-stpnode1a logs_STProxyServer_SystemOut [12/4/15 13:17:06:098 GMT] 00000039 ServerCollabo A   WSVR0024I: Server STProxyServer stopped
181 BHT6A-stpnode1a logs_STProxyServer_SystemOut ************ Start Display Current Environment ************
182 BHT6A-stpnode1a logs_STProxyServer_SystemOut WebSphere Platform SCRUB_2_IP_ADDRESS [ND SCRUB_2_IP_ADDRESS cf031430.01] running with process name stpnode1aProx
183 BHT6A-stpnode1a logs_STProxyServer_SystemOut Host Operating System is Linux, version 2.6.32-SCRUB_2_IP_ADDRESS 16.x86_64
184 BHT6A-stpnode1a logs_STProxyServer_SystemOut Java version = 1.6.0, Java Runtime Version = pxa6460_26sr8ifx-20140630_01 (SR8), Java Compiler = j9jit26, Java VM
185 BHT6A-stpnode1a logs_STProxyServer_SystemOut was.install.root = /opt/IBM/WebSphere/AppServer
186 BHT6A-stpnode1a logs_STProxyServer_SystemOut user.install.root = /opt/IBM/WebSphere/AppServer/profiles/STPAppProfile
187 BHT6A-stpnode1a logs_STProxyServer_SystemOut Java Home = /opt/IBM/WebSphere/AppServer/java/jre
188 BHT6A-stpnode1a logs_STProxyServer_SystemOut ws.ext.dirs = /opt/IBM/WebSphere/AppServer/java/lib:/opt/IBM/WebSphere/AppServer/profiles/STPAppProfile/classes:/
189 BHT6A-stpnode1a logs_STProxyServer_SystemOut Classpath = /opt/IBM/WebSphere/AppServer/profiles/STPAppProfile/properties:/opt/IBM/WebSphere/AppServer/propertie
190 BHT6A-stpnode1a logs_STProxyServer_SystemOut Java Library path = /opt/IBM/WebSphere/AppServer/lib/native/linux/x86_64/:/opt/IBM/WebSphere/AppServer/java/jre/l
191 BHT6A-stpnode1a logs_STProxyServer_SystemOut Orb Version = IBM Java ORB build orb626ifx-20140404.00 (IX90144)
192 BHT6A-stpnode1a logs_STProxyServer_SystemOut ************ End Display Current Environment ************
193 BHT6A-stpnode1a logs_STProxyServer_SystemOut [12/4/15 15:28:10:037 GMT] 00000001 ManagerAdmin  I   TRAS0017I: The startup trace state is *=info.
194 BHT6A-stpnode1a logs_STProxyServer_SystemOut [12/4/15 15:28:10:039 GMT] 00000001 ManagerAdmin  I   TRAS0111I: The message IDs that are in use are deprecated
195 BHT6A-stpnode1a logs_STProxyServer_SystemOut [12/4/15 15:28:10:267 GMT] 00000001 ModelMgr      I   WSVR0800I: Initializing core configuration models
196 BHT6A-stpnode1a logs_STProxyServer_SystemOut [12/4/15 15:28:11:983 GMT] 00000001 ComponentMeta I   WSVR0179I: The runtime provisioning feature is disabled. Al
```

Figure 14 IBM Connections Sametime log (test file)

Set up environment

Firstly, to measure the performance of the code it's useful to get create a variable that will hold various timestamps as per "timeStart = Sys.time". Next we need to tell the R environment which libraries will be used during the script. R packages are installed into libraries, which are directories in the file system containing a subdirectory for each package installed there. These libraries can be common libraries or user libraries. This is similar to Java's Import command. And finally we should tell the R script which is our working directory.

```
timeStart = Sys.time()
timeStart

library(data.table);
library(ggplot2);
library(xlsx);
library(utils);
library(tm)
library(wordcloud)
library(e1071)
library(dplyr)
library(FactoMineR)
library(MASS)
library(stringr)
library(SnowballC)
library(slam)
library(reshape2)
library(tsne)
library(Rtsne)
library(cluster)
library(bigmemory)


setwd("C:/Users/John/Desktop/Project/MediumTest")
```

Figure 15 Setting up R environment

Cleaning the Data

We loop over the entire log file performing regular expression (regex) checks to remove any startup messages. In the event that any unusual or anomalous behaviour was to happen during the startup procedure, it should be trivial to find. Next we concatenate each line into a .txt file while outputting to console a progress count of where the program is (counters m&n)

```
30  toplevel_regex <- '([A-Za-z0-9\\-]+) ([A-Za-z0-9_\\-]+) (.*)';
31  #thread is a hex number always 8 digits long, component is always 13 chars long
32  lower_regex <- '\\[(.+?)\\] ([0-9a-f]{8}) (.{13}) ([A-Z]) (.*)';
33
34  n=1;
35  m=1;
36
37  while(length(linestr <- readLines(fileconn, n = 1, warn = TRUE)) > 0) {
38    test_match <- gsub(toplevel_regex, '\\3', linestr);
39
40    if(grepl("^\\[.*?\\]", test_match)) {
41      part1 <- gsub(lower_regex, '\\1', test_match);
42      part2 <- gsub(lower_regex, '\\2', test_match);
43      part3 <- gsub(lower_regex, '\\3', test_match);
44      part4 <- gsub(lower_regex, '\\4', test_match);
45      part5 <- gsub(lower_regex, '\\5', test_match);
46
47      part5 <- removePunctuation(part5);
48      cat( part1, part2, part3, part4, part5, "\n", sep = ',', file = "mediumTestGrepped.txt"
49      m=m+1;
50
51    } else {
52      ### Log entry is startup mode, which you parse differently or
53      ### discard completely.
54
55      cat(paste("STARTUP ENTRY:",n, " now at line:",m,  test_match, "\n"));
56      n=n+1;
57    }
58
59  }
60
```

The m and n counters allow us to monitor where the code currently is in the log file. They also give us quick metrics as to how many “startup” entries are in the log. When this server was actually running it restarted several times. This could have flagged errors or, as per information from the client, could simply have been standard operating procedures.

Building a Data Table

Next we want to get the data into a data.table. “Data.table inherits from data.frame. It offers fast subset, fast grouping, fast update, fast ordered joins and list columns in a short and flexible syntax, for faster development. It is inspired by A[B] syntax in R where A is a matrix and B is a 2-column matrix. Since a data.table is a data.frame, it is compatible with R functions and packages that only accept data.frame. “ (Anon., 2016)

We also filter common terms that have been identified by the client as noise.

And finally we get a finish time which allows us to track the time cost of the algorithm.

At various points during the code we can insert the standard head(), tail() and summary() functions to output to the console information about the data. [1]

```
63  logfile <- 'mediumTestGrepped.txt';
64  snippet_dt <- fread(logfile, sep = ',');
65  summary (snippet_dt)
66  ### Load and rename importantt columns
67  names(snippet_dt) <- c( "timestamp", "thread_id", "component","level",  "message", "Null?")
68  summary(snippet_dt)
69
70  #remove the null column,
71  snippet_dt$`Null?` <-NULL
72  head(snippet_dt)
73  tail(snippet_dt)
74
75  #Stripping very common terms (noise?) from the dataframe
76  #This is currently hard-coded but these terms need to be dynamically deduced or user input
77  snippet_dt <-(filter(snippet_dt, !grepl("conferenceList size", message)))
78  snippet_dt <-(filter(snippet_dt, !grepl("WIO channel was closed, reason [-2147483084]", message)))
79  snippet_dt <-(filter(snippet_dt, !grepl("CLFRX0027E", message)))
80  snippet_dt <-(filter(snippet_dt, !grepl("channel was closed",message)))
81  snippet_dt <-(filter(snippet_dt, !grepl("MessageDispatcher is null",message)))
82  snippet_dt <-(filter(snippet_dt, !grepl("Policy: com.notnoop.apns.internal.ReconnectPolicies$Every
83  snippet_dt <-(filter(snippet_dt, !grepl("userpref.UserPreferenceService static code block key: 778
84  snippet_dt <-(filter(snippet_dt, !grepl("Policy again: com.notnoop.apns.internal.ReconnectPolicies
85  snippet_dt <-(filter(snippet_dt, !grepl("factory: com.ibm.jsse2.SSLSocketFactoryImpl@28547877 host
86  snippet_dt <-(filter(snippet_dt, !grepl("notnoop.apns.internal.ReconnectPolicies$EveryHalfHour",me
87  summary(snippet_dt)
88
89  timeFinish = Sys.time()
90  timeFinish
```

[1] One thing that drives me crazy is when a computer program is running but the user has no idea what is actually happening. Has the code crashed? Is it actually doing something? A simple output to console, much like a progress bar, gives the user some encouragement that things are actually happening.

Convert to Term Document Matrix

The majority of the data we want to examine is in the "message" column. So our next task is to take the information in our message column and form that into a matrix of terms. [2]

A Term Document Matrix is defined by (Weber, 2013) as "…a two-dimensional array, with rows and columns. In the TDM, rows represent documents, columns represent terms (in the collection vocabulary). Cell values are term frequency counts, or, more generally, "score" attached to a term for a document"

Further cleaning is performed by removing numbers, setting all to lower case and stemming etc.

```
messageSplit <- strsplit(snippet_dt$message, " ")
doc.vec <-VectorSource(messageSplit)
doc.corpus <-Corpus(doc.vec)

doc.corpus <- tm_map(doc.corpus, tolower)
doc.corpus <- tm_map(doc.corpus, removeNumbers)
doc.corpus <- tm_map(doc.corpus, removeWords, stopwords("english"))
doc.corpus <- tm_map(doc.corpus, stemDocument)
doc.corpus <- tm_map(doc.corpus, PlainTextDocument)

inspect(doc.corpus[50:60])

TDM <- TermDocumentMatrix(doc.corpus)
TDM
```

Figure 16 conversion to Term Document Matrix

[2] At this point it would have been possible to take the research down a different route and explored cosine similarity measures as outlined by Simon Caton during a meeting and also as explained by (Weber, 2013, p. 16) however as I was making headway in converting the TDM to graph form and mapping that out I followed that route.

Thresholds & Associations

In the following part of the code we can see some useful (but currently hardcoded) functions which allow us to set thresholds for assorted values in our TDM. We can also use the ability of R to quickly transpose from data.frame to data.Matrix

```
88  # keep only  words that occur >3 times in all docs
89  TDM2 <- TDM[,which(colTotals > 3)]
90  TDM2
91  #find terms that occurred more than 2 times
92  findFreqTerms(TDM2, 2)
93  #find associations between words, example is outlier
94  findAssocs(TDM, "outlier", 0.99)
95  # function t() is transposition "changing columns to rows" and
96  TDM2 = as.data.frame( t(as.matrix(TDM2)) )
97  TDM2M = as.matrix(t(as.data.frame(TDM2)))
98  TDM = as.data.frame( t(as.matrix(TDM2)) )
99  TDMM = as.matrix(t(as.data.frame(TDM2)))
```

Convert to term-term Adjacency matrix

Adjacency matrices have been previously explained in the Background section.

```
100 # change it to a Boolean matrix
101 TDMM[TDMM>=1] <- 1
102 # transform into a term-term adjacency matrix
103 termMatrix <- TDMM %*% t(TDMM)
104 # inspect terms numbered 5 to 10
105 termMatrix[5:10,5:10]
```

Convert to Graph

The layout.fruchterman.reingold & layout.kamada.kawai are based on the force directed graph algorithms introduced by (Fruchterman & Reingold, 1991) & (Kamada & Kawai, 1989). The purpose of a force directed graph is "*…to position the nodes of a graph in two-dimensional or three-dimensional space so that all the edges are of more or less equal length and there are as few crossing edges as possible, by assigning forces among the set of edges and the set of nodes, based on their relative positions, and then using these forces either to simulate the motion of the edges and nodes or to minimize their energy."*- (Kobourov, 2012)

NB It is at this point that the greatest weakness of my approach to this project has become apparent. Force directed graph drawing has many strengths but it's greatest weakness is that it is considered to have a running time equivalent to $O(n^3)$ where n is the number of nodes of the input graph. All is not lost though as (Kobourov, 2012, p. 18) informs us "…force directed algorithms will likely continue to be the method of choice. In the Further Work Chapter, I will expand on solutions to this issue.

```
106 library(igraph)
107 # build a graph from the above matrix
108 g <- graph.adjacency(termMatrix, weighted=T, mode = "undirected")
109 # remove loops
110 g <- simplify(g)
111 # set labels and degrees of vertices
112 V(g)$label <- V(g)$name
113 V(g)$degree <- degree(g)
114 # set seed to make the layout reproducible
115 set.seed(3952)
116 layout1 <- layout.fruchterman.reingold(g)
117 layout2 <- layout.kamada.kawai(g)
118 layout3 <- layout.auto(g)
```

Formatting the graph

The variables here need some tinkering in order to get the graph attractive. Again this will be future work, in Shiny probably using user input sliders.

```
128 #formatting from here , V = vertices, E= edges
129 V(g)$label.cex <- 2.2 * V(g)$degree / max(V(g)$degree)+ .2
130 V(g)$label.color <- rgb(0, 0, .2, .8)
131 V(g)$frame.color <- NA
132 egam <- (log(E(g)$weight)+.8) / max(log(E(g)$weight)+.4)
133 E(g)$color <- rgb(.5, .5, 0, egam)
134 E(g)$width <- egam
135 # plot the graph in layout1
136 plot(g,axes=TRUE,  layout=layout1)
137
138 #tkplot is an external interactive plot
139 tkplot(g, layout=layout.kamada.kawai)
```

Shiny (work in progress as of 08/06/16)

Shiny is a web application framework by RStudio. It allows for fast interactive builds to the browser with no knowledge of HTML, CSS or JavaScript. Shiny apps have two components: a user interface script and a server script. At the moment, the code below transfers the graph plotted above to a web app and allows the user to get the coordinates of vertices on the graph.

The plan is to take those coordinates, compare them to the adjacency matrix coordinates which should correspond to terms in the TDM thus allowing the end user to find the outliers directly from the graph output.

```
163  library(shiny)
164
165  ui <- basicPage(
166    plotOutput(
167      "plot1",
168      click = "plot_click",
169      dblclick = "plot_dblclick",
170      hover = "plot_hover",
171      brush = "plot_brush"
172    ),
173    verbatimTextOutput("info")
174  )
```

Figure 17 Shiny UI script with inputs

```
server <- function(input, output) {
  output$plot1 <- renderPlot({
    plot(
      g, layout = layout1, rescale = FALSE, axes = TRUE, xlim = range(layout1[,1]), ylim = range(layout1[,2])
    )
  })

  output$info <- renderText({
    xy_str <- function(e) {
      if (is.null(e))
        return("NULL\n")
      paste0("x=", round(e$x, 8), " y=", round(e$y, 8), "\n")

    }
    xy_range_str <- function(e) {
      if (is.null(e))
        return("NULL\n")
      paste0(
        "xmin=", round(e$xmin, 8), " xmax=", round(e$xmax, 8),
        " ymin=", round(e$ymin, 8), " ymax=", round(e$ymax, 8)
      )
    }

    paste0(
      "click: "   , xy_str(input$plot_click),
      "dblclick: ", xy_str(input$plot_dblclick),
      "hover: "   , xy_str(input$plot_hover),
      "brush: "   , xy_range_str(input$plot_brush)
    )
  })
}
```

Figure 18 Shiny Server Code

Evaluation and Testing

I believe my approach to this problem has been innovative. After extensive study I cannot find anybody who has taken the approach of finding outliers specifically in Log Files through adjacency graphs. Perhaps this is because of the scaling problem? But at smaller scales the approach is very effective at allowing natural human visual processing to see things sticking out from the norm.

Going from this…

```
BHT6A-stpnodela logs_STProxyServer_SystemOut [12/4/15 13:17:05:875 GMT] 00000039 distSecurityC I   securityServiceStarted is false
BHT6A-stpnodela logs_STProxyServer_SystemOut [12/4/15 13:17:05:888 GMT] 00000039 CGBridgeServi I   CWRCB0103I: The core group bridge service has stopped.
BHT6A-stpnodela logs_STProxyServer_SystemOut [12/4/15 13:17:05:890 GMT] 00000039 DragDropDeplo I   CWLDD0004I: Stopping monitored directory application deploy
BHT6A-stpnodela logs_STProxyServer_SystemOut [12/4/15 13:17:05:890 GMT] 00000039 DragDropDeplo I   CWLDD0005I: Monitored directory application deployment serv
BHT6A-stpnodela logs_STProxyServer_SystemOut [12/4/15 13:17:05:903 GMT] 00000039 TCPChannel    I   TCPC0002I: TCP Channel TCPInboundChannel_ipcc.Default_IPC_C
BHT6A-stpnodela logs_STProxyServer_SystemOut [12/4/15 13:17:06:085 GMT] 00000039 FailureScopeC A   WTRN0105I: The transaction service has shutdown successfull
BHT6A-stpnodela logs_STProxyServer_SystemOut [12/4/15 13:17:06:098 GMT] 00000039 ServerCollabo A   WSVR0024I: Server STProxyServer stopped
BHT6A-stpnodela logs_STProxyServer_SystemOut ************ Start Display Current Environment ************
BHT6A-stpnodela logs_STProxyServer_SystemOut WebSphere Platform SCRUB_2_IP_ADDRESS [ND SCRUB_2_IP_ADDRESS cf031430.01] running with process name stpnodelaProx
BHT6A-stpnodela logs_STProxyServer_SystemOut Host Operating System is Linux, version 2.6.32-SCRUB_2_IP_ADDRESS 16.x86_64
BHT6A-stpnodela logs_STProxyServer_SystemOut Java version = 1.6.0, Java Runtime Version = pxa6460_26sr8ifx-20140630_01 (SR8), Java Compiler = j9jit26, Java VM
BHT6A-stpnodela logs_STProxyServer_SystemOut was.install.root = /opt/IBM/WebSphere/AppServer
BHT6A-stpnodela logs_STProxyServer_SystemOut user.install.root = /opt/IBM/WebSphere/AppServer/profiles/STPAppProfile
BHT6A-stpnodela logs_STProxyServer_SystemOut Java Home = /opt/IBM/WebSphere/AppServer/java/jre
BHT6A-stpnodela logs_STProxyServer_SystemOut ws.ext.dirs = /opt/IBM/WebSphere/AppServer/java/lib:/opt/IBM/WebSphere/AppServer/profiles/STPAppProfile/classes:/
BHT6A-stpnodela logs_STProxyServer_SystemOut Classpath = /opt/IBM/WebSphere/AppServer/profiles/STPAppProfile/properties:/opt/IBM/WebSphere/AppServer/propertie
BHT6A-stpnodela logs_STProxyServer_SystemOut Java Library path = /opt/IBM/WebSphere/AppServer/lib/native/linux/x86_64/:/opt/IBM/WebSphere/AppServer/java/jre/l
BHT6A-stpnodela logs_STProxyServer_SystemOut Orb Version = IBM Java ORB build orb626ifx-20140404.00 (IX90144)
BHT6A-stpnodela logs_STProxyServer_SystemOut ************ End Display Current Environment ************
BHT6A-stpnodela logs_STProxyServer_SystemOut [12/4/15 15:28:10:037 GMT] 00000001 ManagerAdmin  I   TRAS0017I: The startup trace state is *=info.
BHT6A-stpnodela logs_STProxyServer_SystemOut [12/4/15 15:28:10:039 GMT] 00000001 ManagerAdmin  I   TRAS0111I: The message IDs that are in use are deprecated
BHT6A-stpnodela logs_STProxyServer_SystemOut [12/4/15 15:28:10:267 GMT] 00000001 ModelMgr     I   WSVR0800I: Initializing core configuration models
BHT6A-stpnodela logs_STProxyServer_SystemOut [12/4/15 15:28:11:983 GMT] 00000001 ComponentMeta I   WSVR0179I: The runtime provisioning feature is disabled. Al
```

Figure 19 Raw Data

…to this…

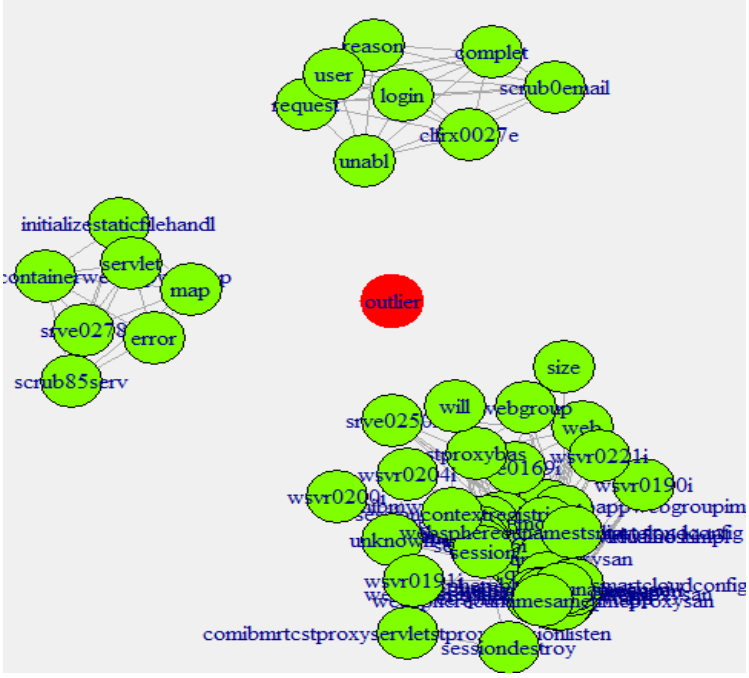


Figure 20 Successful visualization

… I believe to have been a significant success.

Client evaluation and Feedback

The following is feedback directly from the client, John BT O'Connor of IBM

"***Difficulty**:*

This is a very ambitious project with a level of difficulty that would be more akin to a master's thesis. The problem we set John was to find a needle in the haystacks of cloud service logs. To identify what was routine and what was interesting and then point out the interesting things in the logs.

This involved big data scale + cloud architectures + natural language processing.

On top of that we wanted the output visualised for human consumption so that added UX and visualisation into the mix.

Then on top of that again we wanted to understand performance.

Software Engineering:

John went to great lengths to ensure clarity of purpose. He really nailed down the requirements and the business need. After initial proposals from us he went off to digest and experiment. We had a few rounds of clarifying and understanding until we and John were happy with the requirements.

He would stand at whiteboards to get his points across and ensure both sides had the same concepts imprinted in our heads. He understood the power of pictures to ensure such common purpose.

Innovation:

Let there be no doubt here. The project is innovative. Finding the clues to an outage or loss of functionality in GBs of logs.

Furthermore, John learnt new technologies in order to come to a solution (R, Data Science, Natural Language Processing techniques) and new tools (Tableau, Watson, ...). He researched well, he experimented with various data and NLP techniques to find the ones that applied best to this project.

He came up with some non-obvious results. e.g.: applying term adjacency; Watson on Bluemix can actually achieve what the much more specialised Tableau can do and more.

Presentation

John came in to keep us up to date and seek direction at requirement gathering; during data analysis; during implementation. At each stage John was excellent at getting complex points across concisely.

He presented his projects "position" very effectively such that we were able to discuss issues and propose adjustments and expansions to the project every time.

He comes across as very knowledgeable and motivated. And was able to understand the audience (us) and pitch accordingly.

Implementation and Testing

He has come up with non-obvious and promising results. We will definitely go further with this. He got the log analysis working and was able to write it in a way that allowed experimentation.

He worked on sample data and was able to show the outliers we were looking for in the data.

He wants to work on the performance issues. It is currently order $O(n^{2)}$ *in analysis and* $O(n^{3)}$ *in graphing. I suspect not all final year projects discover this, never mind flag it as something to work on. He has taken the time to measure based on increasing data sets.*

Finally, I want to comment on John's personable nature throughout the project. He was patient with our slow responses for meetings but firm and diplomatic about getting them set up. He prepared well for each meeting such that each one of them was completed before the hour we had reserved. " (O'Connor, 2016)

Testing Methodologies

During later iterations of the project the following testing methodologies were used. There are 8 examples of test scripts followed in this section as well as results of those scripts and suggested fixes if any were required.

Black box testing: examining functionality without any knowledge of internal implementation, without seeing the source code. The testers are only aware of what the software is supposed to do, not how it does it. The black box tests used were a combination of *Use Case Tests* and *Fuzz testing*.

White box testing: tests internal structures or workings of a program, as opposed to the functionality exposed to the end-user. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The white box tests used were *Fault injection* and *Mutation testing*

Usability testing: check if the user interface is easy to use and understand. Can a person with zero knowledge of the project quickly recognise outliers as significant data points?

The following pages of this document contain a Test Script that follow a suggested template from Cambridge University press.

The Test Script template contains the following sections:

AUT Name - the definitive name of the Application Under Test (front sheet only)

AUT Version - the definitive version information for the Application Under Test (front sheet only)

Iteration ID - the unique identifier for the iteration this test is being conducted in (front sheet only)

Date of Test - the planned start date of testing (front sheet only)

Test ID - the unique identifier for the test

Purpose of Test - a brief description of the purpose of the test including a reference where appropriate to the requirement that is to be tested (consider providing references to the requirements specification, design specification, user guide, operations guide and/or installation guide), as well as any dependencies from or to other Test Scripts/Test Cases

Test Environment - a brief description of the environment under which the test is to be conducted (may include a description of the state of the AUT at the start of this test,

details regarding the platform or operating system, as well as specific information about data used in this test)

Test Steps - concise, accurate and unambiguous instructions describing the precise steps the Tester must take to execute the test, including navigation through the AUT as well as any inputs and outputs

Expected Result - a brief and unambiguous description of the expected result of executing the test.

Actual Result - a brief and unambiguous description of the actual result of executing the test.

White Box Testing

White Box Test 1 (*front sheet*)			
AUT Name	Data Cleaning	**Version**	2.0
Iteration ID	2.0	**Date of Test**	17/01/16

Test ID	WB1
Purpose of Test	To Ensure that: It is possible to upload log files to R It is possible to clean log files for conversion to data frame/table
Test Environment	The test environment is as follows: Client Hardware: HP pavilion laptop running windows 10 and R Studio latest edition Test data was a 200-line excerpt of log files
Test Steps	From R studio the tester should: Ensure that data set "smallTest.txt" is in working directory Run script Small test from line 1-58
Expected Result	On completing the above steps, the global environment in RStudio should have populated with data. There should be a data table called snippet_dt with 5 columns.
Actual result	The data.table snippet_dt was created but had 6 columns. A null column was created somehow?
Suggested action	Investigate why the extra column is created
Resolution	Still unsure as to why the extra column appeared. Temporary workaround is to remove 'null' column later in code

White Box Test 2 (*front sheet*)			
AUT Name	Corpus cleaning	**Version**	3.0
Iteration ID	3.0	**Date of Test**	27/04/16

Test ID	WB2
Purpose of Test	To Ensure that: It is possible to build and clean a corpus of data in preparation for creation of a term document matrix
Test Environment	The test environment is as follows: Client Hardware: HP pavilion laptop running windows 10 and R Studio latest edition Test data was a 200-line excerpt of log files
Test Steps	From R studio the tester should: Ensure that data set "smallTest.txt" is in working directory Run script Small test from line 82-94
Expected Result	On completing the above steps, the global environment in RStudio should have populated with a term document matrix that was clean of numbers, punctuation, stem words and was in lower case.
Actual result	Test passed without issues.
Suggested action	n/a
Resolution	n/a

White Box Test 3 (*front sheet*)			
AUT Name	Graph building on AWS with large data set	**Version**	3.0
Iteration ID	3.0	**Date of Test**	27/04/16

Test ID	WB3
Purpose of Test	To Ensure that: It is possible to graph a term-term adjacency matrix on a medium data set
Test Environment	The test environment is as follows: Client Hardware: AWS t2.large server instance Test data was a 10,000-line excerpt of log files
Test Steps	From R studio on AWS the tester should: Ensure that data set "mediumTest.txt" is in working directory Run script medium test from line 0-131
Expected Result	Within a reasonable time (under 1 hour was chosen arbitrarily) the code should begin drawing the graph
Actual result	Test was still running after 1 hours
Suggested action	Investigate efficiency of algorithm
Resolution	See notes in implementation on $O(n^3)$ running time of graphs.

Black Box Testing

Black Box Test 1 (*front sheet*)			
AUT Name	Upload Data file	**Version**	1.0
Iteration ID	1.0	**Date of Test**	11/11/15

Test ID	BB1
Purpose of Test	To Ensure that: It is possible to get data uploaded to system
Test Environment	The test environment is as follows: Client Hardware: HP pavilion laptop running windows 10 and R Studio latest edition Test data was a 200-line excerpt of log files called small.txt
Test Steps	From R studio the tester should: Ensure that data set “smallTest.txt” is in working directory Run script medium test from line beginning to end
Expected Result	Global environment should be populated with smallTest.txt
Actual result	File uploaded as expected
Suggested action	n/a
Resolution	n/a

Black Box Test 1 (*front sheet*)			
AUT Name	Upload Data file	**Version**	1.0
Iteration ID	1.0	**Date of Test**	11/11/15

Test ID	BB1
Purpose of Test	To Ensure that: It is possible to get data uploaded to system
Test Environment	The test environment is as follows: Client Hardware: HP pavilion laptop running windows 10 and R Studio latest edition Test data was a 200-line excerpt of log files called small.txt
Test Steps	From R studio on AWS the tester should: Ensure that data set “smallTest.txt” is in working directory Run script medium test from line beginning to end
Expected Result	Global environment should be populated with smallTest.txt
Actual result	File uploaded as expected
Suggested action	n/a
Resolution	n/a

Black Box Test 2 (*front sheet*)			
AUT Name	Data cleaning and corpus cleaning on BIG Data	**Version**	2.0
Iteration ID	2.0	**Date of Test**	02/12/16

Test ID	BB2
Purpose of Test	To Ensure that: Data is cleaned and corpus stemmed etc. in a timely fashion on cloud server
Test Environment	The test environment is as follows: Client Hardware: AWS t2.large server instance Test data was a 1.3Million-line excerpt of log files
Test Steps	From R studio on AWS the tester should: Ensure that data set "IBMLogt.txt" is in working directory Run script medium test from line 0-89
Expected Result	Global environment should be populated with TDM in under 20 minutes
Actual result	Global environment is populated with TDM in under 20 minutes
Suggested action	n/a
Resolution	n/a

Black Box Test 3 (*front sheet*)			
AUT Name	Graph output on local with small data	**Version**	3.0
Iteration ID	3.0	**Date of Test**	22/04/16

Test ID	BB3
Purpose of Test	To Ensure that: Data is graphed and outliers are clearly visible
Test Environment	The test environment is as follows: Client Hardware: HP pavilion laptop running windows 10 and R Studio latest edition Test data was a 200-line excerpt of log files called small.txt
Test Steps	From R studio the tester should: Ensure that data set smallTest.txt is in working directory **Inject an outlier into the Data** Run script medium test from line beginning to end
Expected Result	A graph should be produced with an outlier
Actual result	Global environment is populated with TDM in under 20 minutes
Suggested action	n/a
Resolution	n/a

Usability testing

Usability acceptance Test 1 (*front sheet*)			
AUT Name	Tableau bubble visualization of events	**Version**	1.0
Iteration ID	1.0	**Date of Test**	23/11/15

Test ID	UAT 1
Purpose of Test	To Ensure that: Events are successfully represented in a Tableau bubble chart
Test Environment	The test environment is as follows: Client Hardware: HP pavilion laptop running windows 10 and R Studio latest edition + Tableau latest edition Test data was a 200-line excerpt of log files called small.txt
Test Steps	From R studio the tester should: Ensure that data set smallTest.txt is in working directory Run script medium test from line beginning to end Import smalltest.CSV into Tableau workbook 1
Expected Result	A bubble graph showing all events in the log should be generated
Actual result	A bubble graph showing all events in the log IS generated
Suggested action	n/a
Resolution	n/a

Usability acceptance Test 2 (*front sheet*)			
AUT Name	Outliers recognised from R plot output	**Version**	3.0
Iteration ID	3.0	**Date of Test**	04/5/16

Test ID	UAT 2
Purpose of Test	To Ensure that: User can recognise outlier data points from R plotted graph
Test Environment	The test environment is as follows: Client Hardware: HP pavilion laptop running windows 10 and R Studio latest edition + Tableau latest edition Test data was a 200-line excerpt of log files called small.txt
Test Steps	From R studio the tester should: Ensure that data set smallTest.txt is in working directory Run script small test from line beginning to end Examine output and successfully identify outliers
Expected Result	The user should be able to point at the injected outlier on the plot
Actual result	5 users were able to point at the injected outlier on the plot
Suggested action	n/a
Resolution	n/a

Conclusion and Further Work

From my research for this project it becomes obvious to me that the field of Data Analytics is part science, part engineering and part intuitive art. Nowhere is this more obvious than when we focus on the topic of outlier visualization. Spotting that "one of these kids is doing their own thing, one of these kids is not the same" may be a skill we start to learn watching Sesame Street, but it is a skill that, with modern computers, we are becoming better and better at.

User testing has shown that the idea of using plotted adjacency graphs to find outlying data points is a solid one. Users, when shown the output, had no issue with quickly and correctly identifying the points of interest that they would want to investigate.

However, the concept has a serious flaw. Put simply, the entire concept from Term and Adjacency matrices to graphing of plots, suffers from scaling issues. As we add more data points the matrices grow and algorithms are running in order $O(n^2)$ and the graphs are worse in that the algorithms to run them, run in $O(n^3)$. This is disastrous for scale even with modern cloud computing resources.

But all is not lost. There has already been considerable proven work on dimensionality reduction and if further work on this project was to take place, I would make that the top priority. An exploration of principal component analysis (PCA) for the data matrices would be a good place to start making the project more efficient.

Next, the issue of graphs with over 1000 vertices must be addressed. (Kobourov, 2012, pp. 9-10) explains the work of several other researchers but draws special attention to the coarsening strategies of (Harel & Koren, 2001) as follows:

1. *Perform fine-scale relocations of vertices that yield a locally organized configuration.*
2. *Perform coarse-scale relocations (through local relocations in the coarse representations), correcting global disorders not found in stage 1.*
3. *Perform fine-scale relocations that correct local disorders introduced by stage 2.*

Another approach, under the advice of (Cooney & O'Leary, 2016) is to use methods currently popular in industry and that is to use one of the many algorithms based on the Barnes-Hut n-body simulation method (Barnes & Hut, 1986)

And Finally, I think there is huge value in continuing the work on interactive graphs with Shiny or with Tableau.

Appendices

i. Research Interview with Mick Cooney & Jamie O'Leary

Background

Mick Cooney has been a personal friend for almost 20 years. Mick holds a PhD in Computational Stochastic Finance from Trinity College Dublin and he is currently a Quantitative Analyst working primarily in data science roles in the insurance industry. Mick is also one of the organisers of Dublin R and runs regular workshops on more advanced topics like Bayesian data analysis, time series models, bootstrapping and Gaussian processes.

Jamie O'Leary is a Senior Software Engineer for Informatica Ireland, a company that specialises in data integration. He currently works on semantic analysis of data sets and automated categorization and identification of large volume data.

JR: Why R? or Alternatives?

MC: This might be controversial, but I think there are only two real choices for data analysis: R or Python. I learned R first and always loved Lisp, and R syntax was based on Scheme, a version of Lisp. In particular, there are a series of truly excellent packages in R that are indispensable to me: ggplot2, data.table and dplyr. It also tends to be the statistician's language of choice. I will admit that is subjective for me though. Python is also excellent and Julia is a language to watch for.

Open source tools are the only choice for me. You can use any library available, available features tend to be cutting edge, and do not underestimate never having to pay licensing fees. Finally, new techniques, algorithms and optimisations are mainly implemented first via open source.

JO'L: I primarily work in Java, it's the most common language used in industry because of the wide variety of tooling available. Other languages, like R, are certainly better for visualization and analysis of data, but my job involves producing scalable customer solutions. This inevitably means large frameworks and supporting libraries to handle everything a customer needs from logging and integrated support to high availability and scaling. Java has these, and in many different varieties.

JR: This project seeks to find information from Big Data sets composed primarily of categorical data, what in your opinion would be a successful outcome from this project?

MC: There is a lot of over-selling going on concerning the capabilities of data science, statistical modelling and AI at the moment, so realistic goals are paramount. Working with unstructured data is not trivial, and working with unlabelled data compounds this.

A system that can filter out a large proportion of the data and return events worthy of further investigation would be an excellent result. This may sound conservative, but achieving such a system involves solving a number of challenging issues. Successful projects can always be improved and iterated upon later.

JO'L: The challenge you have set yourself here is a good one. You could set your goals at a few different levels and each one achieved would be a good success story. If you can point out areas of a log file worth paying attention to, you are already saving someone time. If you can point out single lines that are 50% likely to be errors, you are narrowing the search even more. If you get to a point where you can actually understand the sentiment of the message in a log and be 80% certain it's an issue worth addressing, then you have something pretty amazing.

It's a question of how much work you can take away from the user as far as I can see. Any reduction in the user's effort to determine if a problem occurred is a win.

JR: Considering the subject matter of this project, are there any easily avoided pitfalls waiting for me in the data?

MC: Most of the pitfalls in a project like this will be at the start. Successful projects often hinge upon a good choice of tools and approaches, and preparing to quickly drop what is not working and trying something else.

I would also not underestimate the amount of time you will spend doing the boring but necessary work of cleaning and preparing the data. I would expect the large majority of your time will be spent on that.

It is also wise to have a sense for how you will evaluate your work, but be prepared to shift your thinking if the situation warrants it: the more you work on the data and the project, the more you know about it and can properly decide what is working and what is not. Failure is an important learning tool. Do not shy away from it.

JO'L: The data itself is always the worse part of data analysis. It arrives unclean, in unspecified formats, full of errors that can skew analysis in so many unpredictable ways. There is also the age old issue of biting off more than you can chew. I mean this in terms

of the size of data you might think you can crunch through in a reasonably sane amount of time.

JR: What are the most important techniques, processes, or methods that I should understand in order to succeed in this project?

MC: Learning how to process, transform and work with data. This is something you do over and over and over again regardless of the project. Efficient methods for doing this are invaluable.

Secondly - do not fall in love with the fancy. There is a lot of buzz around all sorts of artificial intelligence approaches and such, but I think my mathematical background has biased me towards appreciating simple and elegant approaches. There is nothing wrong with solving a problem with a simple technique implemented well. To me, that is much better than doing something sophisticated and throwing a huge amount of computing power at a problem.

Not over-complicating things often seems unsatisfying, but do not try to keep up with the Jones's.

JO'L: Efficient data structures and data normalisation. You are dealing with large bodies of text, and you will want to analyse and process as much of it as you can to figure out what it means, line by line, as a group, or as a whole. Knowing how to reduce the amount of data you need to work on, and knowing when it is a good idea not to could be very important.

JR: How important is human intuition when approaching unstructured data science problems such as this?

MC: Massively so. The nature of the problem means that gauging success will depend entirely on your intuition about the output. Most clustering-style approaches require a large amount of human explanation.

Another huge issue with all data modelling is the temptation to quickly add narrative to the output you see. Of course, this is the ultimate goal of your task - the whole point of data analysis is to obtain an explanation for your data - but be wary of all interpretation. It is often tempting to rush this part as outputs and explanations seem 'obvious', but I urge caution. Too often I have misled myself.

JO'L: Humans are incredibly powerful problem solvers. The problem is that we can't entirely replicate the processes people go through as sometimes they evolve as they are

employed. Software can't be written to be creative at that level yet. I generally think about how I would work out problems; then I see if it can be replicated in software. It's still surprising to me that I can come up with ways to solve a problem that you just can't write software to do. In general, though, some of the process can be, and that can get you close to a good solution.

JR: Simply getting my data into cleaned for analysis has been a considerable task; would you say that this is a common experience in the field of Data Science?

MC: Definitely. Myself and my two colleagues hold strongly to the old adage about time being spent: 80% of our time is spent on data cleaning, 15% is explaining the work, and 5% is spent on modelling. Every person I met who works in this field has agreed. Having said all that, this is something I would be hugely wary of changing. By going through the sometimes-tedious process of working with the data, asking questions, doing some aggregations of levels and so on, you learn a huge amount about it.

Data cleaning is a task that is often seen as something that junior analysts do, freeing up the time of more senior team members. In the last year or so I have started to disagree with this more and more strongly. Yes, it is tedious, but there is no other way to become familiar with the data. I have worked on a few projects where I was not part of the initial explorations and I never got fully comfortable with it as a result.

As a consequence, I had a lot of unanswered questions and ambiguities. In future, I think any and all new people need to spend some time on those initial explorations as part of getting up to speed on a project. There are no shortcuts for that.

JO'L: Oh yes, entirely. Analyse your data set for mistakes. Clean it. Analyse it again for mistakes. Clean it again. You might think this is a joke, but really I've seen it. Spikes in graphs caused by completely invalid data, interesting stats like "Our product is most popular in N/A" (but that actually isn't North America).

JR: Which is more important, the needle or the haystack? (noise and signal, is the existence of noise and its characteristics significant?)

MC: That is quite a philosophical question! In a lot of cases, you have data and there is a clearly defined concept of what the signal is. Everything else is noise. In other cases, the definition of the signal is not clear, and part of the task is to determine what is there.

As a result, I am usually wary of making definitive statements about anything. This can be annoying - Harry Truman famously once asked if he could have a meeting with a

‘one-handed economist’ - but making stronger statements is almost always naive at best and dishonest at worst.

So, I often find that while the signal is the most important, the noise is often interesting as well.

JO’L: This can depend on the problem domain. If you are only looking for a needle, then it doesn’t matter so much. If you are looking for a needle, but you think there might be some other things in there, then it’s worth analysing things you wouldn’t normally look at. There might be gold in that haystack. For your log analysis, it might be worth knowing that error messages always show up after a repeating access to a resource or repeated actions by a user. This could point to errors caused be load in one particular area.

JR: What does a data scientist need the most?

MC: Curiosity first and foremost. All the people who impress in this field share that quality. They are constantly seeking to learn. Smart and curious is an excellent combination of traits.

On top of that, a healthy scepticism is crucial. At current levels of knowledge, humans are much more valuable and scarce than computing power. There may come a time when machines have automated a large part of this field, but I am still quite sceptical of that - AlphaGo notwithstanding. Algorithms are just a tool to help us understand. They are not standalone and I think we are still a time away from when they are.

As a result of this, it is important to say “the model is wrong” and ignore it. I find fully automated systems a little scary for that, especially if those systems are mission-critical to a business, organisation or enterprise. I would immediately look for ways to bring humans back into the loop in some way.

My concern would be that the more you automate and systematise, the faster and more spectacular the explosion when things go wrong. You need a senior person to have the authority to turn off or ignore the machine if she thinks it is warranted.

JR: What are your top 3 predictions for the next 20 years in Data Science?

MC: Wow! That’s a tough one. We as a race have a long history of being thoroughly wrong about the advances of technology on both sides. In the 1960s the future was all

about flying cars and living on the moon, but people talked via monitor phones with handsets.

The growth of raw computation power has flattened a little in the last few years, and parallel architectures are harder to exploit, but I still think our devices will grow in power over the next few decades.

Mesh networks are an interesting concept: they work on the idea that all the devices that are network enabled also become part of the network itself, so there is much less need for ISPs and wireless network providers. We will see on that front, but I could see some of those ideas being adopted. Such things will have a huge effect on data analysis as it allows the use of more and more computational power to be deployed.

JR: Why does every good data scientist I speak to have that same reaction of "EEK, REALLY? "when I mention outliers in "Categorical" Data?

MC: Interesting. I imagine it is because outlier detection (and clustering in general) is mainly associated with working on numerical data. The basic idea of both is that you define a 'distance' between data, and that concept is less obvious when using categorical data? To give a simple example, how do you define the distance between gender "male" and gender "female" for example? Is such a distance similar to that between "apple" and "orange" for favourite fruit?

Another big issue is that with categorical data, the variables may not contain a huge amount of different values, and so clustering the data splits it up into the various combinations of categorical variables. This is typically less interesting.

That said, I think there are a surprising number of situations, such as your project, where it is valid to start asking questions like yours - it just is not obvious how to go about it.

It is also possible that you have a large number of categorical variables so that even if they individually do not possess a lot of different values the combination of them is huge and so this makes it more numerical-like in its behaviour.

This was why I told you at the very start that your problem was both interesting, but non-trivial.

JO'L: I wonder. Categorical data is usually bound to a specific set of values or within a specific problem domain, so you should be able to narrow what you are looking for with some kind of ontology. With the right reference data for what you should be getting in the categorical data set, it shouldn't be too hard to highlight what isn't. I guess, the scary parts to this are that problem domains are rarely well defined, and finding a definitive

ontology can be a lot of work in itself. Put simply: If you can tell me all the good things to see in an apache log, I can tell you all the bad things I find really easily.

ii. Curse of dimensionality code

```
library(pbapply)
Niter <- 1000000
Ndim <- 1:25
vol_frac <- pbsapply(Ndim, function(iterdim) {
   hits <- replicate(Niter, {
      space_samp <- runif(iterdim, 0, 1)
      sum(space_samp^2) < 1
   })
   sum(hits) / Niter
})
frac_plot <- ggplot() +
   geom_line(aes(x = Ndim, y = vol_frac)) +
   xlab("Dimensions") +
   ylab("Fraction of Volume") +
   ggtitle("Fraction of Volume Occupied by Hypersphere in Hypercube vs Dimensionality of Space")
ggsave(frac_plot, file = "volfrac_plot.pn", height = 10, width = 14)
```

iii. Requirements Elicitation Interview

This interview John BT O'Connor of IBM Ireland conducted by John Rogers 16/12/15 is to further explore the functional and non-functional requirements of a final year undergraduate project with a subject matter of log file analysis, visualization and outlier detection. Any information included in this draft document could be subject to NDA and must be checked with IBM before disclosure in any project related reports.

Also in attendance: Brendan Arthurs (Architect - Connections Cloud Chat), Shaun McGale (Manager - Connections Cloud Chat)

JR: Why? (NB. Driver analysis question): What factors beyond it being an academic work, have caused this project to come into being?

JO'C: John's final year electives of software systems and AI have an obvious relevance to a common issue in cloud computing. We see the opportunity to help John, academia and ourselves. "Symtriosis".

The use case we are targeting is: Can we make log analysis easier for the 24x7 ops teams? It is in our interests for help the ops teams in order to reduce the amount of calls made to technical support and dev teams. Let's make the logs smaller by producing a filtered view through John's system. Reduce the size of the haystack rather than try discover the needle.

JR: What would be considered a successful project from a non-academic standpoint?

JO'C: Any of the following:

1. Is there value in following this approach?
   Can we reduce the haystack accurately without missing abnormal events.
   A "No" answer could be considered a success if we can show that a high risk of losing the markers for abnormal events for any individual strategy.
   The report could point out "Strategy X is more dangerous because of its risk of destroying good evidence is relatively high"?
2. What level of haystack reduction can we achieve?
3. How much work is there in defining patterns of normal behaviour
4. Can a simple data training strategy reduce the size of the haystack?

JR: How will the end result of the project be used?

JO'C: By ops people when they have already identified a problem and want to use this system as a tool to assist analysis.

JR: How is the raw data currently collected?

JO'C: Generated by the system and stored on databases. To be collated by John's system. Test and training data will come from our product test installations.

JR: How might we think about this application differently? (Are we re-inventing the wheel?)

JO'C: Log Entries and New Relic are tools we currently use. But this takes a different angle. Based on specific training and patterns what data can we consider as noise.

JR: When would this application be used?

JO'C: When ops specialist is analysing an issue and looking for the smoking guns.

JR: When would the application be considered detrimental to existing processes?

JO'C: None – if we assume the ops people can invoke it of their own volition.

JR: Who are the end users?

JO'C: Ops Specialists

JR: Who controls the raw input data currently?

JO'C: IBM Connections Chat team

JR: Who receives the outputs of any automated functions?

JO'C: The end user – when they invoke the system. Not running in the background.

JR: In software engineering (and systems engineering), a functional requirement defines a function of a system and its components. A function is described as a set of inputs, the behaviour, and outputs.

JO'C: End user can take a set of logs and view a filter of that data.

There can be many filtered views available to a user. The details of those requirements are TBD

JR: In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours.

JO'C: Able to be run in real time

Bibliography

Aggarwal, C. C., 2013. *Outlier Analysis.* 1 ed. New York: Springer.

Aggarwal, C. C. & Yu, P. S., 2001. *Outlier Detection for High Dimensional Data.* s.l., s.n.

Akoglu, L., Tong, H., Vreeken, J. & Faloutsos, C., 2012. *Fast and Reliable Anomaly Detection in Categorical Data,* New York: StonyBrook Edu.

Anon., 2016. *Home -RDatable/data.table.wiki.* [Online]
Available at: https://github.com/Rdatatable/data.table/wiki
[Accessed 7 May 2016].

Arning, A., Agrawal, R. & Raghavan, P., 1996. *A linear method for deviation detection in large databases.* s.l., s.n.

Assorted, 2016. *R-Bloggers.* [Online]
Available at: http://www.r-bloggers.com/

Beyer, K., Goldstein, J., Ramakrishnan, R. & Shaft, U., 1999. When Is "Nearest Neighbor" Meaningful?. In: *Lecture Notes in Computer Science.* New York: Springer, pp. 217-235.

Biggs, N., 1993. *Algebraic Graph Theory.* 2nd ed. Cambridge: Cambridge University Press.

Boriah, S., Chandola, V. & Kumar, V., 2008. *Similarity Measures for Categorical Data: A Comparative Evaluation, SIAM Conference on Data Mining.* s.l., University of Minnesota.

Breunig, M. M., Kriegel, H.-P., Ng, R. T. & Sander, J., 2000. *LOF: Identifying Density-Based Local Outliers.* Dallas, s.n.

Card, S. K., Mackinlay, J. D. & Shneiderman, B., 1999. *Readings in Information Visualization: Using Vision to Think.* s.l.:Morgan Kaufmann.

Chambers, J., 2008. *Software for Data Analysis.* 1 ed. New York: Springer.

Chaudhary, A., Szalay, A. S. & Moore, A. W., 2002. *Very Fast Outlier Detection in Large Multidimensional Data Sets.* s.l., s.n.

Cooney, M. & O'Leary, J., 2016. *Interview with topic experts for final year project* [Interview] (January 2016).

Das, K. & Schneider, J., 2007. *Detecting anomalous records in categorical datasets.* New York, ACM.

Friendly, M., 2006. *Handbook of Computational Statistics: Data Visualization.* Toronto: York University.

Fruchterman, T. M. J. & Reingold, E. M., 1991. Graph Drawing by Force-Directed Placement. In: *Software – Practice & Experience (Wiley).* s.l.:s.n., pp. 1129-1164.

Ghoting, A., Parthasarathy, S. & Otey, M. E., 2008. Fast Mining of Distance-Based Outliers in High-Dimensional Datasets. In: *Data Mining, Knowledge and Discovery.* Ohio: Ohio State University, pp. 608-612.

Hawkins, D., 1980. *Identification of Outliers.* s.l.:Chapman and Hall.

Houle, M. E., Kriegel, H.-P., Kr¨oger, P. & Schubert, E., 2010. *Can Shared-Neighbor Distances Defeat the Curse of Dimensionality?.* Heidelbreg, s.n.

Kamada, T. & Kawai, S., 1989. An algorithm for drawing general undirected graphs. In: *Information Processing Letters (Elsevier).* s.l.:s.n., pp. 7-15.

Kobourov, S. G., 2012. *Spring Embedders and Force Directed Graph Drawing Algorithms,* s.l.: University of Arizona.

Lantz, B., 2013. *Machine learning with R.* Birmingham: PACKT.

Larman, C. & Basili, V. R., 2003. Iterative and Incremental Development: A Brief History. *IEEE Computer,* 36(6), pp. 47-56.

Lindquist, E., 2011. *Grappling with Complex Policy Challenges: exploring the potential of visualization for,* Canberra: Crawford School of Economics and Government, ANU.

Liu, S. et al., 2015. *Visualizing High-Dimensional Data: Advances in the Past Decade.* s.l., Eurographics Association.

O'Connor, J., 2015. *Elicitation interview* [Interview] (18 12 2015).

O'Connor, J. B., 2015. *Funtional requirements gathering elicititation interview* [Interview] (November 2015).

O'Connor, J. B., 2016. *Client Feedback E-mail to NCIRL (Simon Caton).* Dublin: NCIRL.

Parsons, T., 2015. *Extracting Value from Log files.* [Online]
Available at: https://blog.logentries.com/2015/01/extracting-key-values-from-any-log-format-using-regex/

Press, G., 2013. *A very short history of Data Science.* [Online]
Available at: http://www.forbes.com/sites/gilpress/2013/05/28/a-very-short-history-of-data-science/#43fd518569fd
[Accessed 4 May 2016].

R_Foundation, 2016. *R:What is R?.* [Online]
Available at: https://www.r-project.org/about.html
[Accessed 2016].

Silva, T. & Zhao, L., 2016. *Machine Learning in Complex Networks.* New York: Springer.

Weber, W., 2013. *Lecture 2: The term document matrix.* [Online]
Available at:
http://www.williamwebber.com/research/teaching/comp90042/2014s1/lect/l02.pdf
[Accessed 7 May 2013].