



Market Basket Analysis with Shortened Web
Link Click Data

THESIS

James C. Gallagher, CPT, USA
AFIT-ENS-MS-18-M-123

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Army, the United States Department of Defense or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENS-MS-18-M-123

MARKET BASKET ANALYSIS WITH SHORTENED WEB LINK CLICK DATA

THESIS

Presented to the Faculty
Department of Operational Sciences
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Operations Research

James C. Gallagher, B.S.
CPT, USA

22 March 2018

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT-ENS-MS-18-M-123

MARKET BASKET ANALYSIS WITH SHORTENED WEB LINK CLICK DATA

THESIS

James C. Gallagher, B.S.
CPT, USA

Committee Membership:

Dr. C. M. Smith, LTC
Chair

Dr. B. C. Boehmke
Member

Abstract

Market research is an indispensable part of an organization's ability to understand market dynamics in an area. Over the past 20 years data collection and analysis through Knowledge Discovery through Databases (KDD) has arisen to supplement the traditional methods of surveys and focus groups. Market Basket Analysis is an area of KDD that identifies associations between commonly purchased items. As social media use has grown, link shortening companies help users share links in a constrained space environment and, in exchange, collect data about each user when a link is clicked. This research applies market basket analysis techniques with graph mining to shortened web link data to identify communities of co-visited websites to help analysts better understand web traffic for an area during a time range. Patterns within clusters of web domains regarding hardware platforms, operating systems, or referral sources are then identified and used to gain a better understanding of an area.

*To my Wife and Daughters who supported and cared for me better than I could have
ever hoped*

And to my Friends and Colleagues whose humor and optimism helped immensely

Acknowledgements

I am grateful for the assistance, guidance, and mentoring provided by my faculty research advisor, Dr. Christopher Smith (LTC). His assistance was invaluable throughout this entire process.

James C. Gallagher

Table of Contents

	Page
Abstract	iv
Acknowledgements	vi
List of Figures	ix
List of Tables	x
I. Introduction	1
1.1 Background	1
1.2 Problem Statement	3
1.3 Approach	3
1.4 Assumptions and Constraints	4
1.5 Summary	5
II. Literature Review	6
2.1 Market Research Background	6
2.1.1 Market Research Methods prior to the Internet	6
2.1.2 Traditional Market Research Conducted Online	7
2.2 Knowledge Discovery through Data (KDD)	9
2.2.1 Social Media KDD	10
2.3 Market Basket Analysis	11
2.3.1 Market Basket Analysis through Graph Mining	12
2.4 Summary	12
III. Methodology	14
3.1 Overview	14
3.2 Data Set	14
3.3 Co-Purchase Graphs	14
3.3.1 Build a bi-partite click graph	15
3.4 Community Detection Algorithms	16
3.4.1 Speaker-Listener Label Propagation Algorithm	17
3.4.2 WalkTrap Algorithm	18
3.4.3 Filtering Techniques	21
3.5 Community Detection Toy Example Application	22
3.5.1 Amazon Book Co-Purchase Network Toy Example	22
3.5.2 Implementation of Community Detection Algorithms	23
3.5.3 Results & Algorithm Selection	25

	Page
IV. Results & Analysis	27
4.1 Introduction	27
4.2 Exploratory Data Analysis.....	28
4.3 Community Detection.....	35
4.4 Community Measures	39
4.4.1 January 20-22, 2017	40
4.4.2 February 5-7, 2017.....	44
V. Conclusion	48
5.1 Summary	48
5.2 Conclusions.....	49
5.3 Future Research	50
Appendix A. Web Click Example Data.....	52
Appendix B. Software Packages Used	53
Appendix C. Analysis Code	54
Bibliography	79

List of Figures

Figure	Page
1	Amazon Co-Purchase Ground Truth Network 23
2	SLPA Community Detection results 24
3	Walktrap Community Detection results 25
4	Methodology 28
5	Access Methods by Hardware Platform 30
6	Access Methods by Operating System 31
7	Access Methods by Referrer Type 32
8	Link Click-Time Distributions 33
9	Link Creation-to-Click Times 34
10	Inauguration-related Clicks by access type 35
11	Super Bowl-related Clicks by access type 36
12	Community Detection Measures related to Filter size (January 2017) 37
13	Community Detection Measures related to Filter size (February 2017) 38
14	Number of Domains per Identified Communities 39
15	Community Information Measures (January 2017) 41
16	Access Types by Community (January 2017) 42
17	Distribution of Click Times by Community (January 2017) 43
18	Community Information Measures (February 2017) 45
19	Access Types by Community (February 2017) 46
20	Distribution of Click Times by Community (February 2017) 47

List of Tables

Table		Page
1	Transaction Data Example [1]	11
2	Toy Example Classification Accuracy Results	25
3	Remaining Web Clicks following Data Cleaning	29
4	Click-, Unique Domain-, and User-Counts	35
5	Domain and Web-Clicks for January 2017	39
6	Domain and Web Clicks for February 2017	40
7	Web Click Example Data	52
8	Software Packages Used	53

I. Introduction

1.1 Background

Market research is an important endeavor for many companies. Market research is geared towards identifying markets and customers in an attempt to tailor advertisements and products to interested customers. Prior to the growth of information systems, companies relied on surveys and focus groups to better understand potential markets. These methods allowed researchers to interact with customers in controlled environments to learn about new products, try to improve existing products, and how to better position products in existing or new markets.

In the past 20 years, one focus area for companies has been the collection of transaction data. Stores collect data on which products are purchased together and then use this data to determine its more popular products, identify any related seasonality, or compare store-to-store purchases. This field of research, called Market Basket Analysis, helps analysts better understand purchase patterns. For example, grocery stores use market basket analysis to determine which products should be placed together (e.g. milk and bread) to ensure product co-purchase. Additionally, retailers use this information to determine sales prices. If two products are commonly purchased together, then a retailer does not need a sale price for both items as they will likely be bought together regardless.

In addition to identifying useful relationships between tangible products, market basket analysis is a technique used to find associations in other domains. This research

applies market basket analysis techniques to shortened web link data. By clustering web domains by common users, interesting patterns arise providing insight to analysts regarding common internet use in an area during a specified timeframe.

Internet use has grown tremendously over the past 20 years and is a growing area for market research. In December 2016, Facebook boasted over 1.86 billion users [2] and marketers have been trying to determine how best to include these platforms into marketing plans for products. The communities on the social media platforms represent a large market for companies trying to buy and sell both products and information. However, as social media marketing activities have increased, a company's ability to make informed decisions about the effectiveness of these activities still has not improved very much [2]. Companies are necessarily interested in determining which marketing and information dissemination methods are most effective and attempt to maximize these efforts to ensure that pertinent information is readily available to all who wish to access it.

One aspect of social media websites is a need for condensed content. Twitter, for example, limits users to 280 characters for each sent tweet. The typical link to an article or advertisement is approximately 130 characters and estimates show that as many as 31% of tweets containing links would have exceeded the character limit without shortening the original link[3]. This has led to the rise of link-shortening services that redirect users to important pages. In exchange for providing a shortened link, the link-shortening companies collect information on users each time a link is clicked. The information includes how many times a link has been clicked, the referrer application (e.g. Twitter or Facebook), and the user's location [3] while saving users an average of 91% of space on their post [4]. Shortened link companies host this information and market basket analysis should help provide insight into interesting patterns in internet use for an area.

1.2 Problem Statement

Bitly and Google host a large amount of data from web clicks publicly available on their respective websites. Given similar link-shortened data, we wish to explore the network between visited sites/message types and message delivery methods (i.e. Facebook, Twitter, etc.). This research transforms text-based data and finds meaningful relationships within the data. Graphic visualizations are geared towards helping an analyst build a better understanding the information environment of a location. This method could help organizations refine and better understand information access and dissemination patterns within a specified area and time.

1.3 Approach

This research uses the transactional link-click data to identify useful relationships between web domains and users. After cleaning the data, a bipartite transaction graph is built that connects unique users to web domains. The web domains (e.g. www.cnn.com) host the the end location that the user is attempting to visit. The bipartite graph is then transformed into a co-purchase graph with nodes connected by common users. Graph-mining techniques, specifically community detection, are then applied to identify clusters of common co-accessed web domains. Upon evaluating these clusters, patterns of use emerge that will help analysts better understand the information access patterns of a geographic location. For example, within communities, the hardware platforms, operating systems, or referrer platforms differ which provides insight into how that information could be better delivered.

1.4 Assumptions and Constraints

Companies, when hosting information (i.e. articles) are interested in maximizing the number of people who view and interact with information they host online. Even though these transactions do not involve the exchange of currency as traditional purchasing does, the information contained at the end location can be considered the product to be sold, therefore that item is sold when the link is clicked. This association is assumed to extend to the entire web domain.

Additionally, on the user end, users treat the links as products to be sold. Despite there generally being no monetary cost associated with clicking a link, users will click on links in line with their interests and likes often to maximize their use of time online. So when multiple links are clicked by a user, the common thread between those links can be interest by the user. This commonality extends beyond just individual links as well. When a user clicks on a link, the user is assumed to be aware of the end location and incorporates that into the decision of whether to click or not on the link. Then, ultimately, the community detection methods discussed are focused on discovering the hidden connections between information sites themselves and not just the individual article or blog post.

Next, shortened link use is similar to general internet use. As shortened links have become more common, the links have been accepted as useful methods for sharing links especially through social media. For example, from a one week sample of 20M tweets containing hyperlinks, 50% of those links had been shortened by bit.ly while only 13% contained full length links [3]. Shortened links are a commonly used method for distributing web links across the internet especially over social media and can be useful in better understanding overall internet use in a particular area.

Finally, because this research is intended to be transformed into a useful portable tool, the focus of the analysis will be limited in scope to one city at a time. Despite this

limitation, the data sets will still be large (approximately 2Gb or 1M Observations) which limit investigation to nearly linear time community detection methods.

1.5 Summary

To demonstrate the market basket research techniques, analysis was performed on two case studies in Charlotte, North Carolina. The case studies investigated shortened web link use around the 2017 Presidential Inauguration (Friday, 20 January 2017-Sunday, 22 January 2017) and the 2017 Super Bowl (Sunday, 5 February 2017-Tuesday, 7 February 2017). The analysis indicated that the plurality of shortened web links clicked during both weekends came from a smartphone device and originated from a Facebook-hosted link.

After building a co-purchase graph and filtering low-incidence clicks, community detection was conducted on the data to identify clusters of commonly accessed web domains. For each weekend, two communities were identified that demonstrated differences in access patterns between each community in terms of information accessed (i.e. article topics) and the methods used to access the information (e.g. device type or referral source). These differences help an analyst better understand how and by what information is being accessed and can use the information in building advertising campaigns or optimizing products and services (e.g. apps or web sites) to match these trends.

II. Literature Review

2.1 Market Research Background

Market Research [5] is the "systematic and organized gathering, analysis, and presentation of information... for strategic and product planning". Market research can be an important endeavor for companies as they try to expand their market share. Market research has a number of important endeavors for businesses including informing decision makers' goals, identifying new markets, or identifying new technologies[5]. Traditionally, market researchers have conducted this research using two methods: market surveys[6] and focus groups[7].

2.1.1 Market Research Methods prior to the Internet

Since historic times, surveys have been used by governments and companies to gather information about their citizens or customers. For example, the United States Constitution mandates the federal government conduct a census every 10 years to gather basic information about its citizens. The data collected in these surveys, either governmental censuses or business research, focused mainly on the collection of objective data such as age, ethnicity, and living information. In the mid-20th century, businesses adapted and modified surveys to identify "what people knew, felt, and thought"[6, p. 4]. These surveys collected qualitative information with the aim of identifying information such as whether customers may purchase certain products. These surveys have been conducted on small scales, such as a local politicians trying to identify which policies are most supported by a city or town. Surveys are also used in large scale applications like the Survey of Consumers, funded by the United States Government and major businesses, which measures consumer confidence in the American economy. Large government agencies such as the US Census Bureau and

Bureau of Labor Statistics rely on surveys in the completion of their duty. While no clear research has been conducted to determine how many private industries use survey data, some researchers estimate total use to exceed government many times over[6].

While surveys were useful, and are still used today, another method, called a focus group, rose to augment the shortcomings created with surveys. A focus group is a curated group of approximately 10 individuals with no prior relationships or interactions. Researchers recruit participants that fit a set of desired characteristics like occupation (e.g. doctors) or product user (e.g. cell phone owners) but are varied in others characteristics (e.g. age or ethnicity). The individuals would meet with market researchers and discuss topics in a loosely guided open format[7]. One benefit of this method is the ability for respondents to cover topics outside of the restrictive survey. For example, in the 1980s using focus groups, beverage companies identified that in addition to a consumer's thirst, the brand's social status and packaging designs influenced consumer's soda consumption[7]. As a result, the companies transitioned and expanded their advertising campaigns to encompass these new areas.

2.1.2 Traditional Market Research Conducted Online

As the internet has become more prevalent and accessible, researchers have transitioned to collecting market information online. Between 2006 and 2015, the proportion of research conducted online increased from 40% to 54% [8], with a corresponding decrease in the proportion of paper surveys from 21% to 7%. Additionally, advertisers spent more than \$101.5 million on online advertising in 2013[9] as opposed to 2012. The online spending constituted a 14% increase in spending in 2013 relative to a 4% increase in traditional media advertising spending. One benefit of online advertisements is the ability to collect data about its users without having to conduct surveys

or other previously accepted market research methods[10]. As consumers moved online, the amount of available data grew exponentially. Additionally, this data captured what users actually did as opposed to relying on users telling researchers what they did [11].

Traditionally, advertisers have used the estimated number of impressions, i.e. the number of people who will see the advertisement, to measure the effectiveness of ads. Advertisements that will be seen by many people are considered more effective and valuable than other less visible ads. However, for online ads, click through rate has become the most widely used metrics. Click through rate is the proportion of ad viewers who click on, and are redirected to, the advertiser's desired website[12]. For example, a web advertisement with 100 page views and five clicks on the advertisement has a calculated click through rate of 5%. Baltas[11] developed a model to predict an online advertisement's click through rate based on a number of aesthetic design elements. Lin and Chen expanded on this model to include animated advertisements to help advertisers maximize click through rate. Another model for marketing effectiveness[13] evaluated a company's advertising budget, average pages viewed, average web site quality, and other variables to predict the click-through rate of an advertising campaigns. Researchers were able to use this data to learn about their target market. For example, Ilfeld and Winer [13] conclude that unlike a brick-and-mortar store with products visible on shelves, many of which are unknown to the consumer prior to entering the store, online users are limited to products to which they are specifically exposed whether by advertisement or a web link. Therefore, because of this limitation, companies should focus investment in creating greater exposure and inducing the user to visit the website as opposed to building a brand image or brand loyalty with their online advertisements. These examples demonstrate how traditional market research methods transitioned online but ultimately use the same

techniques to learn about their target audience. However, as internet use continued to grow, researchers discovered that tremendous amounts of non-traditional data was being created and stored in various databases.

2.2 Knowledge Discovery through Data (KDD)

As information technology systems have evolved, the amount of information and data available has increased at record rates[14]. To deal with the increased information, Knowledge Discovery through Data[1, 15] has been applied to the problem. KDD is a systematic process to glean useful information from the massive amounts of data held through various databases around the world. Medicine, bio-informatics, industrial processes, computer security, and, of course, marketing are a number of areas in which KDD and data-mining has been applied to learn new patterns of behavior [16]. There are five main techniques in KDD:[17, 1, 18]: Association, Classification, Clustering, Prediction, and Outlier Analysis models. Kaur and Kang [1] defines association modeling as discovering a relationship between items in a market. This type of association modeling is often applied to marketing problems in the form of market basket analysis or cross-selling programs [1]. While there are a number of ways to apply KDD, typically they follow four general steps. Raval[17] outlines the steps as follows. First, researchers and analysts must identify a business purpose for the data and then select data that will help accomplish the business objective. This is an important step as data used to conduct association modeling might be different than the type needed to conduct a cluster model or prediction model. Next, the data is pre-processed or cleaned and prepared for model fitting. This step entails taking previously unprocessed data and transforming it into useful data. Some necessary data decisions are made in this step such as what to do with missing data or manipulating the data into a more useful format (e.g. normalizing numerical data). After

the data is cleaned, the following step is to apply the algorithms and fit the models to the data. This step allows for the output of useful patterns and conclusions. Finally, those conclusions and rules must be synthesized and interpreted in a way to provide a better understanding of activity and behavior.

2.2.1 Social Media KDD

Social Media data mining provides avenues to explore markets and understand locations outside of traditional market research methods and “can yield interesting perspectives on human behavior” [19, p. 327]. According to Kaplan and Heinlein [20], “Social Media is a group of internet-based applications that build on the ideological and technological foundations of Web 2.0, and that allow the creation and exchange of User Generated Content”. In 2009 [20], for example, every minute 10 hours of video was being uploaded to Youtube®. As the social media users increase use, companies are looking to tap into the wealth of information stored there [21][9] to help their company brands to expand market share or “go viral”.

2.2.1.1 Shortened Click Data

Social networking sites have a need for condensed content, whether in file size or actual post length. Twitter®, for example, limits users to 280 characters for each tweet that is sent. Constraints like this led to the rise of link-shortening services as users looked to share links to content but in a manageable number of characters to ensure context could be provided to the links. In fact, users save an average of 91% in their post’s character size [4]. In exchange for providing this service, web link shortener companies collect information about users each time a link is clicked, including how many times a link has been clicked, the referrer application (e.g. Twitter or Facebook), and the user’s location. Antoniadou, et al. [3] found that Twitter and

Facebook were two of the top five most common referrer applications and referred users most often to news websites (25% of all links), followed by entertainment, personal, and commercial websites and others. In addition to the end location of links, Antoniadou, et al.[3] found that in a one-week period over 20M tweets contained a hyperlink of some sort. Of these 20M tweets, 50% had been a shortened link from bit.ly while only 13% were unshortened links of any kind. This implies the value of shortened links as a common and accepted method for sharing information over the internet.

2.3 Market Basket Analysis

A common application of association modeling from KDD and data mining is a technique known as market basket analysis. Solnet, et al. [22] discuss how consumers rarely make isolated purchasing decisions. Since purchasing decisions are often made a basket at a time, companies are interested in determining which products, regardless of category, are often put in the basket together. This is useful to retailers as they try to combine products or use sale prices to drive the purchase of multiple products simultaneously. Market Basket Analysis was first proposed by Agrawal and Srikant [23] in 1994 as a database mining technique using association rules. Table 1 shows an example transaction log.

Table 1. Transaction Data Example [1]

Transaction	Items
1	Butter, Milk, Cheese
2	Butter, Milk, Bread
3	Butter, Milk

In this simplified example, an identified association rule is that milk and butter are bought simultaneously. This information may be valuable to companies who will try to ensure its co-location or understand that market changes to butter will affect milk

and vice versa. The association rule is not a causative relationship as purchasing milk does not necessarily cause the consumer to purchase butter. However, there is value in a retailer understanding that when milk is purchased, often butter is purchased with it. Due to the computational complexity of association rule mining, Kaur [24] discusses the evolution of association rule mining algorithms. In addition to the problem of computational complexity, Kaur[24] identified three problems with current association rule mining algorithms: uninteresting results, complexity (i.e. number of rules discovered), and low algorithm performance. Videla-Cavieres and Rios[25] then demonstrated that by using graph mining techniques in market basket analysis could mitigate the problems identified by Kaur.

2.3.1 Market Basket Analysis through Graph Mining

After applying traditional association rule mining to a transaction dataset and obtaining uninteresting results, Videla-Cavieres and Rios[25] used graph mining techniques on the dataset. Graph mining is defined as “the extraction of novel and useful knowledge from a graph representation of data” [26, p. 2]. Specifically, Videla-Cavieres and Rios[25] transformed a transactional database into a co-purchase graph. The co-purchase graph then connected nodes with similar co-purchasers. From this graph, Videla-Cavieres and Rios then applied community detection to the co-purchase graph to identify common items. The outputs were more interesting and beneficial than using association rule mining.

2.4 Summary

Market Research has evolved from its inception. Initially focused on fact-based information gathering, research evolved into trying to better understand consumers and their thoughts and feelings related to products and brands. As technology evolved, the

fundamentals basically stayed the same. This changed with the advent of information systems capable of maintaining vast stores of various types of data. New market basket analysis algorithms were then developed and applied to this newly collected data each with their own advantages and disadvantages. Eventually graph mining techniques for market basket analysis showed that large datasets could be mined efficiently with more interesting results than previously used association rule mining algorithms. these graph mining techniques are discussed in Chapter III.

III. Methodology

3.1 Overview

A common method for understanding and visualizing social network structures is to apply graph theory fundamentals to the network [27]. This chapter discusses the building of the bipartite link-click graph, transforming this graph into a co-purchase graph, and then highlights the community detection methods chosen for analysis. Finally, these techniques are applied to a toy example to select an appropriate community detection method.

3.2 Data Set

When a user clicks on a shortened link, the web link is redirected through the link-shortening service through to the final linked destination. During the re-direction, the link shortening service collects some information about the user’s system including the geo-location of the system, the date and time of the click, operating system, the referral source (i.e. what application hosted the clicked link), and the intended link destination. The data consists of a collection of these link transactions over a given time period with each observation in the set being a unique click on a short link. An example of this data is shown in Appendix A.

3.3 Co-Purchase Graphs

The primary technique used in this research is using graph mining techniques, specifically community detection, on a co-purchase network. A co-purchase graph is a graph containing products connected by edges of common purchasers. In the case of shortened web links, products are web domains and edges are common users between web domain pairs.

3.3.1 Build a bi-partite click graph

A graph G contains a set of edges, E which interconnect a set of vertices, V . A special subset of graphs is a bipartite, or bi-modal, graph. These graphs contain two types of vertices which do not contain any inter-connectivity between the two. Given graph G , users are defined as $V(G) \in \{A_1, \dots, A_m\}$ where m is defined as the total number of observed users. Web domains are then defined as $V(G) \in \{B_1, \dots, B_n\}$ where n is defined as the total number of visited web domains. A common method for displaying graphs is through an edgelist. An edgelist is a two-column matrix that displays edges by row with the originating vertex in the first column and the ending vertex in the second column. The edgelist

$$G_I = \begin{bmatrix} A_1 & B_2 \\ A_1 & B_3 \\ A_2 & B_1 \\ A_2 & B_2 \\ A_2 & B_4 \\ A_3 & B_1 \\ A_3 & B_2 \\ A_3 & B_3 \end{bmatrix} \quad (1)$$

shows that user A_1 visited web domains B_2 and B_3 . Additionally, the edgelist shows that no interconnections exist between the two types of nodes A_i and B_i .

Another common representation of the graph is the adjacency matrix $\mathbf{A}$. This transforms the edgelist which contains edges by rows into an $m \times n$ matrix with users A_i along the rows, web domains B_i along the columns, and entries in the matrix indicating if the i^{th} user visited the j^{th} web domain. The adjacency matrix

$$\mathbf{A}_{bipartite} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \quad (2)$$

shows that user A_1 visited web domains B_2 and B_3 which is identical to the edgelist from Equation 1.

From this point, the co-purchase graph must be built using

$$\mathbf{A}_{copurchase} = \mathbf{A}_{bipartite}^T \mathbf{A}_{bipartite} \quad (3)$$

which generates a new $n \times n$ matrix with web domains along the rows and columns and entries in the $\mathbf{A}_{copurchase}$ matrix, $A_{ij} \in \mathbb{N}$, indicate the number of co-users between web domain B_i and B_j . Web domains should not have common users with itself, so the main diagonal is zeroed out[27]. The resulting co-purchase adjacency matrix from graph G above is

$$G_1 = \begin{bmatrix} 0 & 2 & 1 & 1 \\ 2 & 0 & 2 & 1 \\ 1 & 2 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \quad (4)$$

3.4 Community Detection Algorithms

Community detection is “one of the fundamental tasks in social network analysis” [27, p. 8] and a number of algorithms have been proposed to help find meaningful clusters or groups of nodes within graphs. However, original community detection methods were insufficient in large scale networks. These algorithms had been created prior to the explosion of large scale networks, they were often too time or resource intensive

to be used [27, 28]. From this need for more resource-responsible arose different community detection algorithms that run closer to linear time and provide useful results. The two selected for further investigation are the Speaker-Listener Label Propagation method proposed by Raghavan et al. [28] and the Walktrap method proposed by Pons and Latapy[29].

3.4.1 Speaker-Listener Label Propagation Algorithm

The first community detection method used is speaker-listener label propagation (SLPA). Raghavan et al.[28, p. 4] proposed that “each node in [a] network chooses to join the community to which the maximum of its neighbors belong to, with ties broken uniformly randomly.” SLPA relies on nodes passing community labels between each other in rounds. During each round of label passing, a node collect labels from connected nodes and changes its current label based on the set of labels received. After each node has had labels passed to it, the algorithm moves to the next round and conducts label passing for each node again. This continues until each node has reached a maximum number of identical labels based on their neighbors. Nodes are then grouped together into communities based on their common labeling.

At time (i.e. round), $t = 0$, each node x in the graph is initialized with a community label $C_x(0) = x$. The nodes are then randomly arranged and set to X . Then, for each $x \in X$ in that order,

$$C_x(t) = f(C_{x_{i1}}(t), \dots, C_{x_{im}}(t), C_{x_{i(m+1)}}(t-1), \dots, C_{x_{ik}}(t-1)) \quad (5)$$

where $C_{x_{im}}(t)$ are the community labels of the nodes preceding x and $C_{x_{i(m+1)}}(t-1)$ are the community labels of the nodes after x . The function counts each passed label and selects the maximum occurring label with ties broken uniformly randomly. Equation 5 is defined as asynchronous updating because previously updated (i.e. $C_{x_{im}}(t)$) nodes

at time t pass updated labels. This is opposed to synchronous updating where at time t each node sends the label assigned from time $t - 1$.

After each node $x \in X$ has been updated with its new label, t is incremented by one and the nodes are randomly re-arranged into X unless every node x has the same label as the maximum of their neighbors, at which point the algorithm is stopped.

Due to the random variation in both ordering X and the random uniformly broken ties between labels, the algorithm can produce different results depending on the ordering of the nodes[28]. These effects can be mitigated by running the algorithm through the graph multiple times and combining the results together if desired. Regarding time complexity, initializing the labels requires $\mathcal{O}(n)$ time while each label propagation step requires $\mathcal{O}(m)$ time where m is the number of edges in the network[28].

3.4.2 WalkTrap Algorithm

The Walktrap algorithm is community detection method that focuses on merging sub-communities until optimality conditions are met. The algorithm uses *Random Walks* to determine which nodes should be grouped into communities which logically posits that vertices traveled to often are more likely to be in a community than vertices that are not often visited.

3.4.2.1 Random Walks

With *Random Walks*, imagine a hypothetical person standing at a random vertex, V_i , in graph G who then takes a step towards another random node traveling only through interconnecting nodes. Each independent path that walker can take is called a *Markov Chain*[29]. The probability of the walker traveling ending at node j after

one step is

$$P_{ij} = \frac{A_{ij}}{d(i)} \quad (6)$$

where P_{ij} is the probability of transitioning from vertex i to vertex j , A_{ij} is the adjacency matrix A of graph G , and $d(i)$ is the sum total of all edges leaving vertex i . For example, in graph G_1 above, the probability of traveling from vertex, V_1 to vertex V_4 is 1 divided by the total weights of edges leaving V_1 which is 4. By transforming $d(i)$ into matrix form where D is the diagonal matrix with $D_{ii} = d(i)$ and $D_{ij} = 0$ for all i nodes, the total probability matrix is calculated as

$$P = D^{-1}A \quad (7)$$

where each P_{ij} is the transition probability between vertex i and vertex j . Extending this from one step, P_{ij}^t is the transition probability of starting at vertex i and ending at vertex j after t steps[29]. Applied to the co-purchase adjacency matrix in 4, P_{ij}^t is the transition probability of a random walk originating at web domain i and ending at web domain j after t steps.

3.4.2.2 Distance Calculations

After determining the transition probabilities, Walktrap then calculates the distance between nodes and communities. Beginning with the distance between two nodes, Pons and Latapy [29] show that the distance between vertex i and vertex j , r_{ij} after t steps is

$$r_{ij} = \sqrt{\sum_{k=1}^n \frac{(P_{ik}^t - P_{jk}^t)^2}{d(k)}} \quad (8)$$

Generalized, the distance $r_{C_1 C_2}$ between communities C_1 and C_2 is

$$r_{C_1 C_2} = \sqrt{\sum_{k=1}^N \frac{(P_{C_1 k}^t - P_{C_2 k}^t)^2}{d(k)}} \quad (9)$$

When nodes or communities are closely related, their Euclidean distance as calculated above should be small while further apart communities will have large distances.

3.4.2.3 Merging Communities

With the transition probabilities and distance calculations above, we must now merge nodes into communities. The algorithm begins by assigning each vertex to its own community $C_i = V_i$. The algorithm then begins merging communities that “minimize the mean σ_k of the squared distances between each vertex and its community.[29, p. 8]” where σ_k is the mean distance between node k and community C

$$\sigma_k = \frac{1}{n} \sum_{C \in \mathcal{P}_k} \sum_{i \in C} r_{iC}^2 \quad (10)$$

where r_{iC}^2 is the distance of node i and community C calculated using equation 9. This is then summed across the total distance of all nodes within community C and for all communities within the current partition set $\mathcal{P}_k$

By iteratively combining communities based on minimizing distances, communities are merged into larger communities. This algorithm is repeated until the modularity Q is maximized. Modularity [30] is a measure of the interconnectedness of a system’s substructures and is defined as

$$Q = \sum_i (e_{ii} - a_i^2) \quad (11)$$

where e_{ii} is the proportion of edges that begin and end within community i and a_i is the proportion of *all* edges that end in community i . Modularity ranges from $[-0.5 - 1)$, and as modularity approaches 1, these graphs have much larger propor-

tions of edges contained within communities relative to edges leaving the community. This indicates that substructures occur within the system and can be identified. A graph with a negative modularity score indicates that more edges leave communities than are contained within communities which implies that there is no identifiable substructure relationship in the graph. Conversely, as modularity increases towards 1, the proportion of edges wholly contained within communities begins to outweigh the proportion of edges that leave those communities until eventually all edges are contained within communities. A modularity score approaching 1 implies that identified communities have no interactions with each other.

When Q is maximized, the algorithm is terminated and the communities are returned. This algorithm at its worst case in time $\mathcal{O}(mn^2)$ while most real-world graphs will run in $\mathcal{O}(n^2 \log(n))$ [29].

3.4.3 Filtering Techniques

When investigating large graphs, such as from the shortened web link click data, a number of spurious connections can influence the overall communities detected. To help reduce the impact of these connections on the overall graph, a filtering method must be used. Videla-Cavieres and Rios [25] proposed a top-three heavy edges threshold ($tthet$) which calculated the mean of the top three heaviest weighted edges

$$tthet = \frac{E_{max} + E_{2^{nd}max} + E_{3^{rd}max}}{3} \quad (12)$$

from the co-purchase adjacency matrix, $A_{copurchase}$.

The $tthet$ is then multiplied by a vector of percentages, $percentages = \{1\%, 2\%, \dots, 10\%\}$. The cross product of $tthet$ and $percentages$ provides vector of applicable filters $filters = \{0.01 * tthet, \dots, 0.1 * tthet\}$ that can be applied iteratively and the resulting communities can be investigated at each filter step. Selecting the best filter level is

done by determining which filter level generates the highest modularity for a given graph as calculated in 11.

3.5 Community Detection Toy Example Application

Next, we will apply the two community detection methods discussed above to a toy example in the R Software environment[31] using the iGraph package[32] available for R (see Appendix B for notes on all software packages used). The toy example with known labels will help determine provide a basis against which to compare the two methods. Finally, we will select the best performing method from the case study.

3.5.1 Amazon Book Co-Purchase Network Toy Example

In 2004, Krebs created a dataset of a co-purchase network of Amazon books[33]. Krebs classified each book as either liberal, conservative, or neutral and is shown in figure 1.

As expected, there are clear clusters between political leanings. There are many more edges within books of a particular leaning than books outside of a political leaning. This follows logically as shoppers would probably be more likely to purchase multiple books within a particular political classification. Therefore, we will try to determine by co-purchase methods which books can be grouped together and test the accuracy of each method. While the ground truth labeling happens to be by political leaning, the only information used by the community detection methods is whether the books were purchased by common shoppers. Therefore, this graph should provide a good representation of performance on co-purchased shortened web links.

Ground-truth Co-Purchase Community Plot

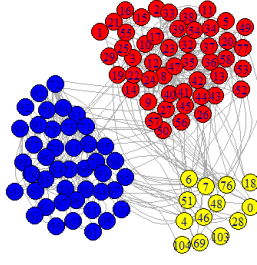


Figure 1. Amazon Co-Purchase Ground Truth Network

3.5.2 Implementation of Community Detection Algorithms

The two methods, SLPA and Walktrap, are applied to the co-purchase matrix. Additionally, due to the random label selection and passing in SLPA, the methods are applied 100 times each with sequential random seeds applied identically between the two methods. After classifying, each method's accuracy is then calculated. Within the network each node, v is assigned a true label, l_{tv} . Then within each detected community, i , the preponderance of true labels within the community is assigned as the predictive label, l_{pv} , for each node, v , within the i^{th} community [34]. The accuracy is then calculated as

$$\text{Accuracy} = \frac{\sum_{v=1}^n \text{equal}(l_{tv}, l_{pv})}{n} \quad (13)$$

where $\text{equal}(x,y) = \begin{cases} 1 & \text{if } x \text{ is identical to } y \\ 0 & \text{otherwise} \end{cases}$.

Figure 2a shows how the books were classified individually while Figure 2b shows the SLPA classification accuracy for random seeds 1 to 100.

The accuracy centers around 85% and shows some variation as the random seed changes. This classification accuracy is better than naively classifying books randomly

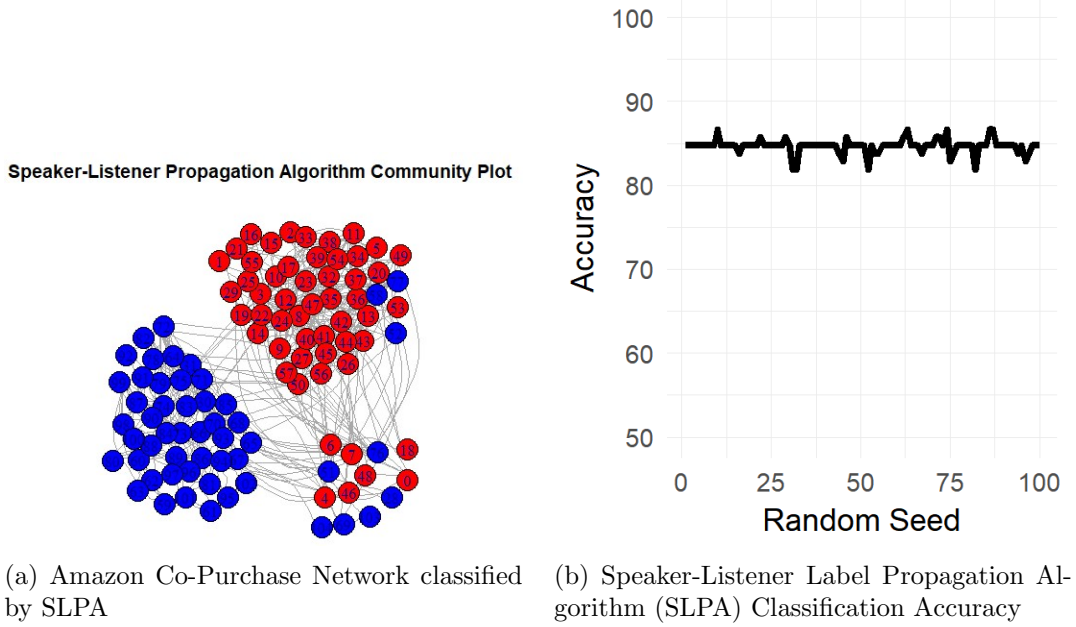


Figure 2. SLPA Community Detection results

between the three different political leanings. Additionally,

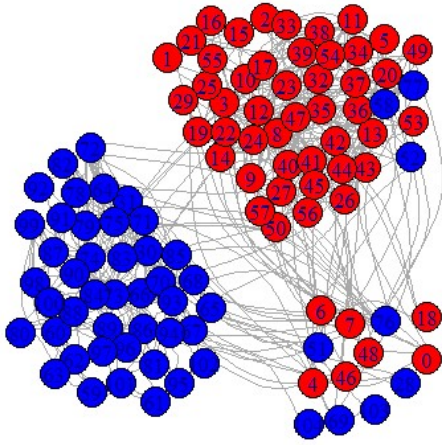
We can see that SLPA did a fairly good job classifying each book based on its political leaning with the exception of the neutral-leaning books. There were not enough common book purchases between the neutral-leaning books to overcome the connections made with either liberal- or conservative- leaning books.

Figure 3a shows the WalkTrap classified network and Figure 3b shows the accuracy as the Walktrap algorithm is applied to the co-purchase graph using the same 1 to 100 random seeds as before.

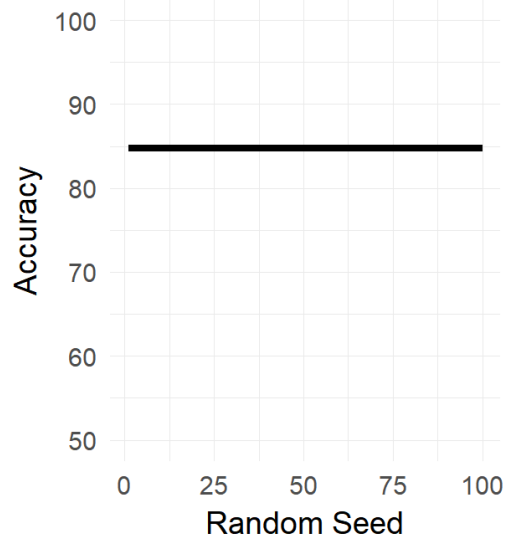
This method is clearly determinant in its detection method. There is no variance in the classification accuracy which is also approximately 85%.

The Walktrap Algorithm had the same issue with classifying neutral-leaning books. The graph in Figure 3 is identical to the SLPA graph in Figure 2a which indicates that the two community detection methods provide similar results.

WalkTrap Algorithm Community Plot



(a) Amazon Co-Purchase Network classified by Walktrap



(b) Walktrap Classification Accuracy

Figure 3. Walktrap Community Detection results

3.5.3 Results & Algorithm Selection

Both of the methods described above worked very well in classifying books based on ground-truth labeling according to the co-purchase behavior of consumers. This indicates that the classification methods should work fairly well on a co-purchase network of shortened links as well. The results of the 100 trials of each method are shown in Table 2

Table 2. Toy Example Classification Accuracy Results

	Average Accuracy	Standard Deviation
SLPA	84.73%	0.759%
Walktrap	84.76%	0.0%

In addition to the performance results in Table 2, the computational time difference was negligent with each algorithm completing in less than 0.1 seconds. Based on the classification results above, the Walktrap community detection method will be used for future analysis. The determinant nature of the algorithm reduces variability

without sacrificing accuracy levels. The results indicate that the Walktrap community detection method will provide reliable results that provide meaningful clusters between co-purchased items. Finally, algorithms were performed on a small graph consisting of fewer than 100 nodes. A larger dataset allows for more variability in the results of SLPA, unlike the Walktrap method, which places a higher importance on the agglomerative methods and adds another layer of complexity to the problem. Finally, applying these algorithms to a much larger dataset will cause an increase in computational time, but the linear-time nature of the algorithm should mitigate most of the time increase.

IV. Results & Analysis

4.1 Introduction

As discussed in Chapter I, the purpose of this research is to understand better a particular area's information environment. Due to the constraints on file size, a subset of the United States was needed for analysis. The United States Census Bureau classifies Charlotte[35] as a large city which helped ensuring enough observations to conduct analysis but small enough to fit the file size thresholds. A 2017 report[36] showed Charlotte as the 12th most diverse city in the United States. In addition to city size and diversity, Charlotte[35] mirrors the overall United States trends in a number of demographic statistics. According to the census, the population of Charlotte is approximately 842,000 people of whom 74.2% are over the age of 18. Approximately 88.4% of Charlotte residents have a high school diploma and 42% of Charlotte residents have at least a Bachelor's degree and the median income in Charlotte is \$55,599. In 2015, the US Census Bureau also studied internet access across the United States and found that 76.7% of American households had an internet subscription of any kind. Additionally, nearly 61% of American households maintained both a smartphone and a personal computer in addition to their internet subscription.

Based on the above, Charlotte fit the criteria needed to be a representative subset of the United States and two case studies are investigated to demonstrate the types of information that can be learned about a specific area. These case studies investigate Charlotte, North Carolina during two major events in 2016: the Presidential Inauguration and the Super Bowl[®]. Each dataset began at 5 a.m. on the first listed day and ended at 5 a.m. 48 hours later. Using the methodology shown in Figure 4, each data set was investigated for overall patterns.

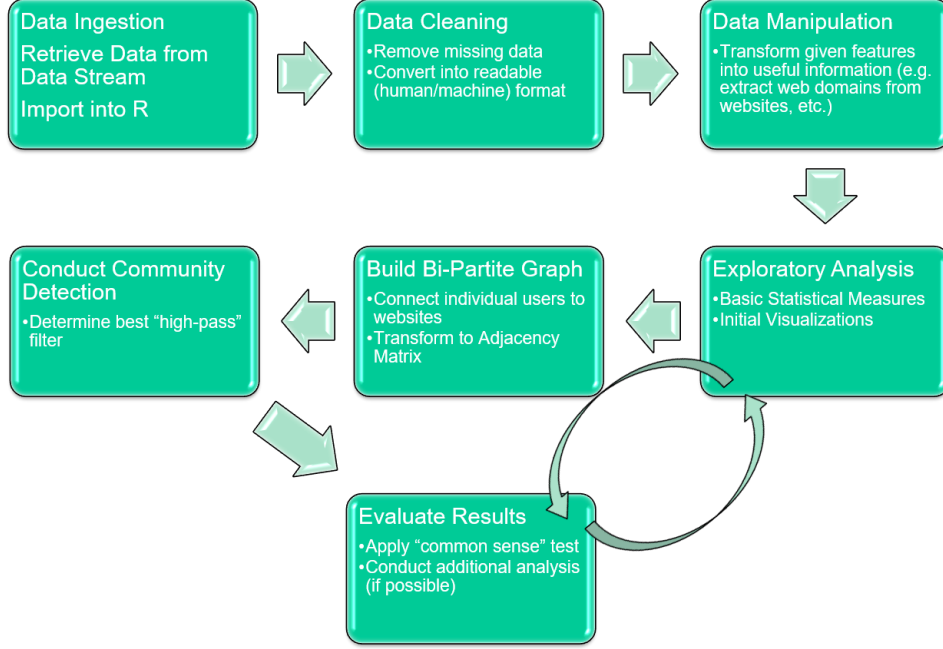


Figure 4. Methodology

Following the exploratory data analysis, we generate a copurchase graph for both data sets and conduct community detection to try to find differences or similarities in any identified communities.

4.2 Exploratory Data Analysis

The purpose of exploratory data analysis (EDA) is to provide some initial observations about the data. The initial observations, mostly count statistics, will help to provide a baseline about which future results can be compared. They provide an overall view of the area and can provide an idea of the general methods used to access information in a given area. First, the data is loaded and is cleaned. The dataset originally consists of 47 columns (see Appendix A for column names) of which we are primarily interested in the HTTP User Agent and Authority URI columns as these are the columns that define the user profile for devices accessing the shortened links and the web domains that were clicked. Web domains, such as Internet Protocol (IP)

addresses, web advertisement domains, and NA values, are removed to help maximize information-based websites. The results of this data cleaning step is shown in Table 3.

Table 3. Remaining Web Clicks following Data Cleaning

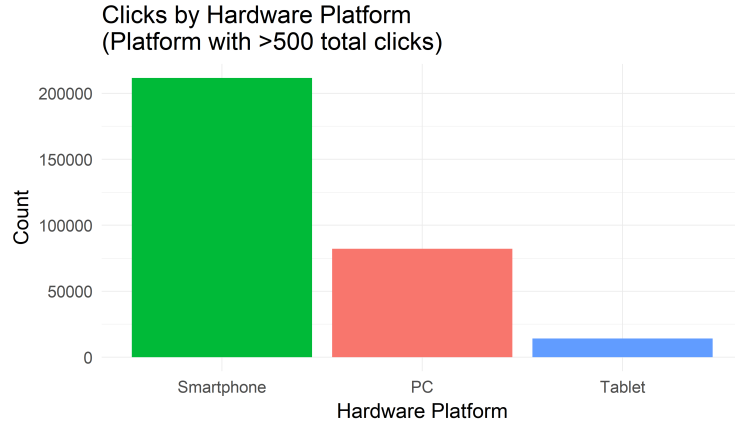
Process Step	Jan 20-21, 2017		Feb 5-6, 2017	
	Remaining	Loss(%)	Remaining	Loss(%)
Load Data	381,445	-	303,883	-
Remove Ads/ IP Addresses	313,018	-17.9	255,611	-15.9
Remove Single Click Domains	303,656	-3.0	246,336	-3.6
Remove NA Domains	303,406	-0.001	246,074	-0.001

After removing the domains, the final count for January 20-22, 2017 (Friday morning-Sunday morning) and February 5-7, 2017 (Sunday morning-Tuesday morning) is 303,406 and 246,074 clicks respectively. The total number of web clicks for each weekend are within 20% of each other indicating a fairly standard amount of shortened web link usage between the two weekends which should follow considering only approximately two weeks had passed between the two events.

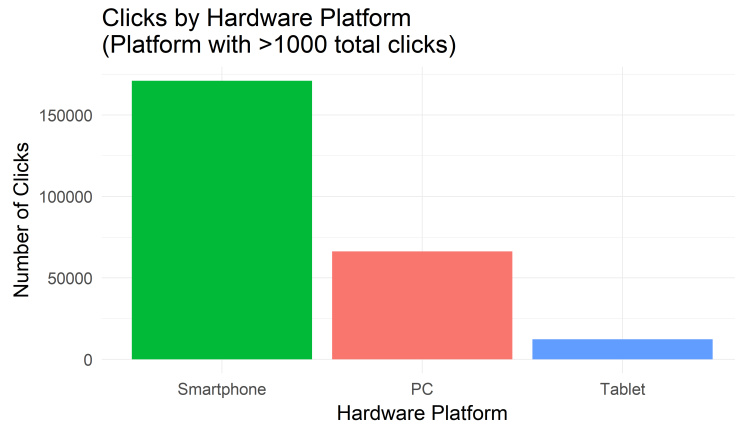
Next, the clicks, by weekend, are investigated based on the device Operating System (OS) type, Referrer Type, and Hardware Platforms and the results are shown in Figures 5-7.

Both weekends show fairly consistent activity between OS type, referrer type, and hardware platforms. For each weekend, Figure 5 shows that the majority of link clicks originate on a smart-phone type device followed by a personal computer. Other internet-connected devices such as tablets or gaming systems constitute a vastly smaller proportion of web clicks in Charlotte.

Next, we can look at the Operating Systems used to access the web domains. Figure 6 shows that the greatest number of clicks originate from an Apple iOS Operating System. This piece of information, combined with the hardware platform



(a) January 20-22 2017



(b) February 5-7 2017

Figure 5. Access Methods by Hardware Platform

indicate that a plurality of users accessing shortened links are using Apple’s iPhone while Google’s Android OS smartphone are a close second.

Finally, regarding referral type, a plurality of clicks are originating from Facebook as opposed to other locations such as Twitter or a website. Based on this information, we gather that information in Charlotte is best disseminated via Facebook and should be optimized for a smartphone device.

Next, a time-analysis is conducted to identify access patterns related to time. Figure 8 shows the distribution of link clicks by time of day for January and February respectively.

For both weekends, a fairly consistent social pattern of web-clicks emerge with

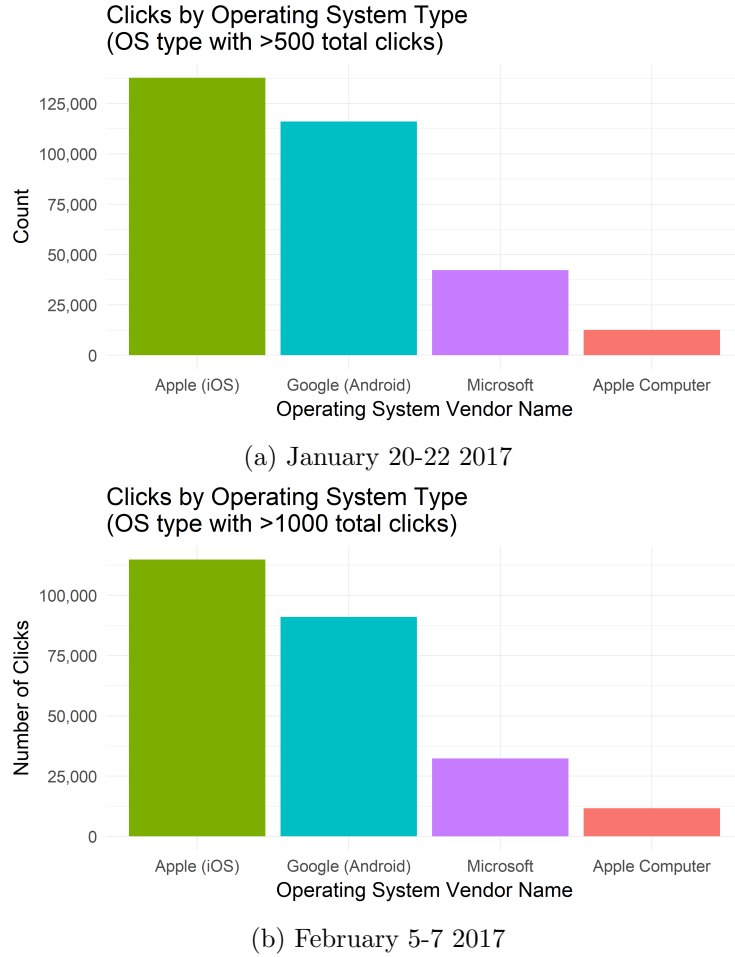
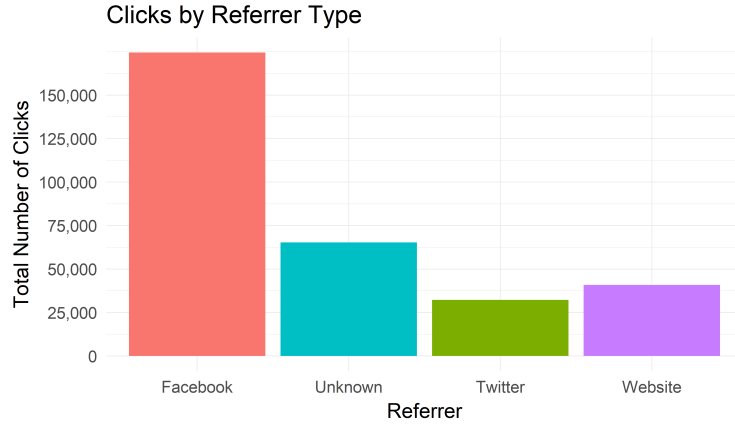


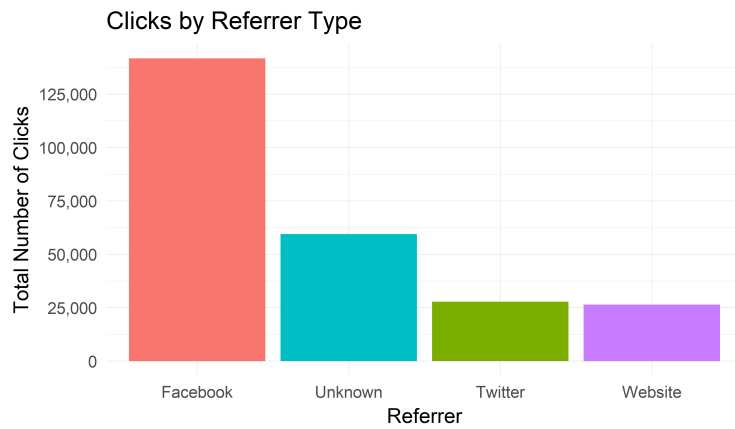
Figure 6. Access Methods by Operating System

most clicks increasing during the morning hours (after 8 a.m.) until peaking during the evening where they fall off in the early morning hours (pre-8 a.m.). In January, there is a sharp peak around noon on 20 January which marked the beginning of the Presidential Inauguration. This peak is not replicated on any other days, even for the Super Bowl (February 5th). The inauguration was clearly a driver for web traffic in Charlotte. In addition to the social cycle of clicks, Figure 9 shows a difference in the latency between link creation and link-click times.

Twitter showed the fastest turn around with a sharp peak of the density curve at a link age of approximately 1 hour. The other referral sources skewed towards a shorter link age indicating that most links are clicked on very shortly after having



(a) January 20-22, 2017



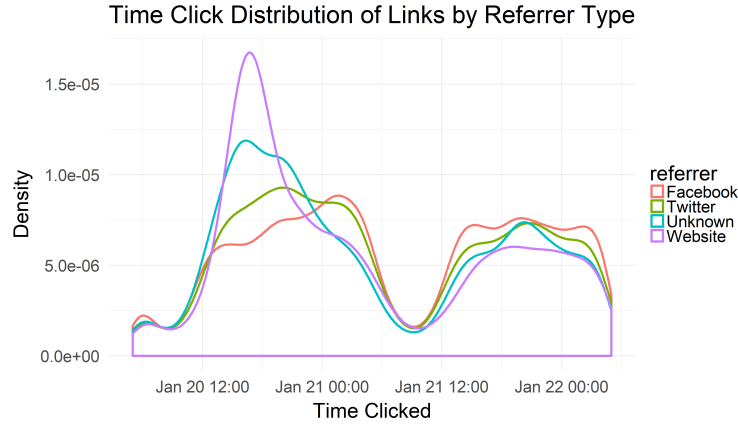
(b) February 5-7, 2017

Figure 7. Access Methods by Referrer Type

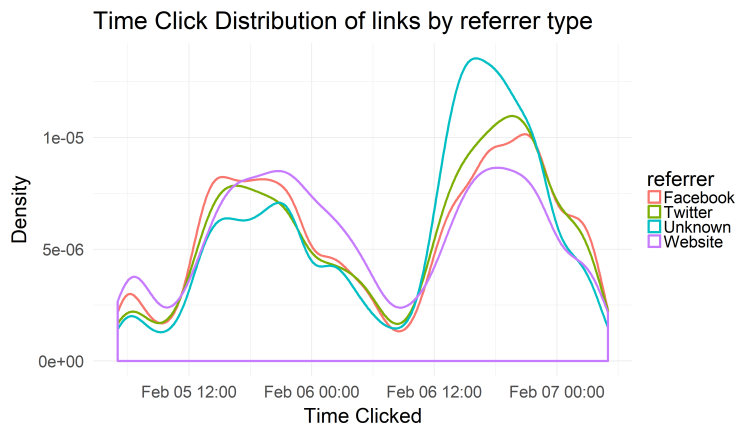
been created. However, referral sources besides Twitter have a longer tails indicating that the links on webpages and Facebook seem to have a longer life.

Next, each of the individual links were investigated for basic information regarding their respective weekends. Figures 9a and 9b highlight the proportion of link clicks that contained keywords related to the inauguration or Super Bowl respectively. These links are then broken down to determine if there are significant differences in the overall accessing of the event-related links.

Figure 10 shows the proportion of users in January who clicked on a link related to the Inauguration. Figure 11 shows the number of users who clicked on a link related to the Super Bowl in February.



(a) January 20-22, 2017

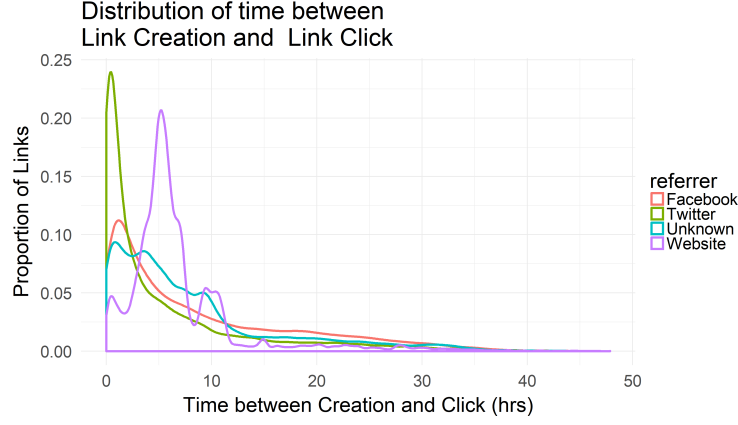


(b) February 5-7, 2017

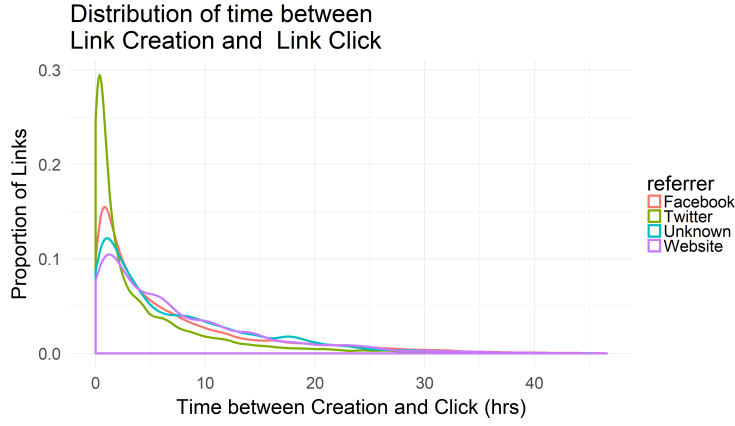
Figure 8. Link Click-Time Distributions

These charts show for both weekends that more links unrelated to either the Inauguration or the Super Bowl were clicked. However, Figure 10 shows a higher proportion of clicks related to the inauguration than the proportion of clicks related to the Super Bowl in Figure 11. These charts also show that clicks to this type of information was not affected by OS system, hardware platform, or referrer type.

In aggregate, both weekends provided a fairly consistent look at shortened web link activity for Charlotte. The data showed that web clicks followed a social-type of activity with peaks of activity during the evening and low-points in the early morning hours. Additionally, the majority of clicks originated from smart-phone devices and users were primarily using Facebook to access the links. A key distinction between



(a) January 20-22, 2017



(b) February 5-7, 2017

Figure 9. Link Creation-to-Click Times

the two weekends is that the inauguration weekend generated more clicks than the Super Bowl in Charlotte. This is shown in the greater number of overall clicks as well as the higher proportion of clicks related to the inauguration as opposed to the Super Bowl. Despite this difference, the breakdown of links related to the weekend's event were not impacted by OS, hardware platform, or referrer type. However, conducting community detection on the co-purchase graph should allow us to determine if these overall observations hold within each sub-community.

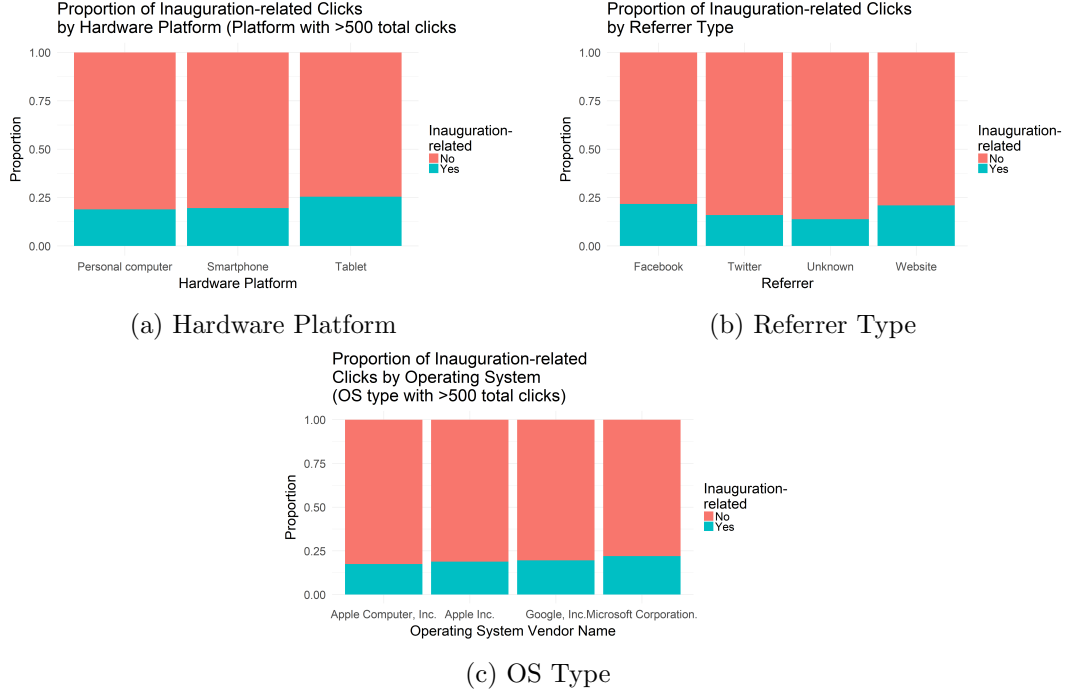


Figure 10. Inauguration-related Clicks by access type

4.3 Community Detection

To try to draw conclusions from the types of websites common users visited, a graph is built connecting users to visited domains. To do this, the HTTP User Agent column is paired with the Latitude and Longitude location columns and a unique id is created for each user. Table 4 shows the breakdown of links, unique domains, and unique users for the two weekends provided with an average of 6.6 and 5.77 average clicks per user for each weekend, respectively.

Table 4. Click-, Unique Domain-, and User-Counts

	Total Clicks	Unique Domains	Unique Users
January 2017	303,406	9,600	45,980
February 2017	246,074	9,234	42,646

A bipartite graph is created connecting users with the individual domains and a co-purchase graph is built using the techniques from Chapter III. Before the community

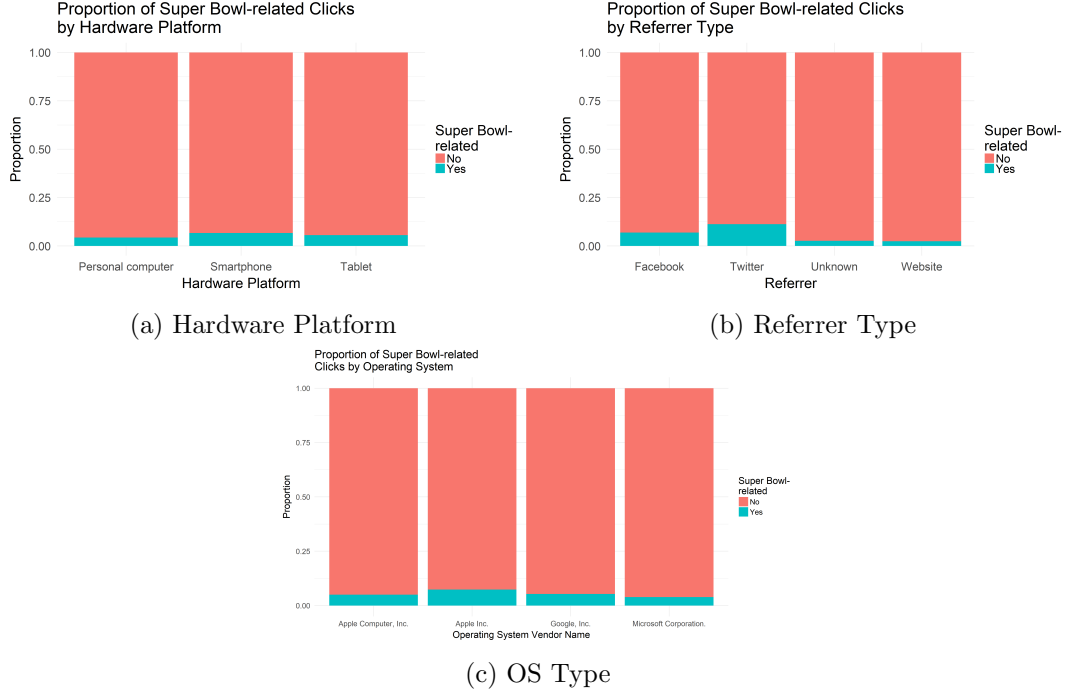


Figure 11. Super Bowl-related Clicks by access type

detection techniques can be used, the filter is applied to the co-purchase matrix. The filtering of 1% to 10% of the top-three heaviest edge threshold is used to remove the spurious connections. After filtering the Walktrap community detection method is applied to the filtered co-purchase matrix and the modularity score is calculated for the communities. The modularity score for each filter as applied to the January 2017 dataset is shown in Figure 12a.

Figure 12c shows the rapid decrease of remaining domains as the filter is increased. This shows the power-law relationship between the filter and remaining products(domains) discussed by Videla-Cavieres and Rios[25]. Additionally, as the number of domains decreases, we would expect the number of identified communities to decrease as well. This relationship is shown in Figure 12b. Figure 12a is maximized at $Q = 0.10$ at a filter of 9%. This value is low, but still indicates that connections between the web domains in each community occur at a higher probability than random chance. Applying this filter leaves 32 domains in 12 communities.

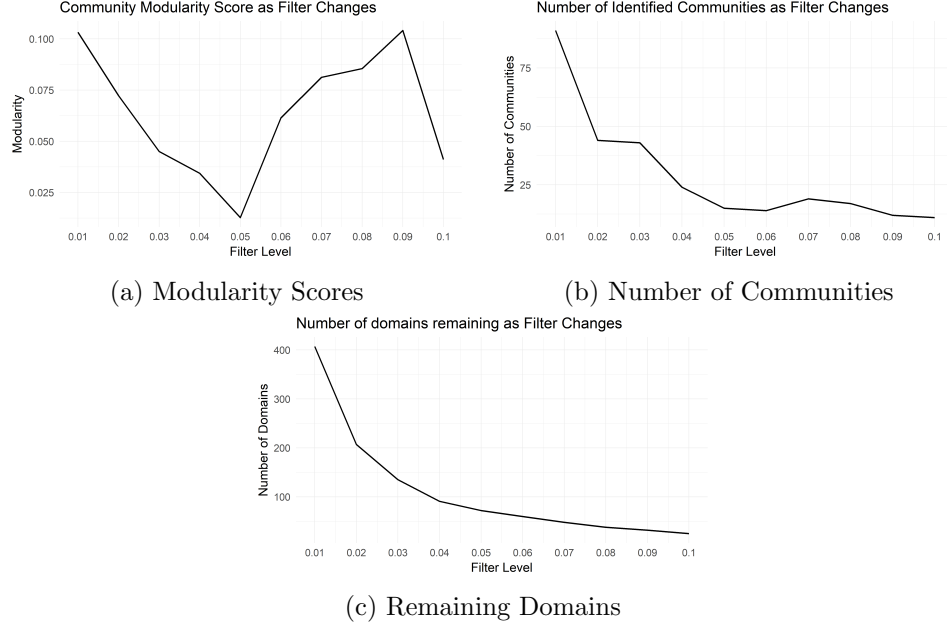


Figure 12. Community Detection Measures related to Filter size (January 2017)

We can apply this same methodology to the February dataset and the results are shown in Figure 13

The modularity is also maximized at $Q = 0.11$ in the February dataset when a 9% filter is applied as seen in Figure 13a.

Next, we investigate the communities identified after applying the filter. Figure 14 shows the number of domains per identified communities.

In both datasets, there are two identified communities which contain more than one domain. Communities 3-n are all single domain clusters indicating less connectedness to the remaining domains.

The domains within each community and the associated number of clicks are shown in Tables 5-6. Based on the general similarities of the domains within each of the communities, we can also determine names for the groups for ease of use. Looking at community 1 in Table 5 which contained 13 domains, we see that most of these refer to news service websites and therefore rename it to the News community. The second community, containing 9 domains, is re-named Entertainment.

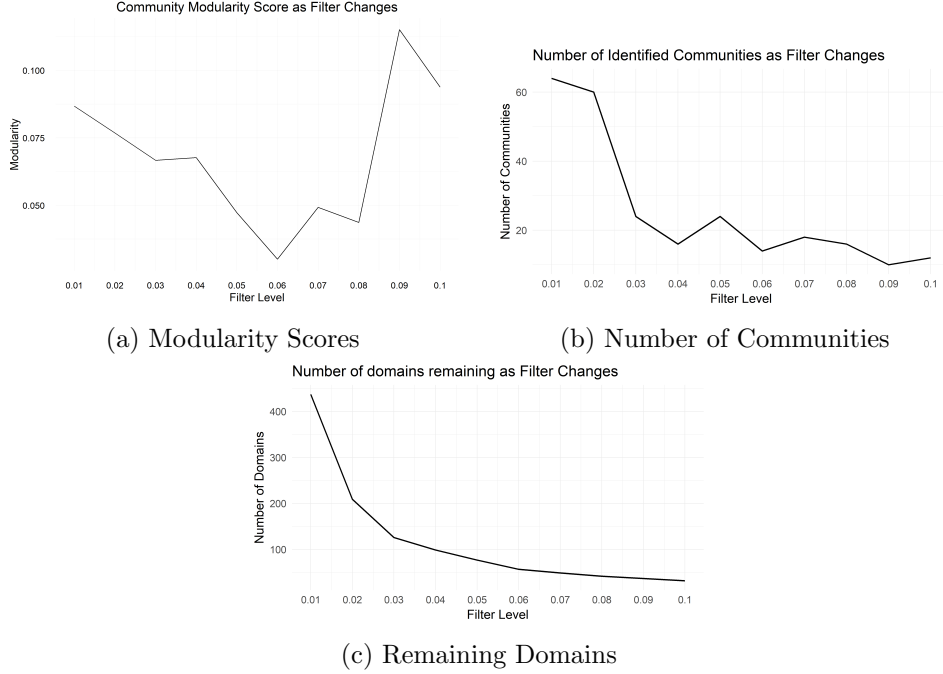


Figure 13. Community Detection Measures related to Filter size (February 2017)

This is then applied to the February Dataset as well. We see in Table 6 that the two domains contain more websites than January. Additionally, there is not as clear a breakdown between News and Entertainment websites as in January. This is most likely due to the absence of the news-dominated coverage of the Inauguration. Looking at the web domains present, we identify the first community as consisting mostly of national- or mainstream-focused websites while the second community seems to contain mostly local or alternative-focused information. Therefore, these two communities are renamed National/Mainstream and Local/Alternative respectively.

Applying the community detection technique on the co-purchase graph allows for some observations. The edge weights of the co-purchase graph are the number of co-users of web domain pairs, therefore pairs with high weights have large numbers of co-users (i.e. a large number users visited both web domains that weekend). This implies that shortened link traffic is not random but that users accessing websites are much more likely to travel to a specific set of other websites. However, Tables 5-6

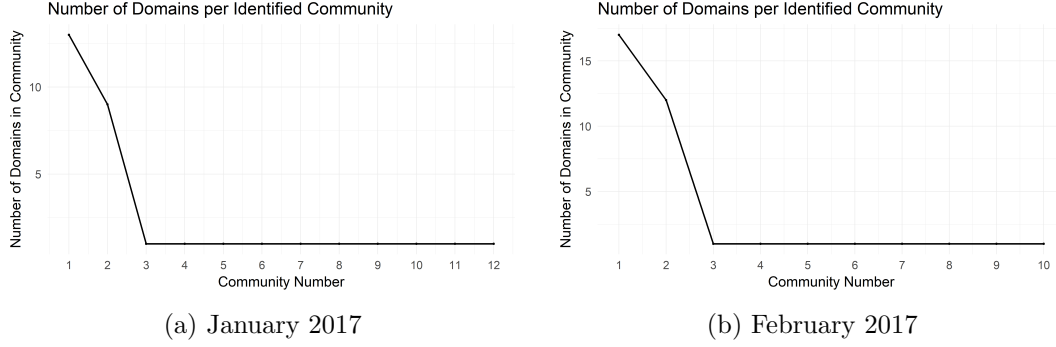


Figure 14. Number of Domains per Identified Communities

Table 5. Domain and Web-Clicks for January 2017

“News”		“Entertainment”	
Web Domain	# of Clicks	Web Domain	# of Clicks
www.cnn.com	13,879	boomely.com	8,993
libertywritersnews.com	9,295	www.theblackloop.com	5,396
www.wbtv.com	5,633	www.bet.com	4,184
www.espn.com	5,091	blackdoctor.org	3,178
ilovemyfreedom.org	4,751	michaelbaisden.com	2,850
www.buzzfeed.com	3,459	www.wistv.com	2,601
www.youtube.com	2,640	www.live5news.com	1,870
www.hometalk.com	2,535	www.sunfrog.com	1,733
www.businessinsider.com	2,203	www.iloveoldschoolmusic.com	1,663
www.washingtonpost.com	2,171		
trib.al	2,101		
www.foxnews.com	1,674		
www.nydailynews.com	1,560		

demonstrate that users do not necessarily access websites similarly as time passes. This indicates that web clicks will probably have a more dynamic community structure that can be affected by political and social events.

4.4 Community Measures

Continuing with the methodology in Figure 4, the two identified communities from both January and February are investigated separately and compared to their respective overall dataset. These communities contain the largest co-user pairs after

Table 6. Domain and Web Clicks for February 2017

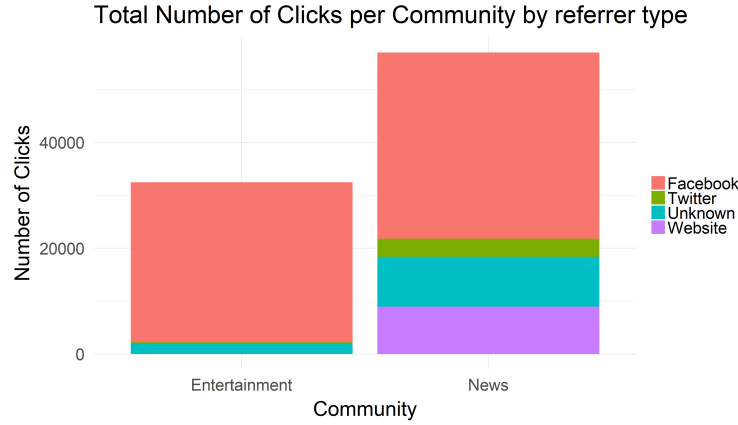
“National/Mainstream”		“Local/Alternative”	
Web Domain	# of Clicks	Web Domain	# of Clicks
www.espn.com	6,597	www.iloveoldschoolmusic.com	3,519
www.wbtv.com	4,642	rewely.com	3,170
www.buzzfeed.com	3,037	blackdoctor.org	2,997
www.youtube.com	3,018	shareably.net	2,808
www.simplemost.com	2,930	www.theblackloop.com	2,755
www.hometalk.com	2,493	socialtrendbuzz.com	2,703
awm.com	2,182	www.wistv.com	2,446
www.amazon.com	2,119	www.live5news.com	1,950
www.businessinsider.com	2,076	michaelbaisden.com	1,520
trib.al	2,070	www.tnztoday.net	1,338
en.newsner.com	1,729	thebreakfastclub.iheart.com	1,257
www.tlc.com	1,500	rickeysmileymorningshow.com	1,179
www.totalprosports.com	1,081		
radaronline.com	1,077		
littlethings.com	1,045		
www.nytimes.com	930		
www.nydailynews.com	906		

a 9% filter is applied to the co-purchase graph. Therefore, similarities between users should be detected between the two communities in each month. Additionally, the trends within a community should be sufficiently different from either each other or the overall data set to draw conclusions about those users.

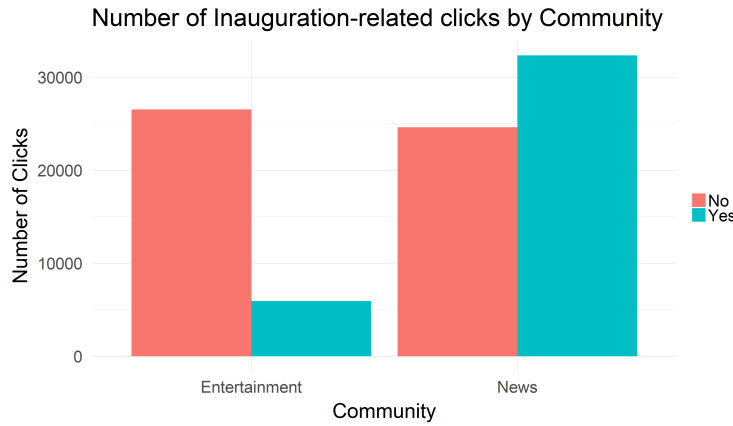
4.4.1 January 20-22, 2017

In January 2017, the News and Entertainment communities earned a disproportionate number of clicks. The 22 domains from these two communities (originally 9,600) received 89,460 of the clicks or 28.5% of the total number of clicks that weekend. As shown in Figure 15, a larger number of clicks went to the News community. This is mostly due to the larger number of domains contained within the News community.

Additionally, based on referrer types, both communities are overwhelmingly rep-



(a) Clicks Per Community by Referrer Type



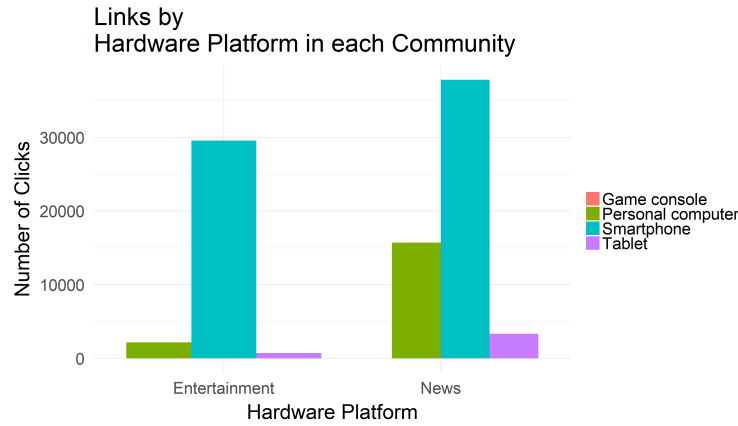
(b) Referrer Type

Figure 15. Community Information Measures (January 2017)

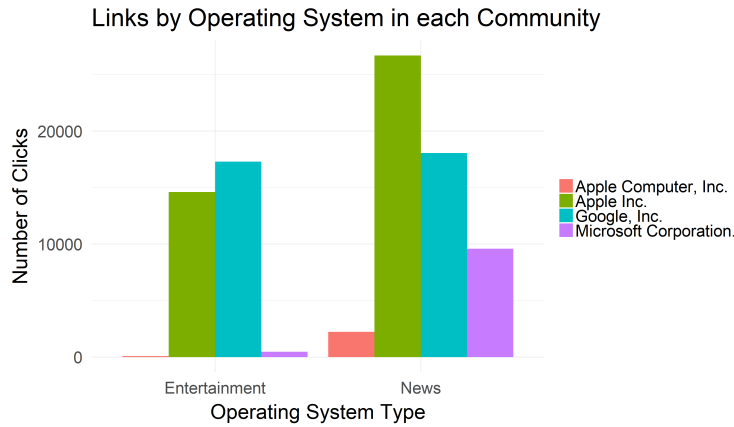
resented by Facebook referrals. This is in line with the referral pattern demonstrated in 4.2. However, Figure 15a shows a much larger spread of News referrers outside of Facebook, indicating that users accessing these web domains were more likely to be accessing directly from another website or from an unknown location. Figure 15b shows the breakdown of clicked links by community when separated by link containing inauguration-related terms. The News community contained a larger proportion of links related to the Inauguration (32,362 clicks/56.8%) than both the Entertainment community (5,932 clicks/18.3%) and the overall dataset as shown in Figure 10. This indicates that users accessing the web domains in the News community for Inauguration-related information were more likely to access other web domains that

also contained Inauguration-related information.

When comparing hardware platforms and OS's for each community, similar patterns emerge. Figure 16a shows that the majority of web clicks to these web domains came from smartphone devices. While Entertainment domain clicks were almost exclusively from smartphones, a much larger proportion of clicks originated from personal computers in the News community web domains. Figure 16b shows the breakdown by OS for each community.



(a) Clicks Per Community by Platform Type



(b) Operating System

Figure 16. Access Types by Community (January 2017)

When combined with Figure 16a, we see that more web domains were accessed by an Apple iPhone device in the News community than by an Android phone which is in line with the overall breakdown discussed in Section 4.2. This does not hold

for the Entertainment community where more clicks came from Android-type devices as opposed to Apple's iPhone. Additionally, a much larger portion of the clicks originated from a Microsoft OS in the News community which accounts for the larger portion of access from personal computers in this community as well. Clearly, both communities have distinct differences in the hardware methods used to access the information.

In addition to the hardware differences, temporary differences arise as well. As seen in Figure 17, two very different access patterns emerge.

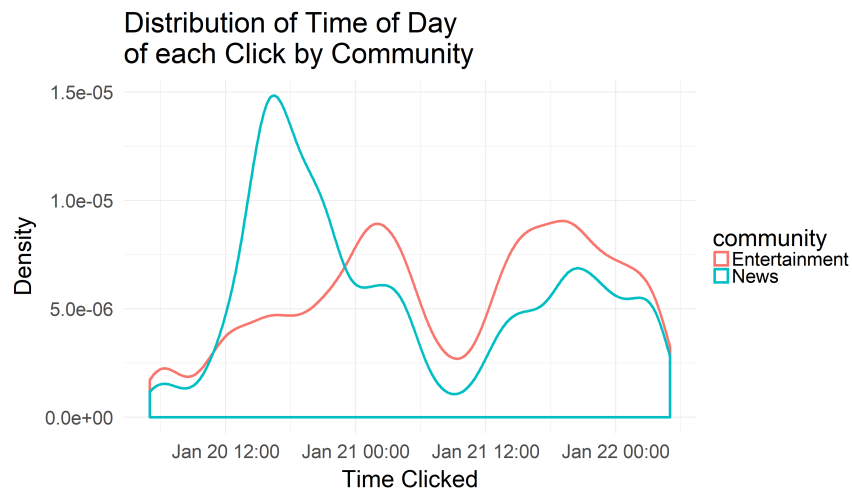


Figure 17. Distribution of Click Times by Community (January 2017)

The Entertainment community follows a very similar click-time distribution to the overall distribution in Figure 8a. This distribution again follows a social-type pattern of slowly increasing click counts during the day to a peak in the late evening followed by a trough in the early morning hours.

This is in sharp contrast to the News community which shows a sharp peak shortly following the beginning of the Presidential inauguration (Noon on 20 January). The total number of clicks coming from each day sharply differed as well with 49,378 clicks (55.6%) originating on January 20th while the remaining 39,722 clicks (44.4%) came

on January 21st. This access pattern follows from Figure 15b which showed that more clicks were related to the inauguration than not.

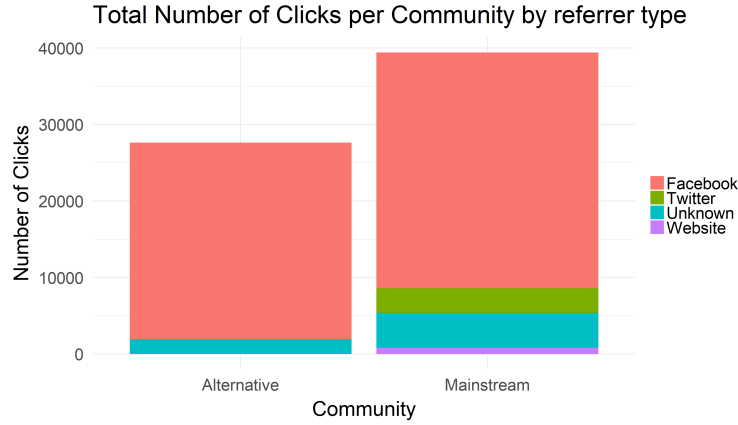
The two identified communities from January 2017 displayed different trends and access patterns relative to each other and to the overall data set. Specifically, the News community featured web domains connected by common users more interested in the Presidential Inauguration. Meanwhile, the Entertainment community was connected by users less interested in the Inauguration and more interested in other information not related to the Inauguration. These differences also manifested in the types of devices used to access the information provided.

4.4.2 February 5-7, 2017

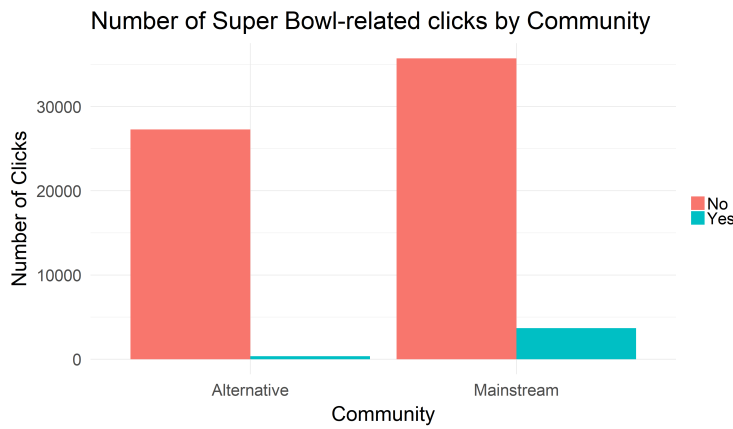
In February, a similar pattern where a very small portion of web domains garners a disproportionate number of web clicks exists as well. The 29 web domains that remained had 67,074 clicks (26.2%) of Charlotte's shortened web link clicks. Additionally the inter-community metrics exhibited similar activities to those in January.

Again, the breakdown of links by referrer type for each community demonstrates that the vast majority of one community, in this case Alternative/Local originates from Facebook while the second community, Mainstream/National has a large Facebook response while showing a wider dispersion of links originating from non-Facebook sites. These other non-Facebook sites include direct clicks from Twitter, Websites, or other unknown locations. One key difference between January and February is the lack of dominance of the event's weekend in February (the Super Bowl).

As seen in Figure 18b, both communities are dominated by non-Super Bowl related links unlike in January where the News community had a preponderance of clicks related to the inauguration. Despite the performance of the Super Bowl for clicks, the Mainstream/National community had more clicks (3,695 clicks/9.4%) regarding



(a) Clicks Per Community by Referrer Type



(b) Referrer Type

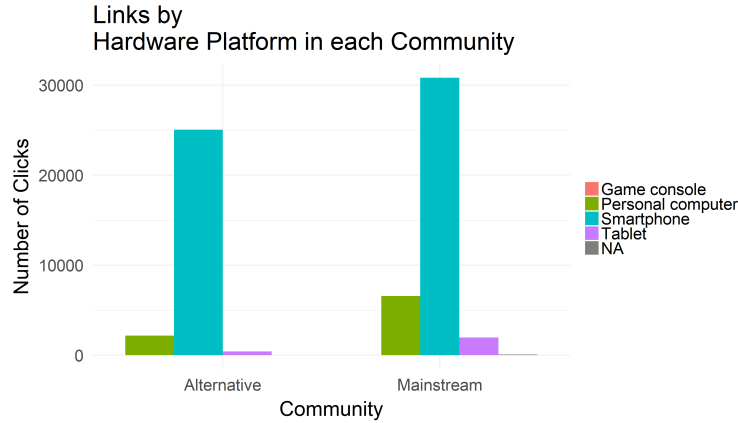
Figure 18. Community Information Measures (February 2017)

the Super Bowl than the Alternative/Local community (352 clicks/1.2%).

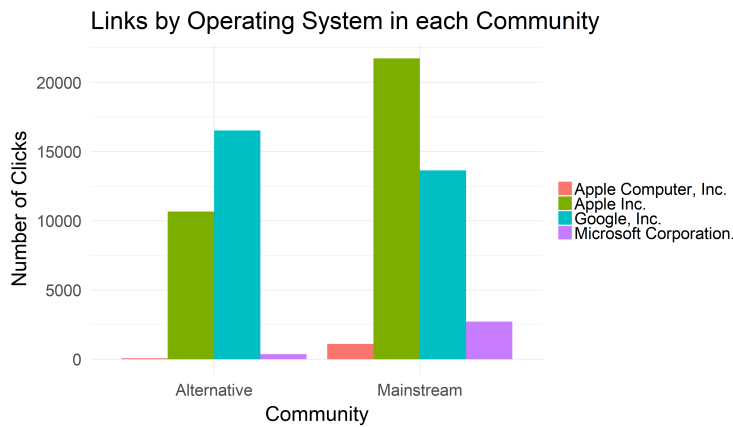
Next, as before, the communities are broken down based on hardware type and OS type and are shown in Figure 19.

Again, there is a clear difference between the two communities in terms of OS type and device. the Mainstream/National community domains tended to be accessed by Apple iPhone devices while the Alternative/Local tended to be accessed by Android OS smartphones.

Finally, the time-of-click distribution is shown in Figure 20. As shown previously, the Super Bowl did not have a noticeable impact on the link-clicking patterns in Charlotte. Both communities exhibit the social-type cyclic pattern with link clicks



(a) Clicks Per Community by Platform Type



(b) Operating System

Figure 19. Access Types by Community (February 2017)

peaking in the evening and reaching a nadir in the early morning. Additionally, both days have approximately the same number of clicks with 31,687 clicks (47.2%) on February 5th and 35,387 clicks (52.7%) on February 6th.

Ultimately, the benefit of this process is the ability to cluster commonly accessed websites together to determine which information users are most interested in learning together. As shown in January 2017, users interested in the inauguration tended to visit multiple web domains that all offered inauguration coverage. Meanwhile, users seemingly not interested in online coverage of the inauguration steered clear and visited web domains that provided content outside of the inauguration. In February, a similar, although much less pronounced, type of effect occurred. While the Super

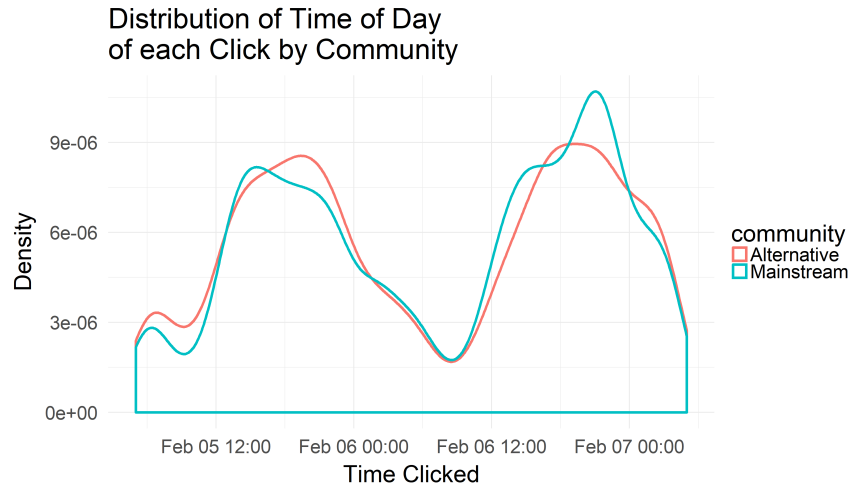


Figure 20. Distribution of Click Times by Community (February 2017)

Bowl did not dominate coverage in either identified community as it did in January, the National/Mainstream coverage covered the national event Super Bowl at a higher rate than the Alternative/Local web sites. After clustering these web domains and, implicitly, the information they offer, various patterns emerged regarding the underlying methods used to access the information. The communities showed differences in proportions in hardware type, referrer types, and operating systems. These patterns are useful in learning about an area and how certain locations access information.

V. Conclusion

5.1 Summary

Governments and companies are fundamentally interested in understanding the actions of citizens and consumers. Market research arose to help answer these questions. Through the use of censuses, surveys and focus groups, researchers were able to gradually better understand how people reacted to brands and various marketing techniques. As information systems became more prevalent, market research methods began to incorporate KDD fundamentals to help identify consumer patterns and learn how to best use these identified patterns. From this application of KDD to market research arose the field of Market Basket Analysis. Market Basket Analysis is concerned with identifying which products are often purchased together. Companies then use the identified association rules to help improve sales for associated products. Market Basket Analysis was then extended by using graph mining techniques to identify rules and clusters (communities) of products that had been previously unidentified.

This research adapted market basket analysis using graph mining techniques and applied the technique to shortened web link data. Market basket analysis was applied to two datasets of shortened web link clicks from Charlotte, North Carolina from the weekend of 20-22 January 2017 (Friday-Sunday) and 5-7 February 2017 (Sunday-Tuesday) to identify which web domains were most commonly accessed together by internet users in Charlotte, North Carolina. From each dataset, two plurality communities were identified. In January, the dataset contained 9,600 unique web domains, and after filtering low-occurring pairs, a total of 21 web domains remained which accounted for 28.5% of all shortened link clicks that weekend. A similar result happened in February. A total of 29 web domains from the original 9,234 unique web

domains remained after filtering. These domains accounted for 26.2% of all shortened link clicks from that weekend.

5.2 Conclusions

First, internet access patterns over the two weekends seem fairly consistent. Despite a three week gap between the two time periods, the number of clicks between the two weekends differed by less than 20%. Additionally, the proportions of the various categories, such as hardware platform or referrer type, examined also exhibited strong similarities between the weekends. This indicates that the users in Charlotte, while accessing different types of information both weekends, were accessing the information in basically the same way. The similarities discovered at the group-level were also observed at the community level. While the two communities were not identical from January to February, a number of web domains remained within the same groupings. For example, the web domains “www.espn.com” and “www.nydailynews.com” were clustered together in both weekends. Alternatively, the web domains “www.wistv.com” and “www.iloveoldschoolmusic.com” were clustered together in both weekends.

Following from this, we see that each community exhibited differences in the typical user profile. For example, in January and February, users accessing the News community or the National/Mainstream community domains tended to access using an iPhone smartphone device while users accessing the Entertainment or Local/Alternative community domains tended to access using an Android smartphone. Additionally, while these communities all preferred Facebook as a referrer, the News and National/Mainstream communities did show a larger spread in alternative referral methods while the Entertainment and Local/Alternative community were almost exclusively accessed via Facebook. This suggests that while some aspects of shortened

web link activity in Charlotte might be volatile, such as the specific topics of links accessed, other aspects are less volatile and are consistent over time and can be used to gain a better understanding of information access patterns of an area.

Finally, the identified communities indicate that the topics of information being accessed differs between communities. The identified communities, from both January and February 2017, show that users in Charlotte co-access websites based off of the type of information provided rather than by other means. For example, in the January News Community, link clicks were dominated by the presidential inauguration coverage (55%) while fewer than 20% of clicks in the entertainment community were about the inauguration. The only connection between web domains is through common users, so users in Charlotte interested in the inauguration were more likely to click to other web domains providing inauguration coverage. Alternatively, users who did not access web domains providing inauguration coverage tended to continue to access other web domains not providing inauguration coverage. Therefore, companies with information that fit a certain category (i.e. news-related) should anticipate that propagating information outside that category might be more difficult than within that category and will have to mitigate these effects if trying to disseminate information to as many people as possible.

Market basket analysis has been used to great effect in other domains such as tangible products. However, this technique can be applied to shortened web links to identify communities of co-accessed web domains and provide meaningful results to market researchers and analysts.

5.3 Future Research

To extend the market basket analysis, two areas should be explored further. This research was limited in scope to scraping actual web links for keywords such as “in-

auguration” or “president”. However, applying text mining techniques, an analyst could access the links and extract the link’s contents. Following this, topic modeling techniques could be used to help identify the predominant topics that each community. This could help provide some more insight into what information is important to an area. Additionally, over time a baseline could be developed and the volatility of information topics could be measured. A second area of research is related to demographic research of an area. This information would be able to be combined with a typical user profile for each identified community to determine demographically which people in an area are accessing which types of information. This information would help provide more understanding of an area and help to determine what the differences in access pattern might be based on age, ethnicity, or income.

Appendix A. Web Click Example Data

Table 7. Web Click Example Data

Feature Name	Example 1	Example 2
user agent	Mozilla/5.0 (Linux; U; Android 4.1.2; en-us; HTC_PN071 Build/JZO54K) AppleWebKit/534.30 (KHTML, like Gecko) Version/4.0 Mobile Safari/534.30	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; MDDR)
accept language	en-US	en-us
country code	US	
geo city name	Anaheim	
global bitly hash	15r91	ifIpBW
geo region	CA	
encoding user bitly hash	10OBm3W	ifIpBW
hash timestamp	1365701422	1302189369
short url cname	j.mp	1.usa.gov
encoding user login	pontifier	bitly
latlong	[33.816101, -117.979401]	
known user	0	0
referring url	direct	http://www.usa.gov/
timestamp	1368832205	1368832207
timezone	America/Los_Angeles	http://answers.usa.gov/system /selfservice.controller? CONFIGURATION=1000& PARTITION_ID=1&CMD= VIEW_ARTICLE&USERTYPE =1&LANGUAGE=en&COUNTRY =US&ARTICLE_ID=11103
long url	http://www.nsa.gov/	

Appendix B. Software Packages Used

Table 8. Software Packages Used

Software used	Version Used	Notes
R[31]	3.3.2	
RStudio[37]	1.1.383	
iGraph[32]	1.0.1	Conducts graph building and community detection
tidyverse[38]	1.2.1	Allows for use of dplyr, ggplot2 packages
Matrix[39]	1.2-12	Accommodates large, sparse matrices
readr[40]	1.1.1	Helps import data quickly into R
ggthemes[41]	3.4.0	Augments plots from ggplot2 to match a common theme
lubridate[42]	1.6	Allows for easier use and calculations of times
dplyr[43]	0.7.4	Allows for easy and fast data manipulation
ggplot2[44]	2.1.0	Builds charts and plots from data

Appendix C. Analysis Code

```
#####  
#Packages needed  
#####  
  
library(igraph)  
library(tidyverse)  
library(Matrix)  
library(readr)  
library(ggthemes)  
library(lubridate)  
  
#####  
# Political Books Test Set  
#####  
  
#####  
# Read in the File  
#####  
  
#Info about file available here:  
#https://networkdata.ics.uci.edu/data.php?id=8  
coords1 ← readRDS("./Toy Examples/coords1.RData")  
graph.file ← "./Toy Examples/polbooks.gml"  
g1 ← igraph::read_graph(file = graph.file, format = "gml")  
  
print(g1)  
  
#####  
# Build iGraph object with new adjacency matrix  
#####  
  
# copurchase.graph ← copurchase.matrix %>%  
#   igraph::graph_from_adjacency_matrix(mode = "undirected",  
#                                       weighted = T,  
#                                       diag = F)  
#  
# print(copurchase.graph)
```

```
#####
# Plot by ground-truth community color
#####

V(g1)$value ← ifelse(V(g1)$value == "l", "blue",
                     ifelse(V(g1)$value == "c",
                           "red",
                           "yellow"))

V(g1)$value

g1

title1 = "Ground Truth co-Purchase Communities"
plot1 ← tkplot(g1, vertex.label = V(g1)$id,
               vertex.color = V(g1)$value,
               #layout = coords1,
               main = "Ground-truth Co-Purchase Community Plot",
               edge.curved = T)

plot(g1, vertex.label = V(g1)$id,
      vertex.color = V(g1)$value,
      layout = coords1,
      main = "Ground-truth Co-Purchase Community Plot",
      edge.curved = T)

coords1 ← tk_coords(plot1)

#####
# SLPA ground-truth community accuracy
#(Iterate through 1000 seeds)
#####
accuracy.labelprop = vector(length = 100)
for (j in 1:100) {
  set.seed(j)
  community.labelprop ← cluster_label_prop(g1)
  membership(community.labelprop)
  (sizes(community.labelprop))
}
```

```

V(g1)$labelprop <- membership(community.labelprop)

for (i in 1:max(membership(community.labelprop))) {
  V(g1)[V(g1)$labelprop==i]$labelprop <-
    V(g1)[V(g1)$labelprop==i]$value %>%
      table %>%
      sort(decreasing = T) %>%
      names() %>% .[1]
}

accuracy.labelprop[j] <-
  ((V(g1)$value == V(g1)$labelprop) %>%
    as.vector %>% sum) /
    length(V(g1)) * 100

}

V(g1)$labelprop

mean(accuracy.labelprop)
sd(accuracy.labelprop)

plot(g1, vertex.label = V(g1)$id,
      vertex.color = V(g1)$labelprop,
      layout= coords1,
      main = "Speaker-Listener Propagation",
            "Algorithm Community Plot",
      edge.curved = T)

ggplot() + geom_line(aes(x = 1:length(accuracy.labelprop),
                          y = accuracy.labelprop)) +
  scale_y_continuous(breaks = seq(50,100,10),
                     limits = c(50,100)) +
  xlab("Random Seed") +
  ylab("Accuracy") +
  theme_minimal(base_size = 24)

```

```
#####
# WalkTrap ground-truth community accuracy
#####

accuracy.walktrap = vector(length = 100)
for (j in 1:100) {
  set.seed(j)
  community.walktrap ← cluster_walktrap(g1)
  membership(community.walktrap)
  (sizes(community.walktrap))

  V(g1)$walktrap ← membership(community.walktrap)

  for (i in 1:max(membership(community.walktrap))) {
    V(g1)[V(g1)$walktrap==i]$walktrap ←
      V(g1)[V(g1)$walktrap==i]$value %>%
      table %>%
      sort(decreasing = T) %>%
      names() %>% .[1]
  }

  accuracy.walktrap[j] ← ((V(g1)$value == V(g1)$walktrap) %>%
    as.vector %>% sum) /
    length(V(g1)) * 100
}

mean(accuracy.walktrap)
sd(accuracy.walktrap)

ggplot() + geom_line(aes(x = 1:length(accuracy.walktrap),
                          y = accuracy.walktrap)) +
  scale_y_continuous(breaks = seq(50,100,10),
                     limits = c(50,100)) +
  xlab("Random Seed") +
  ylab("Accuracy") +
  theme_minimal(base_size = 24)
```

```
#####
# Plot by ground-truth community color - WalkTrap
#####

plot(g1, vertex.label = V(g1)$id,
      vertex.color = V(g1)$walktrap,
      layout= coords1,
      edge.curved = T,
      main = "WalkTrap Algorithm Community Plot")

#####
# Accuracy by algorithm
#####

accuracy.matrix <- data.frame(row.names = c("SLPA",
                                             "WalkTrap"),
                              stringsAsFactors = F)
colnames(accuracy.matrix) <- c("Mean", "SD")

accuracy.matrix[1,1] <- mean(accuracy.labelprop)
accuracy.matrix[1,2] <- sd(accuracy.labelprop)
accuracy.matrix[2,1] <- mean(accuracy.walktrap)
accuracy.matrix[2,2] <- sd(accuracy.walktrap)
accuracy.matrix

#####
# January Analysis
#####

#####
#Functions that do the analysis
#####

loadWebClicks <- function(data.location) {

  data <- readr::read_csv(data.location, progress = F)

  column.names <- colnames(data)

  nulled.names <- c("APPLICATIONPROTOCOL_REFERRER",
                    "AUTHORITY_URI_REFERRER_REVERSE",
                    "AUTHORITY_URI_REVERSE",
```

```

        "COUNTRYCODE_SRC" ,
        "DOMAINNAME_CUSTOM" ,
        "FIELDVALUE_HTTPFORMELEMENTS" ,
        "FIELDVALUE_HTTPFORMELEMENTS_REFERRER" ,
        "FIELD_HTTPFORMELEMENTS" ,
        "FIELD_HTTPFORMELEMENTS_REFERRER" ,
        "FRAGMENT_URI" ,
        "PORT_DST" ,
        "VALUE_HTTPFORMELEMENTS" ,
        "VALUE_HTTPFORMELEMENTS_REFERRER" ,
        "Visibility" )
data[,column.names %in% nulled.names] ← NULL
data ← data[!data$Id=="Id",] #Remove the appended column
#headers from the smaller files

#data$OSFAMILY_OPERATINGSYSTEM ← as.factor

return(data)

} #LOADS WeBCLICKS FROM CSV
buildAdjacencyMatrix ← function(data) {

  data ← data[ data$AUTHORITY_URI %in%
               names(which(
                 table(data$AUTHORITY_URI) > 1)), ]
  data %>% nrow %>% print
  data ← data[!data$HTTPUSERAGENT == "-",]
  data ← data[!data$HTTPUSERAGENT %>% is.na,]

  data ← data %>% mutate(UserID = paste(HTTPUSERAGENT,
                                         LATITUDE_SRC,
                                         LONGITUDE_SRC,
                                         sep = "_"))
  data$UserID ← data$UserID %>% as.factor %>% as.numeric
  data$UserID %>% max %>% print
  print("The number of rows remaining is: \n")
  data %>% nrow %>% print

  #Build the bipartite graph
  g1 ← data[,c("UserID", "AUTHORITY_URI")] %>%
    as.matrix %>% graph_from_edgelist(directed = F)
  #pull out the adjacency matrix

```

```

adjacency.matrix ← get.adjacency(g1, type = "both",
                                sparse = T)

#Return the row names as users and column names as websites
matrix.rownames ← rownames(adjacency.matrix)

truth ← grepl(pattern = "^[[:space:]]{,4}[[:digit:]]{,5}$",
              x = matrix.rownames, ignore.case = T)

#this selects rows as user names and columns as websites
adjacency.matrix ← adjacency.matrix[matrix.rownames[truth],
                                   matrix.rownames[!truth]]

#changes everything to a binary presence matrix
adjacency.matrix ← Matrix((adjacency.matrix> 0)+0,
                          sparse = T)

return(adjacency.matrix)

} #GETS THE ADJACENCY MATRIX
filterCopurchaseMatrix ← function(copurchase.matrix, n = 3,
                                filterlevel = 0.05){

  time1 ← Sys.time()
  filtervalue ← getTopNvalues(copurchase.matrix) %>%
    mean()*filterlevel
  print(filtervalue)

  time2 ← Sys.time()

  #print(time2-time1)
  websites ← rownames(copurchase.matrix)

  mat ← sparseToMatrix(copurchase.matrix) %>% as.data.frame

  mat ← mat %>% filter(x > filtervalue)
  #print(mat)
  #print(max(mat))
  #print(length(websites))

  time3 ← Sys.time()

```

```

#print(time3-time2)

filteredcopurchase ← sparseMatrix(i = mat[,1],
                                   j = mat[,2],
                                   x = mat[,3],
                                   dims = c(length(websites),
                                             length(websites)))

#print(summary(filteredcopurchase))
rownames(filteredcopurchase) ← websites
colnames(filteredcopurchase) ← websites

time4 ← Sys.time()

#print(time4-time3)

diag(filteredcopurchase) ← 0

filteredcopurchase ← filteredcopurchase[rowSums(
  filteredcopurchase) > 0, colSums(filteredcopurchase) > 0]
time5 ← Sys.time()
#print(time5-time4)

return(filteredcopurchase)

} #APPLIES A FILTER
getCommunities ← function(copurchase.matrix) {

  copurchasegraph ← graph_from_adjacency_matrix(
    copurchase.matrix, mode = "undirected",
    diag = F, weighted = T)
  community.walktrap ← cluster_walktrap(graph = copurchasegraph,
                                         weights = E(copurchasegraph)$weight,
                                         steps = 4,
                                         merges = T,
                                         modularity = T,
                                         membership = T)

  return(community.walktrap)

} #RETURNS COMMUNITIES

```

```

sparseToMatrix ← function(x) (Matrix ←
                                as.matrix(summary(x)))
#Subfunctions needed in other main functions

getTopNvalues ← function(x, n = 3) {

  matrix ← sparseToMatrix(x)

  values ← matrix[, "x"] %>% sort(decreasing = T)
  values ← values[1:n]
  return(values)
} #Another necessary subfunction

#####
# Begin Exploratory Data Analysis
#####

#id File location
data.location ← "../Data/Data/JanCharlotte.csv"

#Load in the data
JanCharlotte ← loadWebClicks(data.location = data.location)

#Remove ads

JanCharlottenoads ← JanCharlotte %>%
  filter(AUTHORITY_URI != "t.co") %>%
  filter(AUTHORITY_URI != "jamdex.com") %>%
  filter(AUTHORITY_URI != "awesome-cool-music.blogspot.com") %>%
  filter(AUTHORITY_URI != "greatmusicstreaming.blogspot.com") %>%
  filter(AUTHORITY_URI != "n2adshostnet.com") %>%
  filter(AUTHORITY_URI != "prpops.com") %>%
  filter(AUTHORITY_URI != "best-streaming-music.blogspot.com") %>%
  filter(AUTHORITY_URI != "www.dropbox.com") %>%
  filter(AUTHORITY_URI != "static-v2.astar.mobi") %>%
  filter(AUTHORITY_URI != "dl.dropboxusercontent.com") %>%
  filter(AUTHORITY_URI != "interactive.tegna-media.com") %>%
  filter(!AUTHORITY_URI %in% grep(pattern = "adf.ly",
                                x = JanCharlotte$AUTHORITY_URI, value = T)) %>%
  filter(!AUTHORITY_URI %in% grep(pattern = "getgiftcards.org",
                                x = JanCharlotte$AUTHORITY_URI, value = T)) %>%
  filter(!AUTHORITY_URI %in% grep(pattern = "weebly.com",

```

```

      x = JanCharlotte$AUTHORITY_URI, value = T)) %>%
filter(!AUTHORITY_URI %in% grep(pattern = "$<U.*",
      x = JanCharlotte$AUTHORITY_URI, value = T)) %>%
filter(!AUTHORITY_URI %in% grep(pattern = "myautodj.com",
      x = JanCharlotte$AUTHORITY_URI, value = T)) %>%
filter(!AUTHORITY_URI %in% grep(pattern = "rackcdn.com",
      x = JanCharlotte$AUTHORITY_URI, value = T)) %>%
filter(!AUTHORITY_URI %in% grep(pattern = "voluumtrk.com",
      x = JanCharlotte$AUTHORITY_URI, value = T)) %>%
filter(!AUTHORITY_URI %in% grep(pattern = "blogspot.com",
      x = JanCharlotte$AUTHORITY_URI, value = T)) %>%
filter(!AUTHORITY_URI %in% grep(pattern = "api.",
      x = JanCharlotte$AUTHORITY_URI, value = T)) %>%
filter(!AUTHORITY_URI %in% grep(pattern = "click.",
      x = JanCharlotte$AUTHORITY_URI, value = T)) %>%
filter(!AUTHORITY_URI %in% grep(pattern = "ad.doubleclick",
      x = JanCharlotte$AUTHORITY_URI, value = T)) %>%
filter(!AUTHORITY_URI %in% grep(pattern = "yakidee.org",
      x = JanCharlotte$AUTHORITY_URI, value = T)) %>%
filter(!AUTHORITY_URI %in% grep(pattern = "app.",
x = JanCharlotte$AUTHORITY_URI, value = T)) %>%
filter(!AUTHORITY_URI %in% grep(pattern =
" [[:digit:]]{,3}[[:punct:]]{,1}[[:digit:]]{,3}[[:punct:]]",
" {,1}[[:digit:]]{,3}[[:punct:]]{,1}[[:digit:]]{,3}",
x = JanCharlotte$AUTHORITY_URI, value = T)) %>%
filter(!AUTHORITY_URI_REFERRER %in% grep(pattern =
" [[:digit:]]{,3}[[:punct:]]{,1}[[:digit:]]{,3}[[:punct:]]",
" {,1}[[:digit:]]{,3}[[:punct:]]{,1}[[:digit:]]{,3}",
x = JanCharlotte$AUTHORITY_URI_REFERRER, value = T)) %>%
filter(!AUTHORITY_URI_REFERRER %in%
  c("video-promo.net",
    "aptrk.com",
    "www.IsraeLIVE.org",
    "watchonlinevideos.org"))

```

```

tomatch <- c("trump", "Donald",
             "donald", "Trump",
             "politics", "inauguration")

```

```

JanCharlottenoads <- JanCharlottenoads %>%
mutate(referrer = ifelse(AUTHORITY_URI_REFERRER %in%

```

```

  grep(pattern = "facebook",
x = JanCharlottenoads$AUTHORITY_URI_REFERRER, value = T),
"Facebook",
  ifelse(AUTHORITY_URI_REFERRER %in%
    grep(pattern = "t.co",
x = JanCharlottenoads$AUTHORITY_URI_REFERRER,
value = T), "Twitter",
  ifelse(AUTHORITY_URI_REFERRER %in%
    grep(pattern = "Direct",
x = JanCharlottenoads$AUTHORITY_URI_REFERRER,
value = T), "direct",
  ifelse(is.na(AUTHORITY_URI_REFERRER),
    "Unknown", "Website")))) %>%

mutate(presrelated = ifelse(URL_REQUEST %in%
  grep(pattern = paste(tomatch, collapse = "|"),
    value = T, x = JanCharlottenoads$URL_REQUEST),
    "Yes", "No"))

```

```

JanCharlottenoads <- JanCharlottenoads %>% as_tibble()
JanCharlottenoads$referrer <- JanCharlottenoads$referrer %>%
  as.factor()
JanCharlottenoads$VENDORNAME_OPERATINGSYSTEM <-
  JanCharlottenoads$VENDORNAME_OPERATINGSYSTEM %>% as.factor()
JanCharlottenoads$TYPE_HARDWAREPLATFORM <-
  JanCharlottenoads$TYPE_HARDWAREPLATFORM %>% as.factor()
JanCharlottenoads$TIMESTAMP_INIT <-
  as.POSIXct(JanCharlottenoads$TIMESTAMP_INIT) %>% ymd_hms()
JanCharlottenoads$DURATION_FROMCLICKTOCREATION <-
  JanCharlottenoads$DURATION_FROMCLICKTOCREATION %>%
  as.numeric %>% duration(units = "seconds")

```

```

#####
# Exploratory Data Analysis – Plots
#####

```

```

source <- "../Plots/JanCharlotte/EDA"

```

```

JanCharlottenoads %>%

```

```

ggplot() +
geom_bar(aes(x = referrer, fill = referrer),
          position = "dodge", na.rm = F) +
xlab("Referrer") +
ylab("Total Number of Clicks") +
ggtitle("Clicks by Referrer Type") +
guides(fill=F) +
scale_y_continuous(breaks = c(0, 25000, 50000, 75000,
                              100000, 125000, 150000),
                  labels = c("0", "25,000", "50,000",
                              "75,000", "100,000",
                              "125,000", "150,000")) +
scale_x_discrete(labels = c("Facebook", "Unknown",
                             "Twitter", "Website"),
                 limits = c("Facebook", "Unknown",
                             "Twitter", "Website")) +
theme_minimal(base_size = 18)

ggsave("Total Number of Clicks by Referrer Type.png",
       last_plot(), path = source)

```

```

JanCharlottenoads %>%
  ggplot() + geom_density(aes(x = TIME_CLICKED,
                              group = referrer,
                              color = referrer)) +
  ggtitle("Time Click Distribution of links by referrer
type") +
  xlab("Time Clicked") +
  ylab("Density") +
  guides(fill=guide_legend(title="Referrer")) +
  theme_minimal(base_size = 18)
ggsave("Time Click Distribution of links by referrer
type.png",
       plot = last_plot(), path = source)

```

```

#plot time-generation distribution of president links
JanCharlottenoads %>%
  filter(TIMESTAMP_INIT > "2017-01-20 00:00:01") %>%
  ggplot() + geom_density(aes(x = TIMESTAMP_INIT,
                              group = referrer,
                              color = referrer)) +

```

```

xlab("Time Link Generated") +
ylab("Proportion of Links") +
ggtitle("Distribution of Link Generation Times") +
guides(fill=guide_legend(title=NULL)) +
theme_minimal(base_size = 18)

ggsave("Distribution of Link Generation Times.png",
      plot = last_plot(), path = source)

#plot time-generation DURATION of president links
JanCharlottenoads %>%
  filter(TIMESTAMP_INIT > "2017-01-20 00:00:01") %>%
  ggplot() + geom_density(aes(x =
    (as.numeric(DURATION_FROMCLICKTOCREATION)/
      as.numeric(dhours(x = 1))),
    group = referrer, color = referrer)) +
  xlab("Time between Creation and Click (hrs)") +
  ylab("Proportion of Links") +
  ggtitle("Distribution of time between \nLink Creation
and Link Click") +
  guides(fill=guide_legend(title=NULL)) +
  theme_minimal(base_size = 18)

ggsave("Distribution of time between Link Creation and
Link Click.png",
      plot = last_plot(), path = source)

JanCharlottenoads %>%
  filter(VENDORNAME_OPERATINGSYSTEM %in%
names(which(table(
JanCharlottenoads$VENDORNAME_OPERATINGSYSTEM) >
500))) %>%
  ggplot() + geom_bar(aes(x = VENDORNAME_OPERATINGSYSTEM,
    fill = VENDORNAME_OPERATINGSYSTEM),
    position = "dodge", na.rm = T) +
  ggtitle("Clicks by Operating System Type
\n(OS type with >500 total clicks)") +
  ylab("Count") +
  scale_x_discrete(limits = c("Apple Inc.", "Google, Inc.",
    "Microsoft Corporation.",

```

```

        "Apple Computer, Inc."),
name = "Operating System Vendor Name",
labels = c("Apple (iOS)", "Google (Android)",
           "Microsoft", "Apple Computer")) +
scale_y_continuous(
breaks = c(0, 25000, 50000, 75000, 100000,
           125000, 150000),
labels = c("0", "25,000", "50,000",
           "75,000", "100,000",
           "125,000", "150,000")) +
guides(fill=F) +
theme_minimal(base_size = 18)

ggsave("Clicks by Operating System Type.png",
       plot = last_plot(), path = source)

JanCharlottenoads %>%
  filter(TYPE_HARDWAREPLATFORM %in%
names(which(table
(JanCharlottenoads$TYPE_HARDWAREPLATFORM) > 500))) %>%
ggplot() + geom_bar(aes(x = TYPE_HARDWAREPLATFORM,
                        fill = TYPE_HARDWAREPLATFORM),
                    position = "dodge", na.rm = T) +
ggtitle("Clicks by Hardware Platform
\n(Platform with >500 total clicks)") +
ylab("Count") +
scale_x_discrete(labels = c("Smartphone", "PC",
"Tablet"),
                  name = "Hardware Platform",
                  limits = c("Smartphone",
                             "Personal computer", "Tablet")) +
guides(fill= F) +
theme_minimal(base_size = 18)

ggsave("Clicks by Hardware Platform.png",
       plot = last_plot(), path = source)

```

```

#####
# Inauguration Related Exploratory Data Analysis

```

#####

```
JanCharlottenoads %>%
  ggplot() +
  geom_bar(aes(x = referrer, fill = presrelated),
           position = "fill", na.rm = F) +
  xlab("Referrer") +
  ylab("Proportion") +
  ggtitle("Proportion of Inauguration-related Clicks
\nby Referrer Type") +
  guides(fill=guide_legend(title="Inauguration-
\nrelated")) +
  theme_minimal(base_size = 18)
```

```
ggsave("Clicks by Referrer Type related to
Inauguration.png",
       plot = last_plot(), path = source)
```

```
JanCharlottenoads %>%
  filter(VENDORNAME_OPERATINGSYSTEM %in%
         names(which(
           table(JanCharlottenoads$
                 VENDORNAME_OPERATINGSYSTEM) > 500))) %>%
  ggplot() +
  geom_bar(aes(x = VENDORNAME_OPERATINGSYSTEM,
              fill = presrelated), position = "fill",
          na.rm = T) +
  ggtitle("Proportion of Inauguration-related
\nClicks by Operating System") +
  scale_x_discrete(name = "Operating System Vendor Name") +
  ylab("Proportion") +
  guides(fill=guide_legend(title="Inauguration-
\nrelated")) +
  theme_minimal(base_size = 16)
```

```
ggsave("Proportion of Clicks by Operating System
related to OS Type.png",
       plot = last_plot(), path = source)
```

```

JanCharlottenoads %>%
  filter(TYPE_HARDWAREPLATFORM %in%
          names(which(
            table(JanCharlottenoads$TYPE_HARDWAREPLATFORM)
              > 500))) %>%
  ggplot() +
  geom_bar(aes(x = TYPE_HARDWAREPLATFORM,
               fill = presrelated),
           position = "fill",
           na.rm = T) +
  ggtitle("Proportion of Inauguration-related Clicks
\nby Hardware Platform") +
  scale_x_discrete(name = "Hardware Platform") +
  ylab("Proportion") +
  guides(fill=guide_legend(title="Inauguration-\nrelated")) +
  theme_minimal(base_size = 18)

ggsave("Proportion of Inauguration-related Clicks by
Hardware Platform.png",
       plot = last_plot(), path = source)

```

```

#####
# Community Detection Method
#####

```

```

adjacencymatrix <- buildAdjacencyMatrix(JanCharlottenoads)
#adjacencymatrix %>% dim

copurchase.matrix <- t(adjacencymatrix) %*% adjacencymatrix

```

```

#####
# Begin Filtering / Plot Results of varying filter levels
#####

```

```

min.filter <- 0.01
max.filter <- 0.10
filter.step <- 0.01

```

```

filters ← seq(from = min.filter , to = max.filter ,
              by = filter.step)
filteredcommunities ← list()

#initialize data frame to begin plotting
iterated.results ← data.frame('Filter Level' =
                              seq(from = min.filter ,
                                  to = max.filter ,
                                  by = filter.step),
                              'Number of Communities' =
                                seq(from = min.filter ,
                                    to = max.filter ,
                                    by = filter.step),
                              'Modularity' =
                                seq(from = min.filter ,
                                    to = max.filter ,
                                    by = filter.step),
                              'Number of Domains' =
                                seq(from = min.filter ,
                                    to = max.filter ,
                                    by = filter.step))

for (i in seq(1,length(filters))) {

  copurchase.matrix1 ← filterCopurchaseMatrix(
    copurchase.matrix ,
                                n = 3,
                                filterlevel = filters[i])
  JanCharlotteCommunities ← getCommunities(copurchase.matrix1)
  iterated.results$Modularity[i] ←
  JanCharlotteCommunities %>% modularity()
  iterated.results$'Number of Communities'[i] ←
    JanCharlotteCommunities %>% sizes %>% nrow
  iterated.results$'Number of Domains'[i] ←
    JanCharlotteCommunities %>% sizes %>% sum
  filteredcommunities[[i]] ← JanCharlotteCommunities

}

#####
source ← ". /Plots /JanCharlotte/Community"

```

```

iterated.results %>%
  ggplot() + geom_line(aes(x = 'Filter Level',
    y = 'Modularity')) +
  ggtitle("Community Modularity Score as Filter Changes") +
  scale_x_continuous(breaks = filters, labels = filters) +
  theme_minimal(base_size = 18)

```

```

ggsave(filename = "Community Modularity Score
as Filter Changes.png",
  path = source, plot = last_plot())

```

```

iterated.results %>%
  ggplot() + geom_line(aes(x = 'Filter Level',
    y = 'Number of Domains')) +
  ggtitle("Number of domains remaining as Filter Changes") +
  scale_x_continuous(breaks = filters, labels = filters) +
  theme_minimal(base_size = 18)

```

```

ggsave(filename = "Number of domains remaining as
Filter Changes.png",
  path = source, plot = last_plot())

```

```

iterated.results %>%
  ggplot() + geom_line(aes(x = 'Filter Level',
    y = 'Number of Communities')) +
  ggtitle("Number of Identified
Communities as Filter Changes") +
  scale_x_continuous(breaks = filters, labels = filters) +
  theme_minimal(base_size = 18)

```

```

ggsave(filename = "Number of Identified Communities
as Filter Changes.png",
  path = source, plot = last_plot())

```

```
#####
# Investigate the communities
#####

JanCharlotteCommunities ← filteredcommunities [[9]]

Community.sizes ← JanCharlotteCommunities %>% sizes %>%
  sort(decreasing = T) %>% as.data.frame
Community.sizes %>%
  ggplot() + geom_line(aes(x = 1:nrow(.),
                           y = Freq,
                           shape = "circle")) +
  geom_point(aes(x = 1:nrow(.), y = Freq)) +
  ggtitle("Number of Domains per Identified Community") +
  xlab("Community Number") +
  ylab("Number of Domains in Community") +
  scale_x_continuous(labels = 1:nrow(Community.sizes),
                     breaks = 1:nrow(Community.sizes)) +
  theme_minimal(base_size = 18)

  ggsave(filename = "Number of Domains per
    Identified Community.png",
          path = source, plot = last_plot())

names ← append(JanCharlotteCommunities [[1]] %>% as.vector(),
               JanCharlotteCommunities [[2]] %>% as.vector()) %>%
  append(., JanCharlotteCommunities [[1]] %>% as.vector())

JanCharlottereduced ← JanCharlottenoads %>%
  filter(AUTHORITY_URI %in% names)

JanCharlottereduced ← JanCharlottereduced %>%
  mutate(community = ifelse(AUTHORITY_URI %in%
                            JanCharlotteCommunities [[1]],
                            "Entertainment",
                            ifelse(AUTHORITY_URI %in% JanCharlotteCommunities [[2]],
                                   "News", 3)))
```

```
JanCharlottereduced$community <- JanCharlottereduced$community
%>% as.factor
```

```
JanCharlottereduced1 <- JanCharlottereduced[
  ! duplicated(JanCharlottereduced$URL_REQUEST),] %>%
  filter(community == "News") %>%
  select(AUTHORITY_URI) %>% table %>%
  sort(decreasing = T) %>%
  as.data.frame
```

```
JanCharlottereduced1 %>%
  ggplot() +
  geom_line(aes(x = 1:nrow(JanCharlottereduced1), y = Freq)) +
  geom_point(aes(x = 1:nrow(JanCharlottereduced1), y = Freq)) +
  xlab("Website") +
  ylab("Number of unique URLs") +
  scale_x_continuous(breaks = 1:nrow(JanCharlottereduced1)) +
  ggtitle("Number of Unique URLs per Domain
  Visited in News Community") +
  theme_minimal()
```

```
ggsave(filename = "Number of Unique URLs
per Domain Visited in News",
  path = source, plot = last_plot())
```

```
JanCharlottereduced2 <- JanCharlottereduced[
  ! duplicated(JanCharlottereduced$URL_REQUEST),] %>%
  filter(community == "Entertainment") %>%
  select(AUTHORITY_URI) %>% table %>%
  sort(decreasing = T) %>%
  as.data.frame
```

```
JanCharlottereduced2 %>%
  ggplot() +
  geom_line(aes(x = 1:nrow(JanCharlottereduced2), y = Freq)) +
  geom_point(aes(x = 1:nrow(JanCharlottereduced2), y = Freq)) +
  xlab("Website") +
  ylab("Number of unique URLs") +
```

```

scale_x_continuous(breaks = 1:nrow(JanCharlottereduced2)) +
ggtitle("Number of Unique URLs per Domain
Visited in Entertainment Community") +
theme_minimal()

ggsave(filename = "Number of Unique URLs
per Domain Visited in Entertainment",
        path = source, plot = last_plot())

#####
# Community Plots
#####

JanCharlottereduced %>% filter(referrer != "direct") %>%
  ggplot() +
  geom_bar(aes(x = community, fill = referrer),
            position = "stack", na.rm = T) +
  xlab("Community") +
  ylab("Number of Clicks") +
  ggtitle("Total Number of Clicks per Community
by referrer type") +
  guides(fill=guide_legend(title=NULL)) +
  theme_minimal(base_size = 18)

  ggsave("Total Number of Clicks per Community
by referrer type.png",
        last_plot(), path = source)

#Plot plot communities 1 and 2 filled relative
#to presidential related links

JanCharlottereduced %>%
  filter(referrer != "direct") %>%
  ggplot() + geom_bar(aes(x = community, fill = presrelated),
                      position = "dodge", na.rm = T) +
  ylab("Number of Clicks") +
  xlab("Community") +

```

```

ggtitle("Number of Inauguration-related clicks
by Community") +
guides(fill=guide_legend(title=NULL)) +
theme_minimal(base_size = 18)

ggsave("Inauguration-related Links by referrer
type in each Community.png",
      last_plot(), path = source)

#Plot inauguration related links by community
#filled by referrer type

JanCharlottereduced %>%
  #filter(presrelated == "Yes") %>%
  filter(referrer != "direct") %>%
  ggplot() + geom_bar(aes(x = community, fill = referrer),
                      position = "dodge", na.rm = T) +
  ylab("Number of Clicks") +
  xlab("Community") +
  ggtitle("Links by \nReferrer type in each Community") +
  guides(fill=guide_legend(title=NULL)) +
  theme_minimal(base_size = 18)

ggsave("Links by Referrer type in each Community.png",
      last_plot(), path = source)

#Plot inauguration-related links by community
#filled by operating system

JanCharlottereduced %>%
  filter(VENDORNAME_OPERATINGSYSTEM !=
"Canonical Ltd.") %>%
  filter(VENDORNAME_OPERATINGSYSTEM !=
"Nintendo of America Inc.") %>%
  filter(VENDORNAME_OPERATINGSYSTEM !=
"Sony Computer Entertainment") %>%
  #filter(presrelated == "Yes") %>%
  ggplot() + geom_bar(aes(x = community,
                          fill = VENDORNAME_OPERATINGSYSTEM),
                      position = "dodge", na.rm = T) +
  ggtitle("Links by Operating System in each Community") +

```

```

xlab("Operating System Type") +
ylab("Number of Clicks") +
guides(fill=guide_legend(title=NULL)) +
theme_minimal(base_size = 18)

ggsave("Links by Operating System in each Community.png",
       plot = last_plot(), path = source)

#plot inauguration related links by community
#filled by hardware type

JanCharlottereduced %>%
  #filter(presrelated == "Yes") %>%
  ggplot() + geom_bar(aes(x = community,
                        fill = TYPE_HARDWAREPLATFORM),
                    position = "dodge", na.rm = T) +
  ggtitle("Links by \nHardware Platform in each Community") +
  xlab("Hardware Platform") +
  ylab("Number of Clicks") +
  guides(fill=guide_legend(title=NULL)) +
  theme_minimal(base_size = 18)

ggsave("Links by Hardware Platform in each Community.png",
       plot = last_plot(), path = source)

#Plot Community 2 (news!) links by hardware
#type when user accesses via Facebook

JanCharlottereduced %>%
  filter(community == "News") %>%
  filter(TYPE_HARDWAREPLATFORM != "Game console") %>%
  filter(referrer == "Facebook") %>%
  ggplot() + geom_bar(aes(x = TYPE_HARDWAREPLATFORM,
                        fill = TYPE_HARDWAREPLATFORM),
                    position = "dodge", na.rm = T) +
  xlab("Hardware Platform") +
  ylab("Number of Clicks") +
  ggtitle("Hardware Platform for News users

```

```

    \naccessing web sites through Facebook") +
    guides(fill = F) +
    theme_minimal(base_size = 18)
ggsave("Hardware Platform for News users
accessing web sites through Facebook.png",
      plot = last_plot(), path = source)

#Plot Community 2 (news!) links by hardware
#type when user accesses not via Facebook

JanCharlottereduced %>%
  filter(community == "News") %>%
  filter(referrer != "Facebook") %>%
  #filter(AUTHORITY_URI != "www.cnn.com") %>%
  filter(TYPE_HARDWAREPLATFORM != "Game console") %>%
  ggplot() + geom_bar(aes(x = TYPE_HARDWAREPLATFORM,
                        fill = TYPE_HARDWAREPLATFORM),
                    position = "dodge", na.rm = T) +
  guides(fill = F) +
  ggtitle("Hardware Platform for News \nusers
#accessing websites",
"other \nthan through Facebook") +
  xlab("Hardware Platform") +
  ylab("Number of Clicks") +
  theme_minimal(base_size = 18)

ggsave("HHardware Platform for News users accessing",
"websites other than through Facebook.png",
plot = last_plot(), path = source)

```

#Plot time-distribution by community

```

JanCharlottereduced %>%

  ggplot() + geom_density(aes(x = TIME_CLICKED,
                              group = community),

```

```

                                color = community)) +
ggtitle("Distribution of Time of Day
\nof each Click by Community") +
xlab("Time Clicked") +
ylab("Density") +
guides(fill=guide_legend(title=NULL)) +
theme_minimal(base_size = 18)

ggsave("Distribution of Time of Day of
each Click by Community.png",
      plot = last_plot(), path = source)

```

Bibliography

1. M. Kaur and S. Kang, "Market Basket Analysis: Identify the Changing Trends of Market Data Using Association Rule Mining," *Procedia Computer Science*, vol. 85, pp. 78–85, 2016.
2. M. Yadav and Z. Rahman, "Measuring consumer perception of social media marketing activities in e-commerce industry: Scale development & validation," *Telematics and Informatics*, 2017.
3. D. Antoniadis, I. Polakis, G. Kontaxis, E. Athanasopoulos, S. Ioannidis, E. Markatos, and T. Karagiannis, "we.b: The web of short URLs," in *20th International World Wide Web Conference*, pp. 715–724, 2011.
4. S. Chhabra, A. Aggarwal, F. Benevenuto, and P. Kumaraguru, "Phi.sh/\$oCiaL," *Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference on - CEAS '11*, pp. 92–101, 2011.
5. E. A. Vella, B. Bartlett, W. Hammersley, J. Kenton, and C. Nugent, *Cost-Effective Market Analysis*. Grants Pass, OR: The Oasis Press, 1989.
6. R. M. Groves, F. J. Fowler Jr, M. P. Couper, J. M. Lepkowski, E. Singer, and R. Tourangeau, *Survey Methodology*. Hoboken, NJ: John Wiley & Sons Inc, 2004.
7. R. Krueger, *Focus Groups: A Practical Guide for Applied Research*. Newbury Park, CA: SAGE Publications Inc., 1988.
8. T. Macer and S. Wilson, "Observations from 12 years of an annual market research technology survey," *International Journal of Market Research*, vol. 59, no. 2.
9. M. Teresa, P. Melo, B. Tiago, and J. M. Cristóvão Veríssimo, "Digital marketing and social media: Why bother?," *Business Horizons*, vol. 57, pp. 703–708, 2014.
10. M. Oxley and B. Light, "Research 2.0: Engage or give up the ghost?," *International Journal of Market Research*, vol. 51, no. 2, pp. 153–161, 2009.
11. G. Baltas, "Determinants of internet advertising effectiveness: an empirical study," *International Journal of Market Research*, vol. 45, no. 4, pp. 505–513, 2003.
12. Y.-L. Lin and Y.-W. Chen, "Effects of ad types, positions, animation lengths, and exposure times on the click-through rate of animated online advertisings," *Computers & Industrial Engineering*, vol. 57, pp. 580–591, 2009.
13. J. S. Ilfeld and R. S. Winer, "Generating website traffic," *Journal of Advertising Research*, vol. 42, no. 5, pp. 49–61, 2002.

14. I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. Ullah Khan, "The rise of "big data" on cloud computing: Review and open research issues," *Information Systems*, vol. 47, pp. 98–115, 2015.
15. A. A. Raorane, R. Kulkarni, and B. D. Jitkar, "Association Rule – Extracting Knowledge Using Market Basket Analysis," *Research Journal of Recent Sciences*, vol. 1, no. 2, pp. 19–27, 2012.
16. "Advances in Data Mining: Applications and Theoretical Aspects," in *10th Industrial Conference, ICDM 2010* (R. Goebel, J. Siekmann, and W. Wahlster, eds.), (Berlin, Germany), Springer, 2010.
17. K. Raval, "Data Mining Techniques," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 10, pp. 439–442, 2012.
18. U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "The KDD process for extracting useful knowledge from volumes of data," *Communications of the ACM*, vol. 39, no. 11, pp. 27–34, 1996.
19. J. Sun and J. Tang, "A Survey of Models and Algorithms for Social Influence Analysis," in *Social Network Data Analytics* (C. Aggarwal, ed.), p. 499, New York, NY: Springer Science + Business Media, 2011.
20. A. M. Kaplan and M. Haenlein, "Users of the world, unite! The challenges and opportunities of Social Media," *Business Horizons*, vol. 53, no. 1, pp. 59–68, 2010.
21. C. Hutchings, "Commercial use of Facebook and Twitter - risks and rewards," *Computer Fraud & Security Bulletin*, vol. 2012, pp. 19–20, 2012.
22. D. Solnet, Y. Boztug, and S. Dolnicar, "An untapped gold mine? Exploring the potential of market basket analysis to grow hotel revenue," *International Journal of Hospitality Management*, vol. 56, pp. 119–125, 2016.
23. R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," *94 Proceedings of the 20th International Conference on Very Large Data Bases*, vol. 1215, pp. 487–499, 1994.
24. G. Kaur, "Association rule mining: A survey," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 2, pp. 2320–2324, 2014.
25. I. F. Videla-Cavieres and S. A. Ríos, "Extending market basket analysis with graph mining techniques: A real case," *Expert Systems with Applications*, vol. 41, pp. 1928–1936, 2014.
26. D. J. Cook and L. B. Holder, eds., *Mining Graph Data*. Wiley-Interscience, 2007.
27. L. Tang and H. Liu, *Community Detection and Mining in Social Media*, vol. 2. Morgan & Claypool Publishers, 2010.

28. U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, vol. 76, no. 3, pp. 1–11, 2007.
29. P. Pons and M. Latapy, "Computing communities in large networks using random walks," *Computer and Information Sciences - ISCIS 2005*, vol. 3733, 2005.
30. M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, vol. 69, no. 6 2, pp. 1–5, 2004.
31. R Core Team, "R: A Language and Environment for Statistical Computing." <https://www.r-project.org>, 2017.
32. G. Csardi and T. Nepusz, "The igraph software package for complex network research," *InterJournal*, vol. Complex Systems, p. 1695, 2006.
33. V. Krebs, "Divided We Stand." www.orgnet.com/divided3.html, 2004. Accessed: 17 October 2017.
34. K. Steinhaeuser and N. V. Chawla, "Identifying and evaluating community structure in complex networks," *Pattern Recognition Letters*, vol. 31, no. 5, pp. 413–421, 2010.
35. "Charlotte City Facts." <https://www.census.gov/quickfacts/fact/dashboard/charlottecitynorthcarolina/RHI125216>, 2016. Accessed: 5 January 2018.
36. R. Bernardo, "2017s Most Diverse Cities in America." <https://wallethub.com/edu/most-diverse-cities/12690/#size>, 2017. Accessed: 15 September 2017.
37. R. Team, "RStudio: Integrated Development Environment for R." <http://www.rstudio.com>, 2015.
38. H. Wickham, "tidyverse: Easily Install and Load the 'Tidyverse'." <https://cran.r-project.org/package=tidyverse>, 2017. R Package Version 1.2.1.
39. D. Bates and M. Maechler, "Matrix: Sparse and dense matrix classes and methods." <https://CRAN.R-project.org/package=Matrix>, 2017. R package version 1.2-12.
40. H. Wickham, J. Hester, and R. Francois, "readr: Read rectangular text data." <https://CRAN.R-project.org/package=readr>, 2017. R package version 1.1.1.
41. J. B. Arnold, "ggthemes: Extra themes, scales and geoms for 'ggplot2'." <https://CRAN.R-project.org/package=ggthemes>, 2017. R package version 3.4.0.

42. G. Grolemund and H. Wickham, “Dates and times made easy with lubridate,” *Journal of Statistical Software*, vol. 40, no. 3, pp. 1–25, 2011.
43. H. Wickham, R. Francois, L. Henry, and K. Mller, “dplyr: A grammar of data manipulation.” <https://CRAN.R-project.org/package=dplyr>, 2017. R package version 0.7.4.
44. H. Wickham, *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2009.

REPORT DOCUMENTATION PAGE					<i>Form Approved</i> <i>OMB No. 0704-0188</i>	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.						
1. REPORT DATE (DD-MM-YYYY) 22-03-2018		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From — To) Sep 16 - Mar 18		
4. TITLE AND SUBTITLE MARKET BASKET ANALYSIS WITH SHORTENED WEB LINK CLICK DATA				5a. CONTRACT NUMBER		
				5b. GRANT NUMBER		
				5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) Gallagher, James C., CPT, U.S. Army				5d. PROJECT NUMBER		
				5e. TASK NUMBER		
				5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT-ENS-MS-18-M-123		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) United States Army Cyber Command 8825 Beulah St Fort Belvoir, VA 22060 COMM 703-706-2728				10. SPONSOR/MONITOR'S ACRONYM(S) ARCYBER		
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION / AVAILABILITY STATEMENT DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.						
13. SUPPLEMENTARY NOTES						
14. ABSTRACT Market research is an indispensable part of an organization's ability to understand market dynamics in an area. Over the past 20 years data collection and analysis through Knowledge Discovery through Databases (KDD) has arisen to supplement the traditional methods of surveys and focus groups. Market Basket Analysis is an area of KDD that identifies associations between commonly purchased items. As social media use has grown, link shortening companies help users share links in a constrained space environment and, in exchange, collect data about each user when a link is clicked. This research applies market basket analysis techniques with graph mining to shortened web link data to identify communities of co-visited websites to help analysts better understand web traffic for an area during a time range. Patterns within clusters of web domains regarding hardware platforms, operating systems, or referral sources are then identified and used to gain a better understanding of an area.						
15. SUBJECT TERMS KDD, Market Basket Analysis, Graph Mining, Web Links, Shortened Web Links						
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT U	18. NUMBER OF PAGES 94	19a. NAME OF RESPONSIBLE PERSON Dr. C. Smith (LTC), AFIT/ENS	
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (include area code) (937) 255-6565, x4318; Christopher.Smith@afit.edu	