Graph Analysis of Tracking Services in the Web with Business Perspectives

Master Thesis

by

Hung Chang

Submitted to the Faculty IV, Electrical Engineering and Computer Science
Database Systems and Information Management Group
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science

as part of the ERASMUS MUNDUS programme IT4BI

at the

TECHNISCHE UNIVERSITÄT BERLIN

July 31, 2015

Thesis Advisors:
Sebastian Schelter, Ph.D.

Thesis Supervisor:
Prof. Dr. Volker Markl

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Berlin, July 31, 2015

Hung Chang

ABSTRACT

Mining massive web crawl data to provide understandings of the web is extremely challenging before Common Crawl published TB of web crawl data recent years. One important insight needs to be realized is third-party web tracking who collects users' footprint in the web and provides personalized service by recognizing user intent. This has led to dramatic increasing economic growth and enormous privacy concerns.

To realize the economics of third-party web tracking, we study the revenue of third-party companies by web traffic (how many visitors a third party can see), run of network (how often a third party can appear) and user intent (how well a third party can understand user's interests). This thesis uses Apache Flink to analyze the bipartite graph extracted from Common Crawl, and we exploit graph statistics to symbolize web traffic, run of network and user intent for third-party companies instead of domains.

Our result shows the distributions of revenue factors are extremely skew. Google dominates web-tracking industry in six out of seven graph statistics that symbolize web traffic, run of network and user intent. Revenue is significantly related to web traffic and run of network for top companies (p-value < 0.01). Top third-party companies can raise their revenue by increasing web traffic (coefficient 1.008 and p-value 0.01). They can ease privacy concerns by decreasing user intent (coefficient -0.8179 and p-value 0). This thesis demonstrates an innovative approach to estimate the revenue of web tracking by using massive and open web crawl data from Common Crawl and open big data framework Apache Flink.

ACKNOWLEDGEMENTS

Firstly, I would like to appreciate my advisor PhD Sebastian for investigating and discussing my master thesis, for his patience, critical feedback, and sharing mathematic and programming knowledge. His guidance often helped me solve the challenges during programming, designing approach and writing of this thesis, and his lecture “scalable data mining” definitely facilitates a lot the programming process in the thesis.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Ralf for his suggestion of thesis title, and Prof. Volker for providing such excellent and exciting researching environment and cultures.

My thanks also goes to researcher Christoph and Max, who provided me great feedback for the evaluation section. Without their support it would be more challenging to improve the evaluation section

I thank my fellow IT4BI classmates Amit, Jacob, Anil, Alexey, David, Karim, Igor, Tamara, and TU student Felix and Philips for the stimulating discussions, for the days and nights we were working and discussing together, and for all the fun we have had.

Also, I thank European Commission and the IT4BI committee selected me as a scholarship so that I can fully concentrate on my study without any financial stress.

Finally, I would like to thank my relatives for supporting me spiritually throughout writing this thesis and my life in general.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The section describes why we want to analyze tracking services in the web with business perspectives by stating the motivation and problem statement.

1.1 Motivation

The initial motivation of this thesis derives from mining massive web crawl data from Common Crawl[1] which provides open data that everyone can utilize. Web crawl data contains the information of web pages, and the first analysis of web crawl data is the snapshot of hyperlink relationships between web pages[2] and gives an overview of the macroscopic structure of the Web in 2000[3]. However, the size of data is much smaller than Common Crawl because developing web crawler already consumes lots of workload. Mining massive web crawl data was rare due to the difficulty to have representative amount of web data[4]. The web is extremely large and web crawler must handle massive data with high throughput to simulate modern search engine companies who exploit thousands of servers and lots of high-speed network links. Except for scale, crawling high-quality web pages and ignoring malicious web pages brings more efforts. Some servers even mislead web crawler to some specific websites for business needs, and block the web crawler due to too high throughput and assume the web crawler as a denial-of-service attack.

Common Crawl contributes to solve those challenges. Common Crawl collects massive web data by crawling the web using Hadoop and particular crawling strategy and provides it publicly through Amazon's web services[5]. One application was extracting the hyperlink relationship between web pages and compared the finding in 2000, and computed graph statistics in 2012[6]. For example, that research revealed the website "word.press.org" had the highest in-degree. They also aggregated the hyperlink graph to different granularity[7] and computed the graph statistics with TLD which are ".org", ".com" and ".de". Other researchers analyze domain statistics[8] and the effect of Google Analytics[9].

Although the effect of Google Analytics using the data from Common Crawl has been researched, there is not yet mining Common Crawl for extensive third-party web tracking. When

user surfs on a website there are numerous hidden third parties are watching user's footprint in the web to understand the interest and provide personalized service. The pros and cons of web tracking has been argued severely in E.U and the United States. Supporters of web tracking mention that web tracking let the web services really know users' interests and provide personalized services for users. The examples are many and almost all websites collect user's behavior and it's a remarkably huge business[10]. Protesters judge web tracking harms privacy by an example. You walk on the road and a person is always behind you and takes non-volatile note about all of your moves no matter you are an adult or a child, and you go to see a doctor or shopping in a store[11]. In E.U and the U.S the law for balancing privacy and the economic aspects are still intensively discussing.

In industry, BuiltWith, eMarketer, Alexa, comScore and other market research companies have analyzed web tracking with web statistics and business performance for popularity and revenues, and you can observe the popularity of web tracking for recent weeks, or read the annual report of digital advertising which is the main benefit of web tracking for free to access. However, their overall data is not public and only free to access for recent weeks and top domains. On the other hand, in academic, the revenue of web tracking at domain level by applying the concept of digital advertising estimation was researched because mostly web-tracking revenue comes from advertising revenue, for example, in the current biggest company Google, it has extremely high web traffic (how many visitors a third party can see), run of network (how often a third party can appear) and user intent (how well a third party can understand user's interests)[12] but their data is not public. Thus, the motivation of this thesis is to show the possibility of estimating the revenue of web tracking, and let us understand the economics of web tracking by using massive open data.

1.2 Problem Statement

In order to discover insight from the massive web crawl data from Common Crawl, we aim to understand the economics of third-party web tracking. Third-party web tracking collects users' footprint in the web and users are not aware of it. This has led to privacy concerns and hug business

by providing personalized services. We want to realize the revenue of third-party web tracking by how much information it can see, how valuable and sensitive the information is.

Currently the understanding of the economics of third-party web tracking is not easily accessible. Those information is either not free to access, or available only for recent weeks and top domains published by the companies who advocate web tracking. In 2013, a research investigated the distribution of web-tracking revenue by using the concept of revenue estimation for digital advertising[12]. They examines three factors affecting the revenue that are web traffic (how many visitors a third party can observe), run of network (how often a third party can appear) and user intent (how well a third party can understand user's interests). Web traffic and run of network are related to how much information web tracking can see, and user interest is related to how valuable and sensitive the data it collects. While the result they published shows a skew revenue distribution, their data source is restrictive and not massive, and the revenue is for domains rather than for companies, therefore, they cannot evaluate their findings.

Web crawl data from Common Crawl is an open and massive data. We analyze web tracking from this data with business perspectives to understand the revenue of third party at company level rather than domain level. We investigate the revenue distribution by web traffic, run of network and user intent. This thesis analyze this massive data with Apache Flink which can process massive data in parallel and graph algorithm efficiently[13]. We propose to provide a comprehensive study for estimating the revenue of web tracking based on graph statistics symbolizing web traffic, run of network and user intent, and we will investigate the correctness of the result.

2. BACKGROUND

This section describes the required knowledge to understand the thesis. We first introduce third-party web tracking, and then we explain the graph structure in the web. Next, we introduce the open framework Apache Flink, WHOIS protocol and regression analysis. Finally, we describe related work.

2.1 Third-Party Web Tracking

This section states the concept of third-party web tracking, the pros and cons and web-tracking technologies. We first explain some terminologies, and then talk about the disadvantages. Finally we describe the web-tracking technologies.

2.1.1 Third Party and First Party

Third party is the instance on the web collecting the information of users' movements in the Internet. Exploiting this information facilitates targeted advertising and improves development and delivery of websites. Nowadays most websites have been involved with third-party web tracking.

To understand how third-party web tracking functions, let's see its interaction with first party in the web. First party, also called publisher, is the website that user enters an URL and directly communicates with. When user enters into a publisher, browser auto-redirects to many third parties. CDN (Content Delivery Network) to accelerate image and video loading, embedded as <img>. Web analytics, advertising, and social network, embedded as <javascript> or <iframe>.

Figure 1 is a publisher and user's browser auto-redirects to those third parties where user don't intent to visit, even web analytics is not visible for user. Web analytics observes visitor's activity using Google analytics as an example in Figure 2.

Figure 1: Third party on the New York Times website[14]

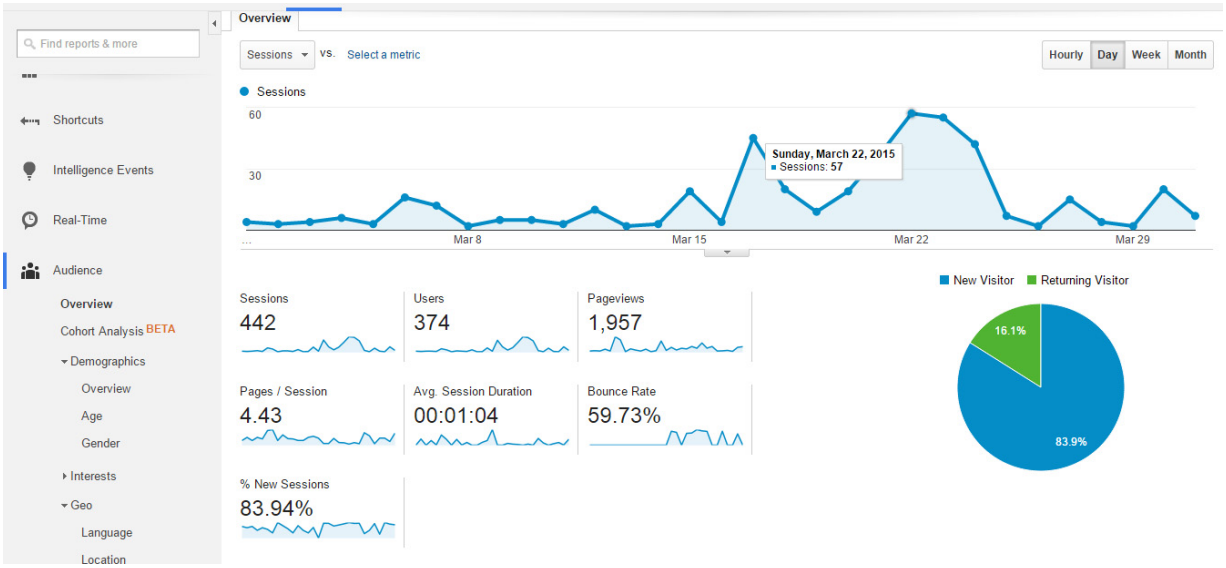

Figure 2: Tracked information from Google analytics

2.1.2 Dark Side

Some information tracked by third party is more sensitive and people usually don't want this information to be stored and recorded. Even law clearly states some specific data of human is forbidden to let others know. For example, the information about health is people don't want others to know, for example, they have psychological disease. When user browses health related website, third party know user is searching for certain keywords found in http requests[15]. Figure 3 shows the example that the information leaks to third party "quantserve" and user is not aware of this leakage.

GET http://pixel.quantserve.com/pixel;r=1423312787...
Referer: http://search.HEALTH.com/search.jsp?q=pancreatic+cancer

Figure 3: Linkage in health website in http requests

Credit card information is also particularly sensitive and the leakage brings trouble. There is no website leak the information about credit card through http requests[15] and these websites are fiduciary sites as websites in health category[16].

Third party can link the information from multiple websites with cookies which contains unique global profile. Third party build extensive user profile by linkage. Without cookies third party can also use email to link the information. Figure 4 shows an example that third-party DoubleClick observes cookie id and email, and user's employment information.

GET http://ad.doubleclick.net/activity;...
Referer: http://f.nexac.com/...http://www.EMPLOYMENT.com/...
na fn=John&na ln=Doe&na zc=12201& na cy=Albany&na st=NY&na a1=24 Main St.& na em=jdoe@email.com...Cookie: id=22a348d29e12001d...

Figure 4: Linkage information using email in http requests

Another approach for information linkage is browser fingerprinting. In 2010 a report shows out of sample of nearly 500,000 browsers, 83.6% were uniquely identified and 94.2% of browsers with Flash or Java enabled were uniquely identified[14]. Figure 5 shows the configuration of browser is leaked in http requests. Evidence shows that linkage of this information is possible.

GET http://std.o.HEALTH.com/b/ss/...global/...p=Google Talk Plugin;Google Talk Plugin Video Accelerator; Adobe Acrobat;Java Deployment Toolkit 6.0.210.7; QuickTime Plug-in

7.6.6;Mozilla Default Plug-in; Google Update;Shockwave Flash;Java(TM) Platform SE 6 U21;...Referer: http://www.HEALTH.com/search/...?query=pancreatic cancer... Cookie: ... s query=pancreatic cancer

Figure 5: Linkage with browser fingerprinting in http requests

When users surf the websites, their information leaks to third parties no matter they are willing to or aware of, and no matter the data is sensitive or insensitive. Third party can link users' web activity from different first parties leaked to them and create broad user profile.

2.1.3 Technologies

Stateful and stateless technologies are two characteristic divide the technologies of tracking[14]. Stateful tracking, also called SuperCookies, tracks user by putting cookies in client-side computers such that website can memory the user. The basic idea is to use cookies which contains global unique identifier and it makes the client's device become unique identifiable and memorable. Many online advertising companies, including ClearSpring, Interclick, Specific Media, and Quantcast, Microsoft and KISSmetrics used such stateful technology to track user[14].

Stateless tracking, also called fingerprinting or browser fingerprinting, tracks user via identifying the properties of browser. The properties of browser represent distinguishably by installed font, plug-in, CPU and operating system and other properties as shown in Table 1. Research shows these properties of browser form a unique identifier[17]. Known companies applying such technology are 41st Parameter/AdTruth, BlueCava[14].

Another way to looking at third-party web tracking from technology side is classifying them based on the embedded code in first party, specifically, the tracking code in first party and five categories of tracking code exist[18]. Basically, different kinds of script such as <iframe>, <script> and the cookies location determine the categories. Table 2 summarizes them[18].

operating system
CPU type
user agent
time zone
clock skew
display settings
installed fonts
installed plugins
enabled plugins
supported MIME types
cookies enabled
third-party cookies enabled

Table 1: Information collected for fingerprinting

Category	Summary	Example
A	Serves as third-party analytics engine for sites.	Google Analytics
B	Uses third-party storage to track users across sites	DoubleClick
C	Forces user to visit directly (e.g., via popup or redirect).	Insight Express
D	Relies on a B, C, or E tracker to leak unique identifiers.	Invite Media
E	Visited directly by the user in other contexts.	Facebook

Table 2: Classification of web tracking based on tracking code

2.2 Web Data

This section describes the graph structure in the web. We first introduce the graph structure in our data. Then, we discuss centrality in hyperlink graph and the granularity of web graph.

2.2.1 Bipartite Graph

The first paper examines this kind of graph in the web calls it as a graph composed of visible nodes and hidden nodes[19]. Visible node is first party and hidden node is third party. In graph theory, this graph is called two-mode graph or bipartite graph whose nodes can be divided into two disjoint sets U and V such that every edge connects a vertex in U to one in V and a node u in U doesn't connect the other nodes in U, and this condition also applies for the nods in V.

Figure 6 is an example of bipartite graph in the web. It means google-analytics embeds in Newyorktimes.com, Mediamarkt.com and Example.com. When surfing those website, "google-anlytics.com" receives their web activity and can link their data to build their footprint. Besides, when surfing in "mediamarkt.com", "google-analytics.com" and "googleadservices.com" can observe their web activity.

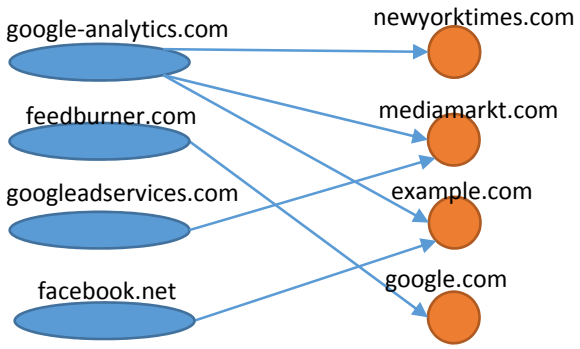

Figure 6: Biparptite graph representing embedding relationship in the web

We obtain the bipartite graph representing embedding relationship in the web extracted from Common Crawl by the work of Sebastian and Felix[1].

2.2.2 Centrality Measures in Hyperlink Graph

The centrality of a node represents the importance of it in the network and has three different groups based on the definition[20]. The first group is Geometric measures which interprets the importance as the number of nodes exists at every distance. In-degree of a node counts the number of nodes at distance one of this node. A node's in-degree is equal to the number of its incoming arcs and these arcs represent the connecting nodes vote for this node. In-degree is a simple measure and has many shortcomings comparing to other sophisticated centralities, but it's a favorable baseline.

[1] https://github.com/sscdotopen/trackthetrackers

Another Geometric measure is closeness and it measures the average distance from a node to the other nodes and interprets the importance as that a higher centrality has smaller average distance reaching other nodes. However, in a directed graph some nodes are not reachable starting from a node called unreachable pair nodes. Harmonic centrality is an improved closeness version that solving the pair of unreachable nodes by replacing computing average distance by harmonic mean of all distances[20]. Here is an example. Suppose a site example.com. The first iteration counts the number of sites links to example.com at distance one suppose there are 10, and then total score now is 10. The second iteration counts the number of sites links to example.com at distance two, and suppose there are 20, however, they are not as important as at distance one, so this score now is weighted and it becomes 20 / 2 and the total now is 30. The iteration continues until certain condition and terminates. An implementation of it for large graphs called Hyperball centrality for extremely large graphs[21], and there is a similar implementation and intuition[22].

The second group is Spectral measures which is related to computing left dominant eigenvector of the matrix derived from the graph and different matrix definition leads to different measures[20]. The intuition is important nodes have important neighbors which is a recursive computation. Ideally, the left dominant eigenvector will become convergence that every node starts with the same score, and then substitutes its score with the sum of the scores of its predecessors. The vector is then normalized, and the process repeated until convergence or certain termination condition. The dominant loses its interpretation if the graph is not strongly connected. Katz follows this intuition and computes the centrality by measuring the number of the immediate neighbors, and also all other nodes that connect to these immediate neighbors with penalized factor. Katz can be expressed as

$$k = 1 \sum_{i=0}^{\infty} \beta^i A^i$$

β is the penalized factor and A is the adjacent matrix. Using the idea of Taylor Expansion can rewrite the above formula as

$$k = 1 (1 - \beta A)^{-1}$$

Displacing eigenvector λ can rewrite the formula as

$$\beta\lambda A + (1 - \beta\lambda)e^T 1$$

e is the unit vector. If β is equal to one, this formula is the same as eigenvector centrality.

Katz counts the number of walks from a node and penalizes the longer walk. However, a node gives its important score to all his neighbors is not fairly reasonable and PageRank improved this issue. Considering "google.com" has extreme high importance and millions outgoing links to other webpages, and then the importance of "google.com" should be evenly distributed and divided to all its neighbors based on the amount of its neighbors due to its neighbor is just one out of million. Thus, the centrality of a node propagates to its neighbors is proportional to their centrality divided by their out-degree[23]. The third group is the path-based measures and they are not used in this thesis because WebDataCommons doesn't provide them and they are not able to be computed in parallel strictly following the definition.

WebDataCommons extracted the hyperlink relationship between webpages and published the centrality of the hyperlink graph in 2012[6] including degree distribution, PageRank, connected components and centrality. They also aggregated the hyperlink graph to different granularity including pay-level-domain, subdomain and page[7]. Those are open data we can use for free.

2.2.3 Granularity

Web graph has granularity and hierarchies based on the domain name. In terminology of web, they are page, host, pay-level domain, and top-level domain. Different hierarchy has different usage. In the following we explain it with examples.

At the page level every web page with all details as single node in the graph. An example for a node in this graph would be "dima.tu-berlin.de/menue/database_systems_and_informationmanagement_group/". At host level each subdomain is represented as node. Two web pages "tu-berlin.de" and "dima.tu-berlin.de" are two different nodes within this graph. At pay level two nodes "tu-berlin.de" and "dima.tu-berlin.de" are represented as a single node "tu-berlin.de". At top level, we can understand this domain is managed by private company or government, for example, .com is for company and .gov is for

government. Also, we can recognize the country of domain if the length of top level domain is smaller than three. For instance, ".de" represents German domain.

With the domain's company information, we can further aggregate to company level. For example, "google.com", "youtube.com" are managed by the same company "Google Inc.". With category of domains information, we can aggregate domain to category level. For instance, all health related website become health category.

Based on the size of data, we can conclude that page-level domain requires the biggest storage, and then host level and pay level. Company level requires less storage than pay-level domain, and top level-domain needs the smallest storage.

2.3 Apache Flink

Apache Flink is a large-scale data processing framework with memory management, native iterative processing and a cost-based optimizer.

Flink has its custom memory management using an internal buffer pool to allocate and de-allocate memory by custom serialization and de-serialization[24]. Flink serializes data object to memory and then write to file system. Meanwhile, Flink reads file from disk to memory and de-serializes it to data object. This decreases the number of data objects on the JVM heap.

Flink has native iterative procession and is considered an efficient API especially for massive graph processing[25]. Flink increases the efficiency of iterative computations which are fairly common in graph algorithms with bulk and delta iterative computations. In an iterative computation, Flink replaces the partial solution in the input and outputs to step function. In delta iteration, Flink ignores the data that needn't process and only processes the data still needed to be computed by checking whether the data has been processed or not in each iteration. This improves efficiency because the data in real world often has this feature that many data only needs to process in early iterations and much less data needs to be process in later iterations.

Flink optimizes the operators by enumerating execution plan and chooses the best plan depending on the relative size of data and the memory of machine. The execution plan will be different depending on the machine is running on a big or small cluster or a laptop. Flink optimizes

operators such as map, filter, and join optimizations including the technique of hash and sort-merge with the properties that the data is sorted or partitioned. To customize program execution, we can also provide the relative size of data as the hint for the operators.

2.4 WHOIS Protocol

WHOIS data is the information of domain name managed by ICANN including identifying and contact information such as owner's name, address, email, phone number, and administrative and technical contacts[26]. One can get a domain's information by typing "whois google.com" in command line, and WHOIS responses with the domain's information.

ICANN only knows domain's information at the top level which means typing "whois google.com", WHOIS responses the information about MarkMonitor which is Google's registrars because technically, WHOIS service is not a single, centrally-operated database. Instead, the data is managed by independent entities known as registrars and there are hundreds of registrars. Thus, to know the information of "google.com", we need to send WHOIS query to the specific WHOIS server by typing "whois -h whois.markmonitor.com google.com" and this will return the information we want while the default WHOIS server is ICANN WHOIS server.

There are some companies don't follow WHOIS policy which means they hide their domains' information due to privacy concerns, and some business help these companies to hide the information[27]. As the result of privacy concerns, WHOIS doesn't allow massive electronic accessing the information although many companies selling massive and historical WHOIS information.

2.5 Regression Analysis

Regression analysis is a method using one or many independent variables to explain one or many dependent variables. This shows not only the relationship between the variables, but has the ability for predicting and explaining the relationship between the variables.

A simple linear regression consists of a dependent variable y and an independent variable x, a constant c and the coefficient b_1, as shown below.

$$y = c + b_1 \times x$$

OLS (Ordinary Least Squares) or gradient decent methods computes the minimum difference between the predicting value and the real value, and it produces c and b_1. That difference is called residual, and given x the value of y is known. b_1 is the slope and indicates the effect that changing one unit of x brings to y as the following formula shown.

$$b_1 = \frac{\Delta y}{\Delta x}$$

In more complex setting, there are more independent variables and dependent variables, different loss functions, and different methods to compute loss functions.

In statistics, we focus on how the regression model fits the assumption that the residual follows normal distribution, homoscedasticity and independent. With any broken assumption the regression loses some power to predict and explain the relationship between variables and coefficient because the formula deriving the coefficient of regression already assume those assumptions are true. Observing the regression line and the real data point discovers the misbehavior of not obeying those assumptions. A common solution is to use log transformation for variables[28] or robust regression[29].

Using logs transformation gives us a log-log model.

$$Log(y) = c + b1 \times log(x)$$

Several benefits of using log transformation for each variable exist. First, with log transformation we can make the units of measurement of each variable more consistent. Second, it interprets the regression in percentage level. The effect of one percentage changes in x brings to $b1$ percentage of y due to the property of natural logarithm[30]. Third, after using log transformation the data usually follows the assumptions and mitigates the problem if initially the residual doesn't follow the assumptions. Forth, log transformation narrows the range of data and makes the data less sensitive to outliers.

Limitations exist in log transformation. The value in log function should be positive numbers, which means if the value is less than and equal to zero, we cannot use log transformation.

2.6 Related Work

This section describes the data that past research have used for analyzing third-party web tracking and the revenue estimation have been studied.

2.6.1 Data Usage for Analyzing Web Tracking

The past researches of web tracking is restrictive and their data is much smaller than our research uses TB data. We are going to explain it based on different researches.

1075 first parties and 2926 third parties were examined in 2006[19]. The amount of URL shown was not the top private domain and that research aggregated URL to higher level. In five periods the leakage of 1200 popular Web sites was discussed in 2009[16]. The leakage of top 120 popular websites was examined in 2011[15].

The leakage to third party of five periods data is gathered in 2012[14] and published their information as the following: Alexa Top 500 United States Sites, April 14, 2012 (25MB zip). Alexa Top 10,000 Global Sites, August 7-9, 2011 (250MB zip). Alexa Top 10,000 Global Sites, July 23-26, 2011 (288MB zip). Alexa Top 10,000 Global Sites, July 21-23, 2011 (331MB zip). Alexa Top 1,000 Global Sites, July 26-27, 2011 (193MB zip).

The leakage to third party of top 500 popular domains in Alexa was examined and found 524 third parties in 2012[18]. Totally 70M http request/response of tracking data was gathered in 2013[12].

Due to the difficulty of having representative volume of web crawl data, those researches have already examined comparatively large data except that comparing to our study. None of those researches even exploit data at GB level comparing to our raw TB data.

2.6.2 Revenue Estimation for Web Tracking

comScore, which managed by ValueClick, and eMarket devote to publish the revenue of digital advertising which is the main benefit and business of web tracking. comScore's concept of estimating the revenue is to exploit web traffic, run of network and user intent. This has been

applied to estimate the revenue distribution of web tracking without evaluating the result[12]. They view the revenue of third party as a function of amount of data, therefore, they examine how much information third party can see and how valuable it is. The factor run of network they used is a constant that from Google AdWords that all third parties have the same value.

They gather the tracking data with HTTP requests/responses by their designed software and extract the data from Alexa and Google AdWords. The result shows the revenue distribution is skewed and Google dominates the industry that are embedded in 80% of first party. Our study applies the same, and we estimate the revenue by using graph statistics to symbolize web traffic, user intent run of network. Although we follow the same idea and extract the related data from Google AdWords and Alexa, we use much bigger data (TB) and extract data from WHOIS which allow us to investigate the revenue at company level rather than at domain level.

3. APPROACH

After knowing what are the factors affecting the revenue of third-party web tracking, this thesis computes graph statistics to symbolize web traffic, run of network and user intent. In other words, we view the computed graph statistics as different business meanings behind.

Computing those graph statistics needs integrate the bipartite graph extracted from Common Crawl with other heterogeneous data. We extract data from WHOIS, Google AdWords and Alexa and integrate those data with the bipartite graph where the join key is domain. Besides, we implement those graph statistics computation with Apache Flink and chapter 4 will describe about it.

3.1 Overview of Revenue Estimation

We process the bipartite graph and compute graph statistics for estimating the revenue[2]. Estimating the revenue is involved with web traffic, run of network and user intent[12]. More specifically, the web traffic a third party can see, the ability to be embedded in first party and the user profile a third party can recognize. User interest is the controversial part because user intent provides customized service but it's also sensitive.

To reveal the relationship of those factors, we first aggregate the bipartite graph from domain level to company level. Then, we start to gather statistics representing web traffic, run of network and user intent, and all graph processing is with Apache Flink.

Figure 7 displays the overall process of revenue estimation. Other researchers provides the bipartite graph from Common Crawl and the centrality of hyperlink graph. Based on these results our study integrate the bipartite graph with the information of domain's company, centrality in hyperlink graph, category of first party, and keyword values we extract. This means we aggregate the third parties in bipartite graph from domain level to company level and category level as we mentioned in chapter 2.2.3.

---

[2] https://github.com/HungUnicorn/trackthetrackers

Then, we integrate the company-level bipartite graph with the centrality in hyperlink graph from WebDataCommons[31] to present web traffic. Meanwhile, we compute weighted PageRank to present run of network by implementing one-mode projection from projected bipartite graph. Besides, aggregating first parties in bipartite graph from domain level to category level and integrate with the keyword value in Google AdWords to present user intent.

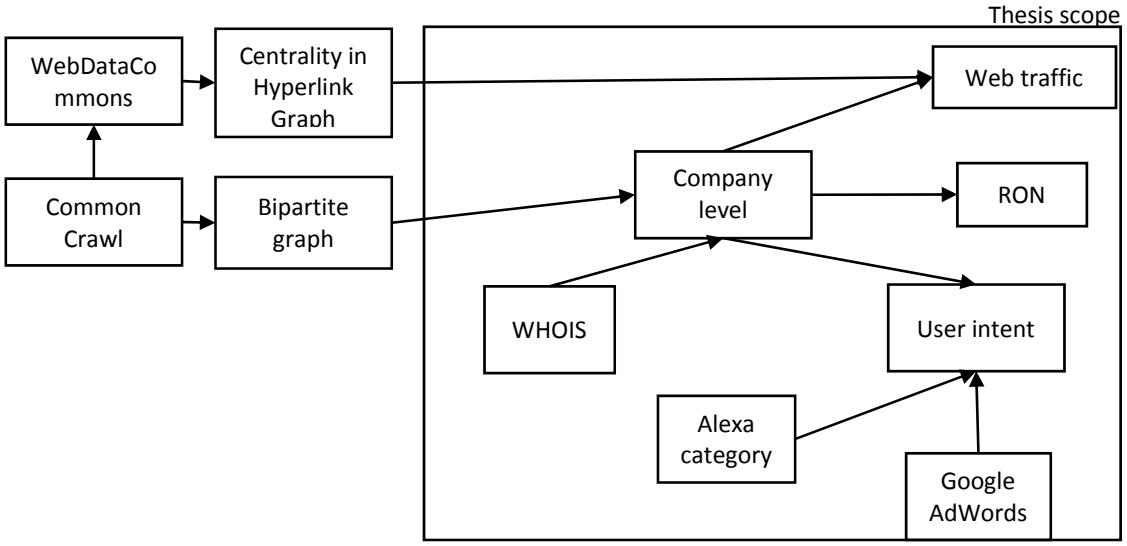

Figure 7: Approach Overview

3.2 Third-party Domain's Company

This thesis wants to estimate the revenue of company. We need to know the domain belongs to which company for each domain, and then can aggregate domain's performance to company's performance. For example, "youtube.com", "googleapis.com" and "google.com" are all the domains owned by Google, and aggregating those three domains performance becomes Google's performance. In other words, after knowing the domain's company information, we aggregate the bipartite graph from domain level to company level.

3.2.1 Company Information of Third Party

There are several ideas to identify domain's company. A domain name containing a symbol, for instance, "google" is easily recognized as the company Google, but it cannot let us know the acquisition, for example "youtube.com", as well as the CDN of Google. Manually checking the acquisition is possible for small data[16] and they collected 15 domains for 7 companies. A better way to check all the acquisition is using DBpedia, however, it still has no information about CDN's company as in Figure 8. In Figure 8 there are acquisitions by Microsoft.

This thesis uses WHOIS protocol to extract company information of domain. The company information locates in the Admin Organization field in WHOIS response as in Figure 9. From this WHOIS response we know Facebook manages "fbcdn.net".

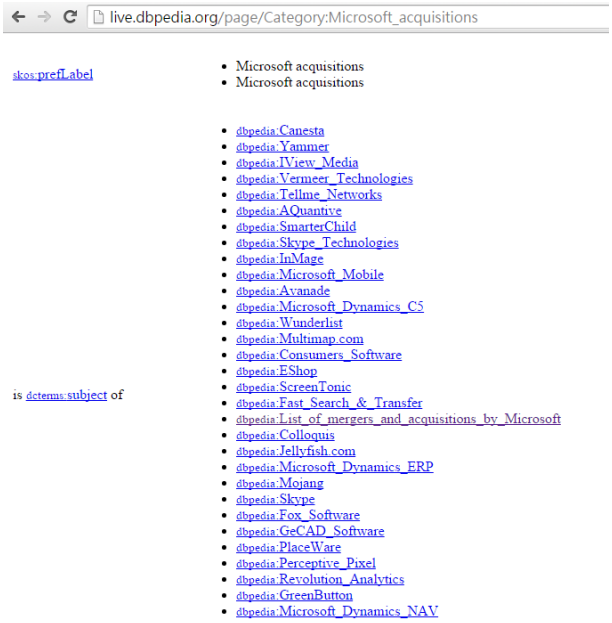

Figure 8: Acquisition of Microsoft

```
Domain Name: fbcdn.net
Registry Domain ID: 956686477_DOMAIN_NET-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2014-10-28T12:38:28-0700
Creation Date: 2007-05-03T11:49:03-0700
Registrar Registration Expiration Date: 2018-05-03T11:49:03-0700
Registrar: MarkMonitor, Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2083895740
Domain Status: clientUpdateProhibited
(https://www.icann.org/epp#clientUpdateProhibited)
Domain Status: clientTransferProhibited
(https://www.icann.org/epp#clientTransferProhibited)
Domain Status: clientDeleteProhibited
(https://www.icann.org/epp#clientDeleteProhibited)
Registry Registrant ID:
Registrant Name: Domain Administrator
Registrant Organization: Facebook, Inc.
Registrant Street: 1601 Willow Road,
Registrant City: Menlo Park
Registrant State/Province: CA
Registrant Postal Code: 94025
Registrant Country: US
Registrant Phone: +1.6505434800
Registrant Phone Ext:
Registrant Fax: +1.6505434800
Registrant Fax Ext:
Registrant Email: domain@fb.com
Registry Admin ID:
Admin Name: Domain Administrator
Admin Organization: Facebook, Inc.
Admin Street: 1601 Willow Road,
Admin City: Menlo Park
Admin State/Province: CA
Admin Postal Code: 94025
```

Figure 9: WHOIS response for Facebook's CDN

3.2.2 Efficiently Accessing WHOIS Information

WHOIS server doesn't allow massive automatically accessing the information written in term of use due to some personal information like address and phone. We have 27275530 third-party domains and accessing the information every 5 seconds is not possible to have all the information.

We observe WHOIS data is not clean especially when the domain less important and popular that the organization information is either missing, junky or in proxy protection. From top 100000 domains the program successfully extracts around 10000 companies though from top 500 it gets around 490 effective domains, and the rest are junky. This finding supports accessing the

information for all 27275530 third parties is not necessary because many of the information would be dirty, useless and missing.

According to the above discussion, except that sending request every x seconds, the crawling process needs to be efficient that the program only sends request to WHOIS server when the domain is important. In other words, rather than sending all the domains to WHOIS server, the program filters some unimportant domains that might have unclean information. And we assume that important domains manage their information well.

We consider the important domains of third party collects lots of web traffic. We view the domain is important, if this domain has high centrality in the hyperlink graph from WebDataCommons and we use PageRank and Harmonic Closeness as in Figure 10.

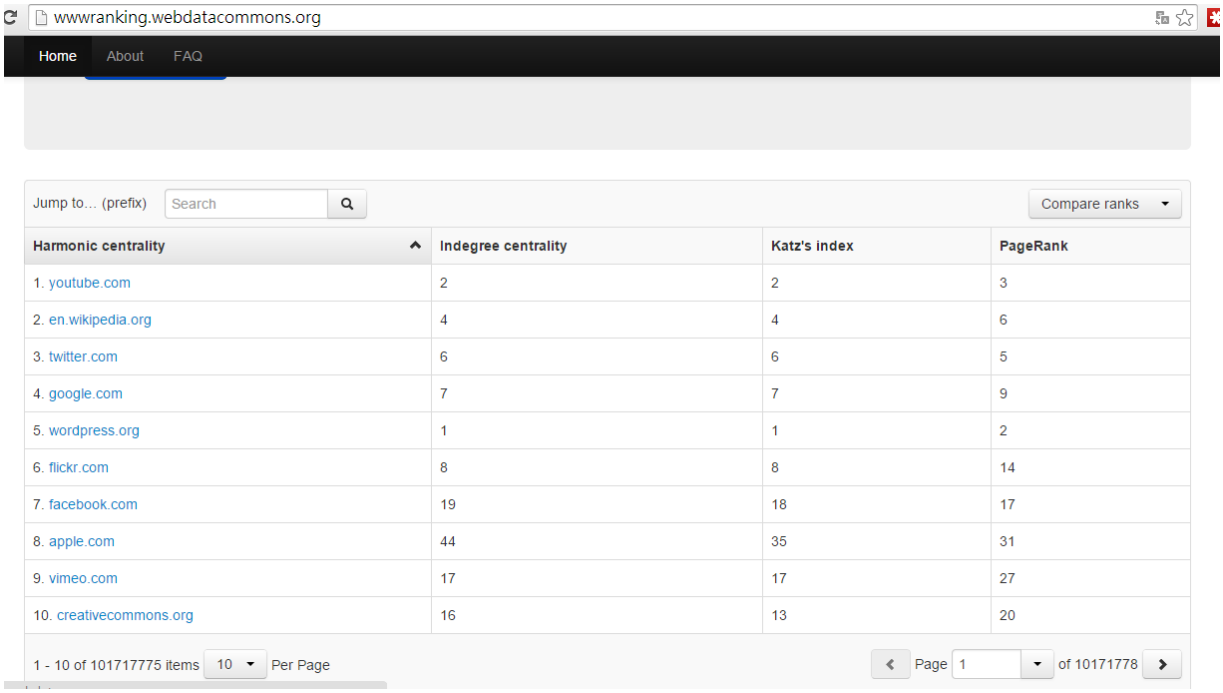

Figure 10: Centrality in hyperlink graph

We measure the importance of third party by sum of centrality that are sum of PageRank and sum of harmonic closeness. Figure 11 shows the computation that a third party's sum of centrality is summing all its embedded first parties' centrality. For example, "Googleanalytics.com" is 1+2+103+400+400 and "Googleapis.com" is 400.

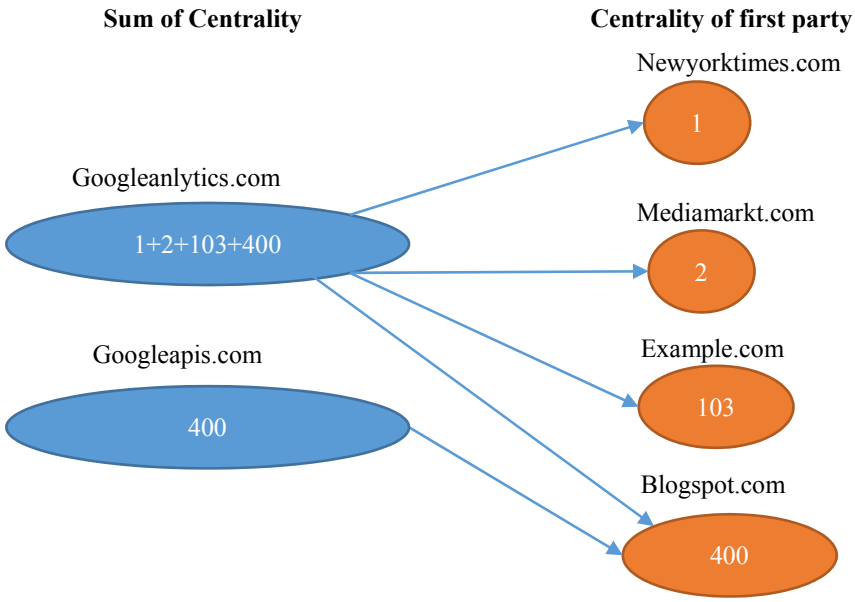

Figure 11: Sum of centrality computation

The concrete extracting progress is as Figure 12. We first use Apache Flink to get the top sum of centrality domains from the bipartite graph. Second, we send this domains and parse the WHOIS response through our WHOIS response parser.

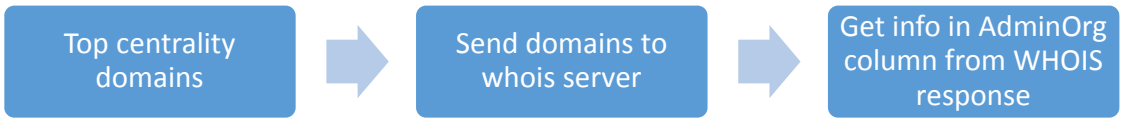

Figure 12: Company info extraction process

Other accelerating techniques we apply are incremental extraction and checking the domain contains the special company symbol. Applying incremental extraction is because the crawling process has to stop for some time, and then continue. WHOIS server block request hourly and daily. Thus, the program only sends the domain to WHOIS server which the company is absolutely not yet known. Checking domain's symbol is to see if a special word like google, yahoo and other symbols is in the domain, these domains have the same company, for example, "google.org" and "google.com" have the same company Google.

3.2.3 Investigation

After this extraction process, it generates a mapping file that each domain has a company name. Due to the limit of access WHOIS server and the poor quality of WHOIS information, we want to keep the data as clean as possible and extract the data efficiently. Therefore, we focus on ".com", ".net" and ".org" domains, and view the WHOIS information which Admin Organization" contains "LLC", "Ltd.", "Inc.", "corporation" standard keywords that recognized as company. For example, Microsoft corporation, Facebook Inc. and Google Inc.

We want to observe that crawling company information is less efficient when accessing the information for less popular domains as our initial assumption. Figure 13 shows that there is a dramatic decrease when accessing the domains whose rank is the top 5000 domains. The program extracts effective result less and less. Before handling the problem in Google API that google related domains are at different granularity than other domains, there is a strange dramatic increase for the domains whose rank is around 25000. The domains that Google API cannot aggregate to the same granularity appears so often around that interval, and many "XXXX.blogspot.com" exist at that interval.

We handle this issue that those domains which Google API cannot process by aggregating them to the same granularity as others. For example, xxxx.blogspot.com becomes only one instance "blogspot.com". After this clean process, Figure 14 shows our assumption is correct. When accessing less important domains, the meaningful results are less and less than accessing more important domains. This means we obtain much more company information from top 1000 domains than top 2000. The phenomena is easier to observe when using sum of harmonic closeness to represent the importance of domains as Figure 15.

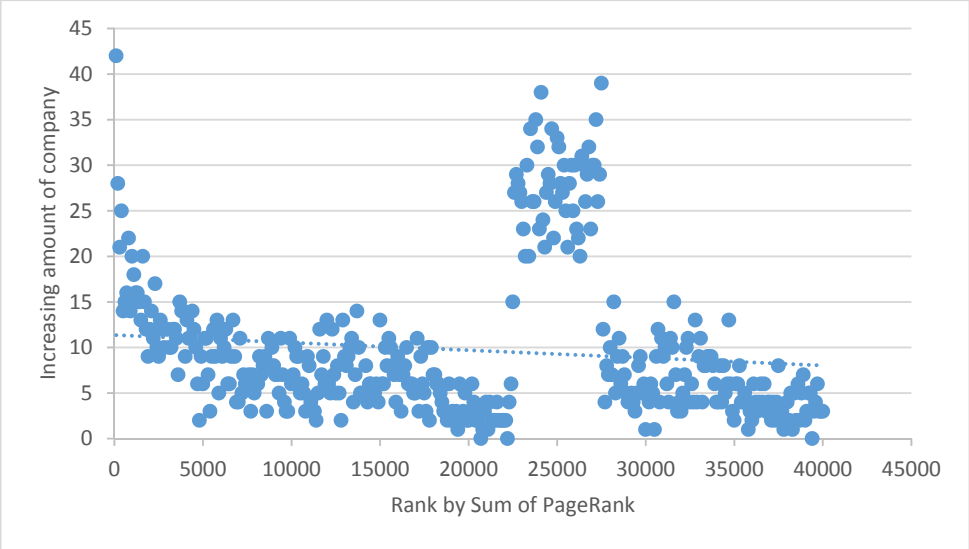

Figure 13: Increasing amount of extracted company information and Rank (Sum of PageRank) before handling Google API problem

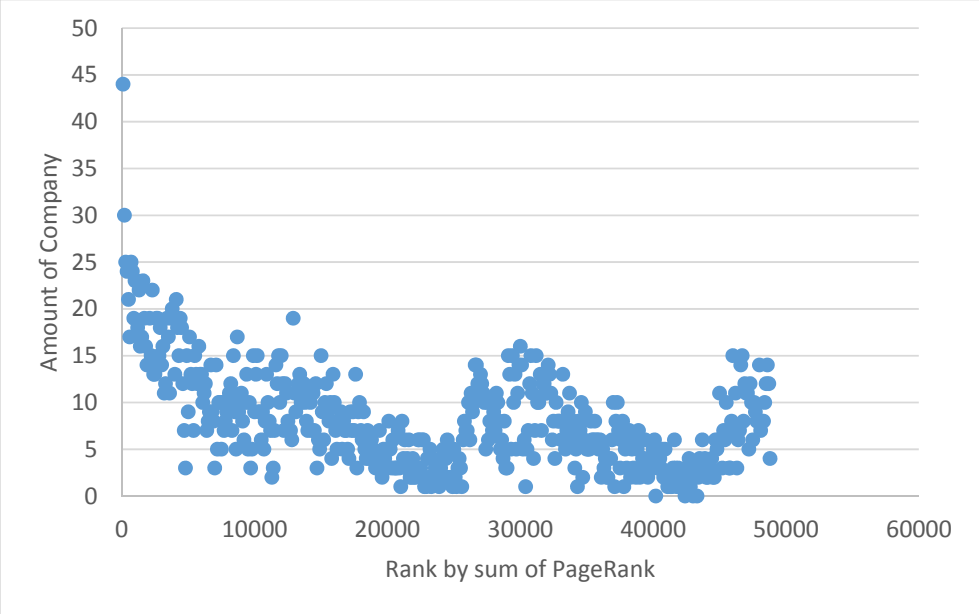

Figure 14: Increasing amount of extracted company information and Rank (Sum of PageRank) after handling Google API problem

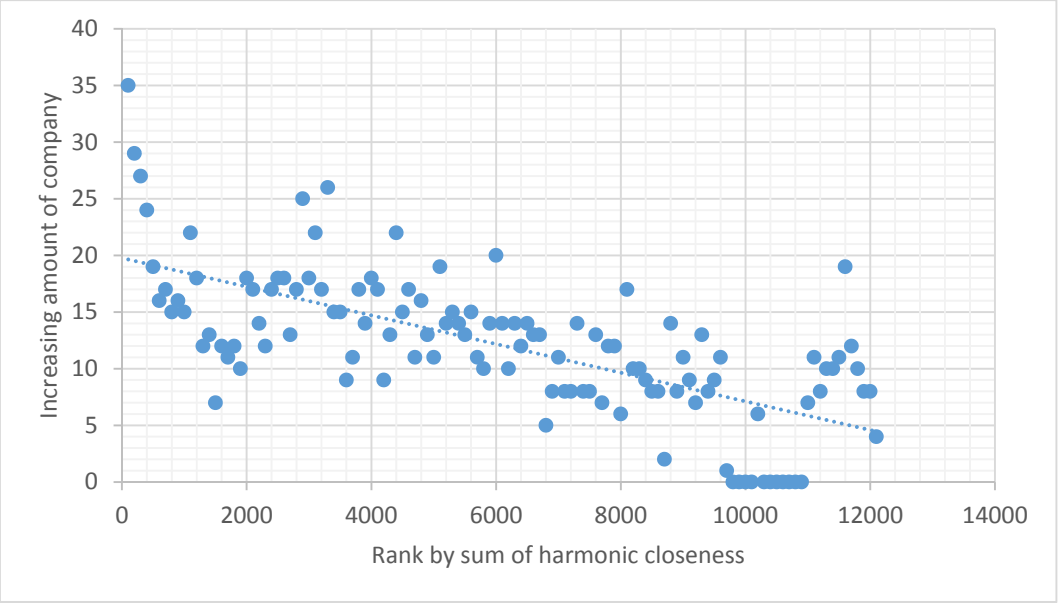

Figure 15: Increasing amount of extracted company information and Rank (Sum of harmonic closeness)

3.2.4 Aggregating to Company level

Using the information containing domain's company aggregates the bipartite graph from domain level to company level with Apache Flink. At the first step, it joins with company information and replaces domain name with company name. Second, one important concept is to get the distinct arc. A company having two different third party embedded by the same first party means this company can watch this domain and there is no need to count twice. This follows the strategy that when the bipartite graph extracted from Common Crawl, if a third party is embedded twice or more in a first party we count the occurrence only once.

Figure 16 demonstrates a concrete example. With domain and company mapping information we replace third party by its company. Then, because Google administrates "google-analytics.com", "feedburner.com" and "googleadservices.com", redundant arc is removed such that there is only one arc from "Google Inc." to "Newyorktimes.com".

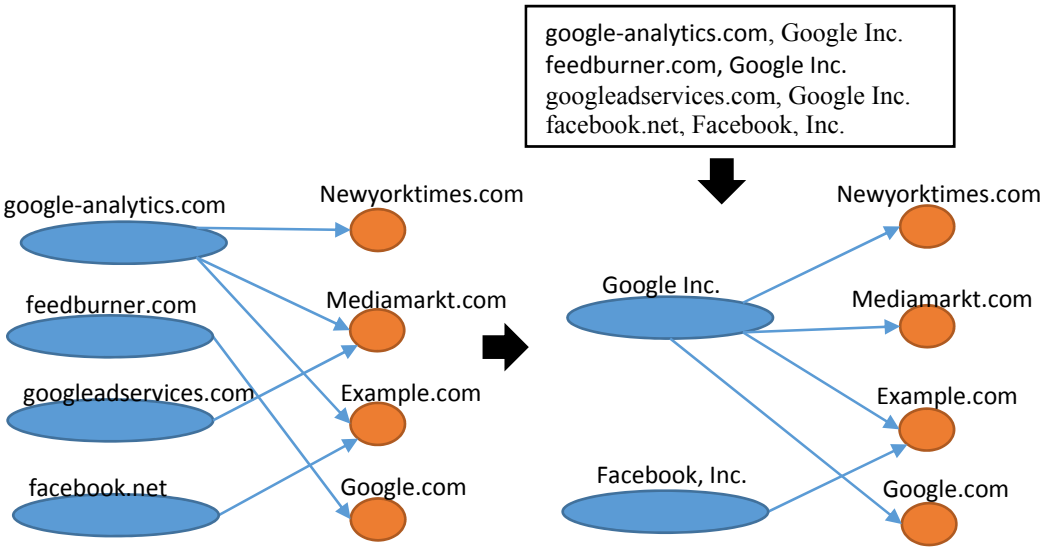

Figure 16: Aggregate to company level

3.3 User Intent of Third Party

Third party knows which content user is surfing, therefore it recognizes user intent. To simulate this situation, we exploit first party's category. For example, a first party in health category means it leaks the intent of health about the user. Knowing this intent helps third party to provide personalized services or advertising whereas also raises privacy concerns.

3.3.1 Category of First Party

Classifying first party based on their content provides the information about what third party can observe. Different content has different price and privacy hazard. Alexa classifies websites into fourteen categories, which are news, kids_and_teens, sports, business, recreation, health, home, reference, society, science, games, adult, arts, computers and shopping in Figure 17[32]. There are sub-categories in each category but we use the main fourteen categories.

We view the domain as top-private domain which is consistent with the bipartite graph. For example, google.com/xxx becomes google.com. One website can also exist in two or more

categories. Alexa provides top 525 websites for free to access and we crawl these 525 websites for each fourteen category.

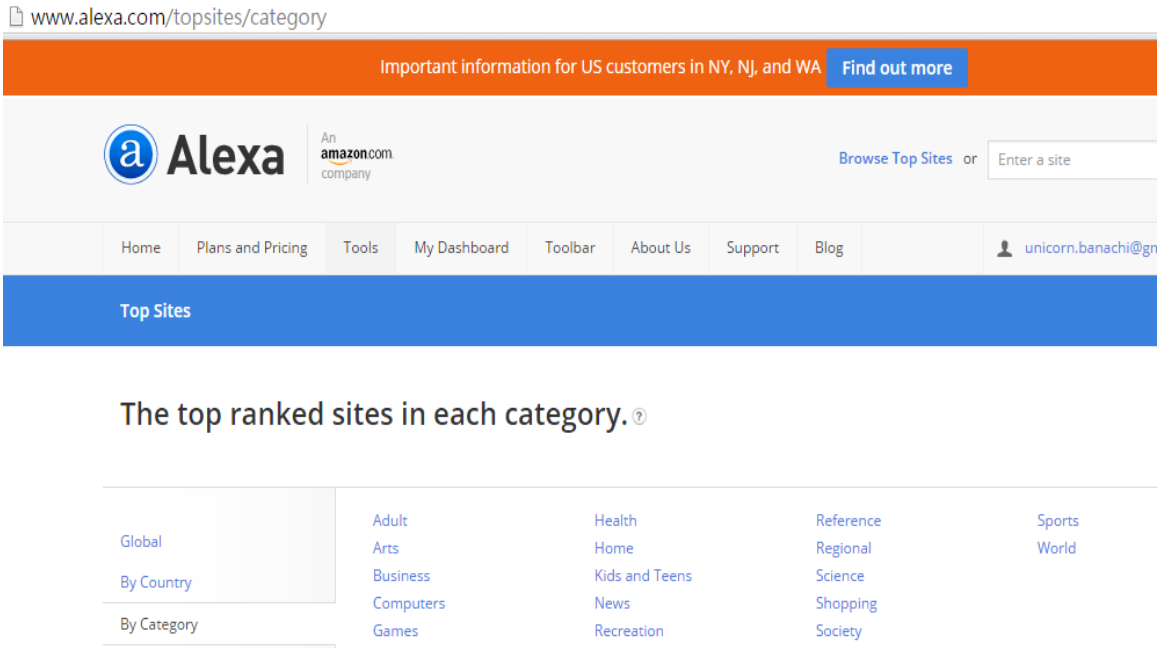

Figure 17: Alexa top category sites

3.3.3 Value of Intent in First Party

The value depends on the category of first party. Some user intents are more attractive and those are easier to become real purchase. Company bids those content for higher price, and those user intents are more valuable for third-party to collect and observe.

Google AdWords provides this kind of information in Keyword Planner[33]. It collects all the keywords' value as suggested bid. Company bids those keywords to have their ads shown in Google's ad network. This suggested bid shows the value and the price of the information

There are two types of suggest bid as shown in Figure 18 and Figure 19. Keyword ideas panel has the value for each keyword and here we put the name of category. Those values are estimated from ad group ideas panel. Ad group ideas are the value for popular related keywords combination. This thesis uses the value in keyword ideas rather than ad group ideas to represent a category's value. We extract the value by putting category name as shown in Figure 18.

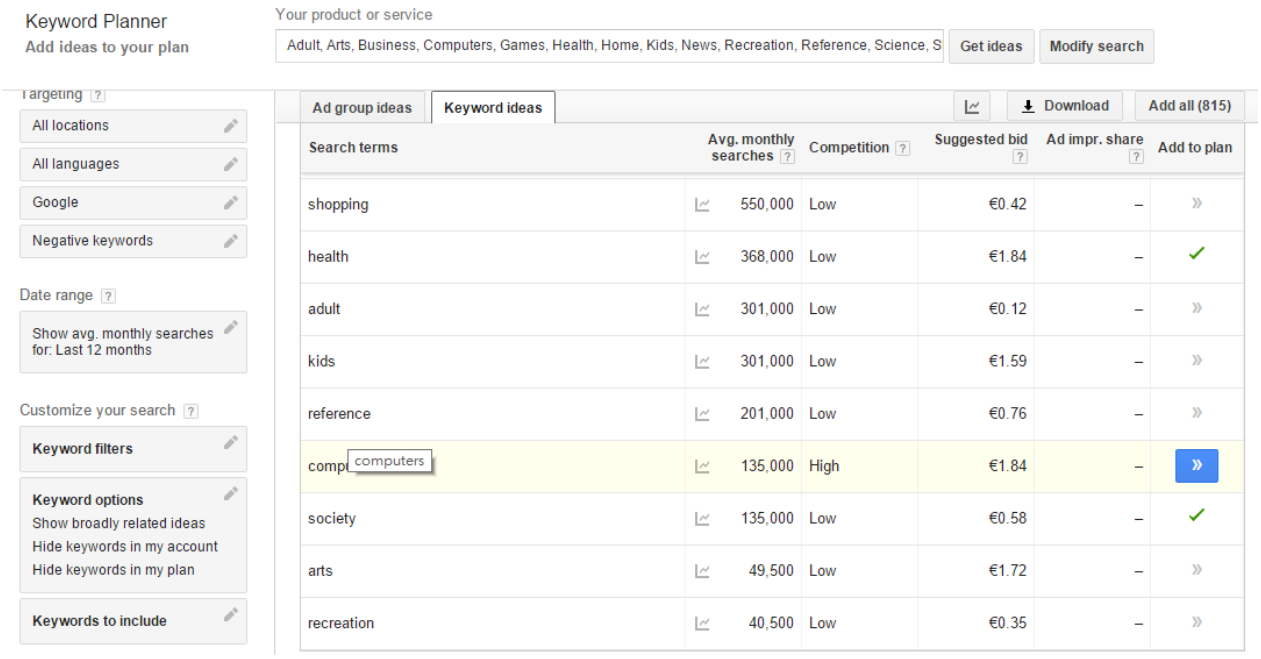

Figure 18: Keyword ideas

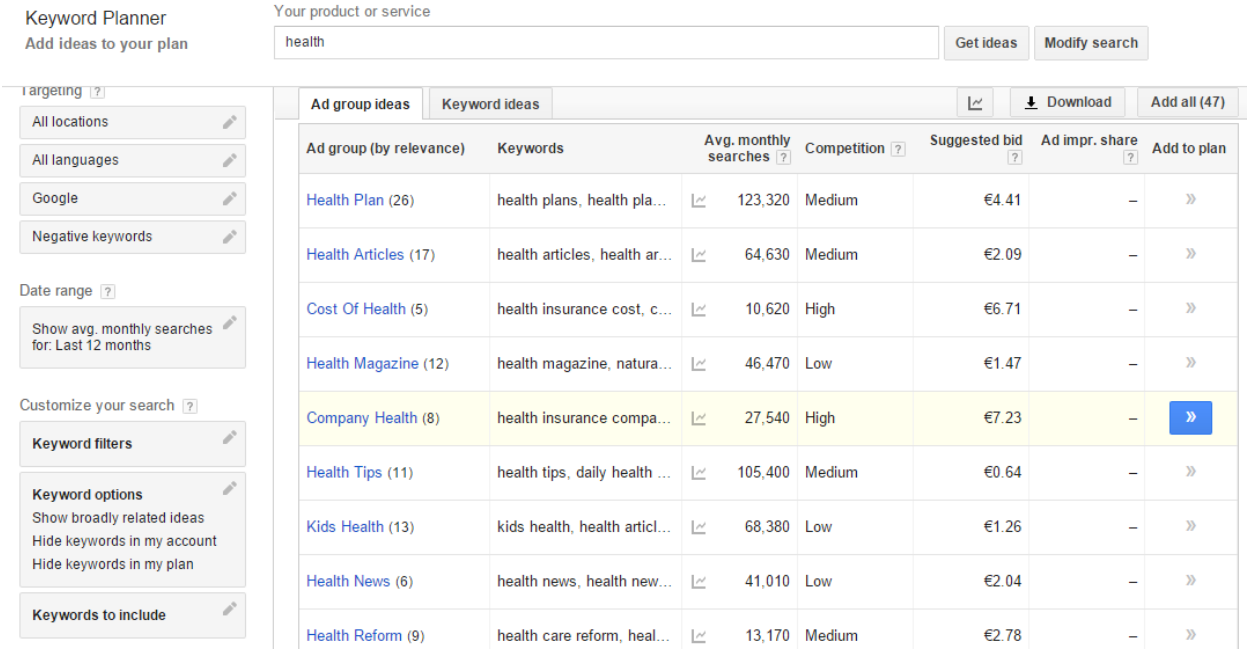

Figure 19: Ad group ideas

3.3.4 Privacy Hazard Index

There is another value for category which is the privacy hazard. The first party in health category contains high privacy hazard. It leaks health related data and is extremely sensitive. In the law describing privacy, health data are mentioned most frequently and geolocation are the second[34]. Researchers also had shown similar finding[15].

We score the privacy hazard as designing questionnaire by given health as 5 points due to its highest sensitive. We score recreation as 3 points as its medium sensitivity. Giving others as 1 point because they still at least contains information about users.

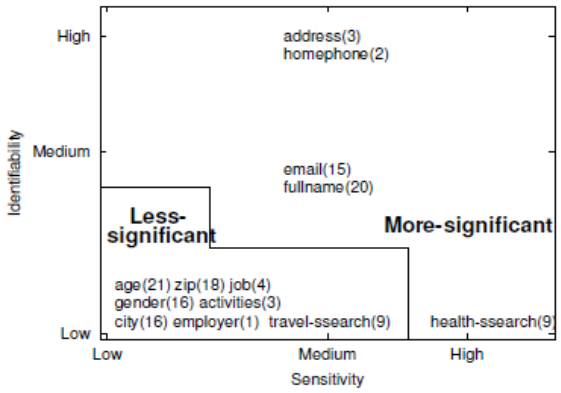

Figure 20: Data sensitivity[15]

3.4 Computing Web Traffic

An important website has higher popularity, more daily page views and collects more web traffic. From graph point of view, centrality measures the importance of website, therefore we use centrality to quantitatively measure the web traffic of the website.

We sum the centrality of each first party embedded by the same third party as sum of centrality. There are two centrality measures including PageRank and harmonic closeness. One important point is to firstly have the graph at company level. It's totally different than summing the centrality by firstly having sum of centrality at domain level, and then summing it up. The

latter case overestimates this statistics. The following example in Figure 21 and Figure 22 illustrate the difference.

In Figure 21, the computation firstly obtains the sum of centrality, and then sums it up if the third party is in the same company. Thus, the sum of centrality of Google is 1+2+103+400+400. While in Figure 22, Google's sum of centrality is1+2+103+400. The mistake is out of counting "Blogspot.com" twice.

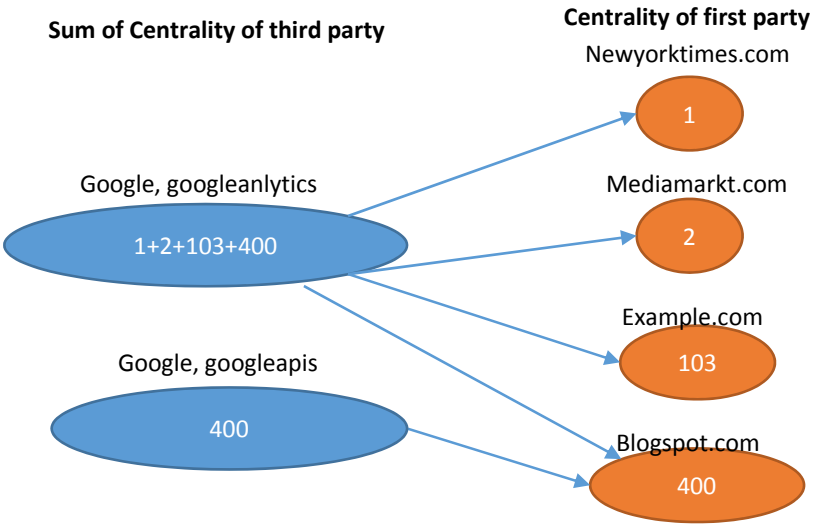

Figure 21: Wrong computation of sum of centrality

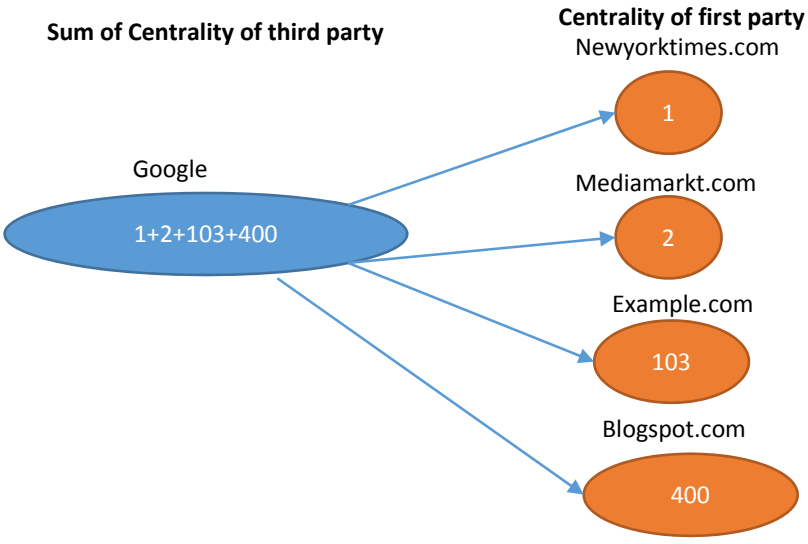

Figure 22: Correct computation of sum of centrality

3.5 Computing Run of Network

The terminology run of network derives from online advertising which means the ability of being embedded by first party[10]. Being embedded more often means this third party has higher revenue because it can send more ad and receive more data for sure. We simulate run of network by computing centrality in bipartite graph in only one mode that a graph contains only third party and excluding all first parties. The centrality of one-mode graphs measures the importance of third party.

There is no centrality algorithm for bipartite graph but there is centrality algorithm for one-mode graph. Thus, we transform bipartite into undirected weighted graph called one-mode projection. Then, we run weighted PageRank algorithm. The overall process to computer run of network ability is in Figure 23. The first two steps are for one-mode projection, and the last step computes centrality.

Figure 23: Run of network computation

3.5.1 Significant One-mode Projection with Resource Allocation

Many centrality algorithms are designed for one-mode graph whose nodes are one kind but not two kinds like bipartite graph. Thus, analyzing the bipartite graph usually needs to transform it into undirected weighted graph. This process is called one-mode projection. The undirected weighted graph is generated, and an edge exits between two nodes if two nodes sharing the same node in the bipartite graph. Applying to the bipartite graph in the web, it means an edge exists between two third parties if they are embedded in the same first party.

Figure 24 shows the process that transforming bipartite graph into undirected weighted graph using a naïve weight computation. Third-party F and L share one first party and in the undirected weighted graph, and the edge weight is 1. Third party L and A share two first parties, therefore, in the undirected weighted graph the edge weight is 2.

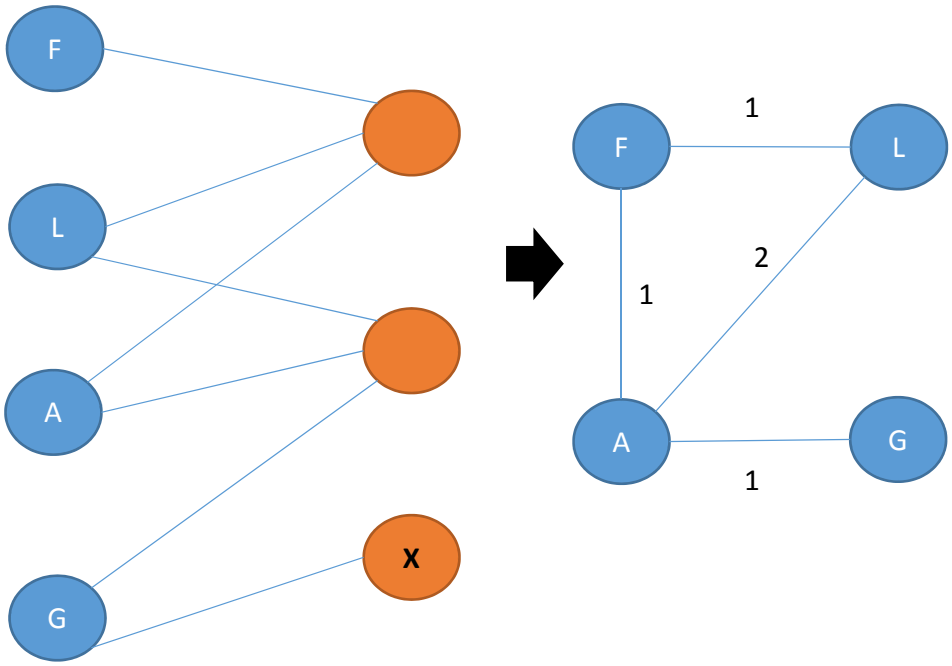

Figure 24: Naïve one-mode projection

Different one-mode projection focuses on different weight computation method. The above naïve weight computation has two main drawbacks. First, first party X in Figure 24 doesn't contribute to the transformation and is ignored. We know that third-party Google embeds in many first parties alone that no other third parties observe those first parties. This naïve computation definitely underestimates Google's ability. Second, this approach assumes that every node has the same power and doesn't consider their degree or centrality.

The more advanced approach is to use resource allocation to compute the edge weight and it solves the above two drawbacks[35]. Resource allocation is to compute weight to each node at one side based on the bipartite graph structure consists of two phases. At first step, this algorithm let the nodes at one side send weight based on the degree. Then, the nodes at the other side sends the weight back based on the degree with the weight they received.

Figure 25 demonstrates an example. Node F, L and A send resource to their neighbors based on the degree of nodes. The top right node receives *F+L/2+A/2*. Then, that top right node send the received resource back. Node F receives *(F+L/2+A/2)/3*.

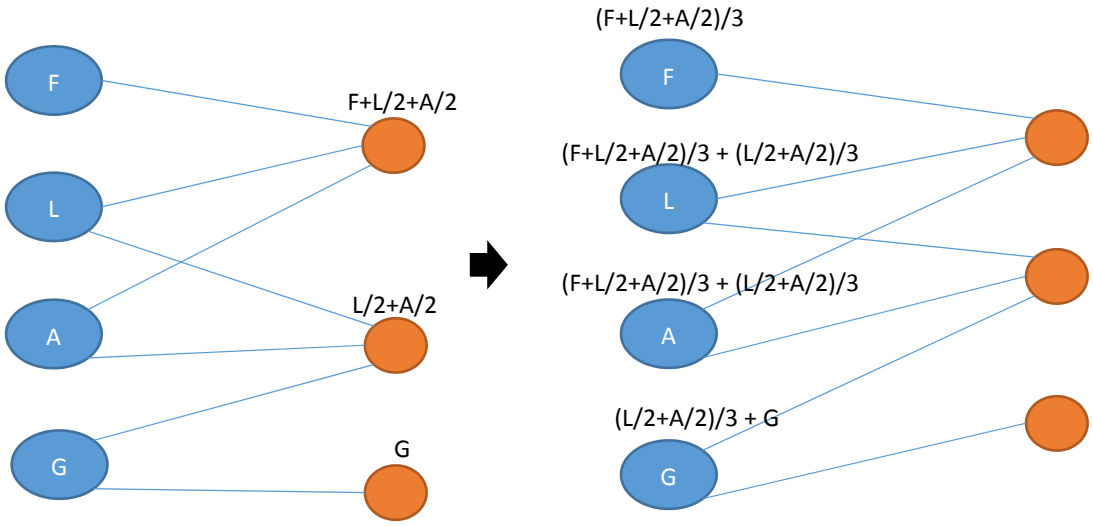

Figure 25: Resource Allocation

One-mode projection with resource allocation keeps the original structure in bipartite graph the most accurate among other approaches. First it runs the resource algorithm to recognize the resource in each node. After obtaining the resource for each third party, it generates edge based on the definition that an edge exists if two nodes share the same node.

Nevertheless, this edge generating step produces too massive edges and it's not suitable for a large graph. In other words, this step generates edges for every node' neighbor in one mode. Therefore, we prune the edges by applying hypothesis test in one-mode projection[36]. The idea is as follows.

For an each edge to exist in undirected weighted graph, we compute the edge weight based on the sum of resource of two nodes in an edge, and add this edge to undirected weighted graph only if the edge resource is significant. The significance of edge is represented by Z-score, and we transform the edge weight into Z-score as the following formula. If Z score is greater than Z (0.05) we add edge to undirected weighted graph.

$$Z = \frac{Resource\ A\ +\ Resouce\ B - 2\ \times Mean_{Resource}}{2 \times Standard\ Deviation_{Resource}}$$

By this pruning techniques, we can efficiently generate an undirected weighted graph from a bipartite graph. We can view this approach as significant one-mode projection with resource allocation. This approach combines the advantages from two papers[35, 36].

3.5.2 Weighted PageRank Algorithm

We run weighted PageRank algorithm to measure the importance of node[37]. The algorithm computes a probability distribution, and represent the likelihood that a user reaches a specific web page by randomly clicking on links. The only difference between normal PageRank and weighted PageRank is the transition matrix. In the latter case, the probability from one node to the other is computed by the edge weight but not only the degree of node. Thus, at first we have to normalize the weight to ensure it follows the axiom of probability. Each weight is divided by the sum of weight to ensure that after weight normalization, the sum of weight is equal to one. The meaning behind this is high weight edge has higher probability to transfer.

In order to compute weighted PageRank, we run the algorithm with iterative computations. Normally, dead ends that nodes having no outgoing edges prevents the algorithm to find stead-state probability, and we must modify dead ends to have self-loop, called as dangling PageRank algorithm[38]. Because the input of our weighted graph algorithm is undirected graph that is definitely stochastic, we can avoid dead ends.

3.6 Computing User Intent

After extracting the category data, we have a mapping that showing some first parties belong to which categories. Also, we have two files that one presents the value that the category earns, and the other presents the privacy that the category harms. Now we integrate the three files with the bipartite graph at the company level.

When computing web traffic, we use the centrality attribute in first party and now we are utilizing the attribute category and the value of category. The progress joins the category and first party at the beginning, and replaces the first party by its category. After obtaining the category,

joining with the category value, and then finally aggregates the value based on the third-party company.

Figure 26 shows an example to compute user intent. With the bipartite graph and domain's category information, we obtain a bipartite graph composed of category and third party. For example, Google observes health, shopping, games and recreation category because Google is embedded in "allinahealth.org", "homedepot.com", "simcity.com" and "easyjet.com".

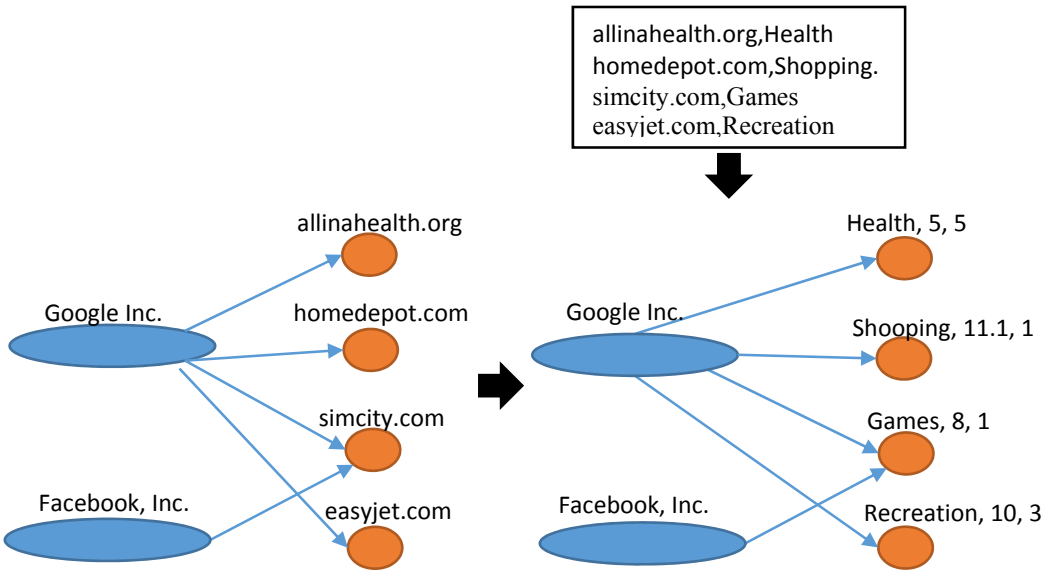

Figure 26: Computing user intent

After joining with category information, we start to use the value represents the category. Suggested bid and privacy hazard are two values used to present quantitative value of the category in Figure 26. Then, we aggregate suggested bid and privacy hazard of category for each third-party company.

We apply distinct function for company and category to realize how wide a third party can see[12]. In other words, if a company can see health category, we ignore other health categories and call it as user intent in breadth. We invent user intent in depth representing how deep third party can see which means running the computation without distinct function. Regarding privacy hazard, we also apply the depth concept.

4. IMPLEMENTATION

This section describes the implementation[3] for the approach in chapter 3. This thesis processes the graph and computes the graph statistics in parallel with Apache Flink. Each individual task takes satisfying time less than an hour with laptop using maximum 4 GB RAM. Most issues derive from the skew data that Google appears too frequently at the third-party side. It throws exceptions including “divide by 0” when building hash table and “too many duplicate keys”. We handle it by configuring join hint and higher memory. Besides, if the job is involved in many operators, we write the temporary result to disk and read again to repartition the data.

4.1 Processing Bipartite Graph in Flink

Processing graph is the step before graph statistics computation. We transform and process the graph to the format needed for computing graph statistics and accelerate computations.

4.1.1 Node Index and Edge Index

To accelerate graph processing, we first want to generate an edge file and a node file in long type to represent graph. Most of time processing graph manipulates edge file and processing string type is absolutely lower than processing long type. Edge file contains one index as third party and the other index as first party.

The raw graph file consists of string and index tuples, for example, “32 feedburner.com“ and it means third-party “feedburner.com“ embeds in first-party “32”. First party in the raw graph file represented as index which is from WebDataCommons node file. Thus, to know the name of first party, it’s needed to join the node file that tuple represents domain name and id, for example, a tuple “newyorktimes.com, 1”.

[3] https://github.com/HungUnicorn/trackthetrackers

Figure 27 shows the overall process generating the edge index and the node index. The input is 3.3 GB file and the outputs are 1.3 GB edge file and 423 MB node file, and it takes about 27 minutes.

At the beginning, due to availability of company information, we filter third party on the top level domain that focusing on .net, .com, .org because they are business related top-level domains.

Second, UnifyGranularity mapper tackles the problem that Google library cannot aggregate some google domains to pay-level domain (Goggle library calls the method topPrivateDomain()). For example, "blogspot.com" in the graph file appears in "xxxxxxx.blogspot.com" and "yyyyyyy.blogspot.com". This causes the granularity problem that those domains are at lower granularity whereas other domains are at higher granularity. We handle this problem by taking the last two tokens in the string and it's only suitable for our case. Our processing domains in this phase have no ccTLD and the last two tokens are definitely pay-level domain. However, in general the last two tokens in ccTLD is definitely not the case. Then, we take distinct edges due to the domains that Google library cannot aggregate have duplicates.

Now we have clean edges contain (string, long) tuples. Projecting the string column and obtaining distinct third party names produces node index. Joining this node index with edges creates edge index which replaces string type column in the tuples by long type.

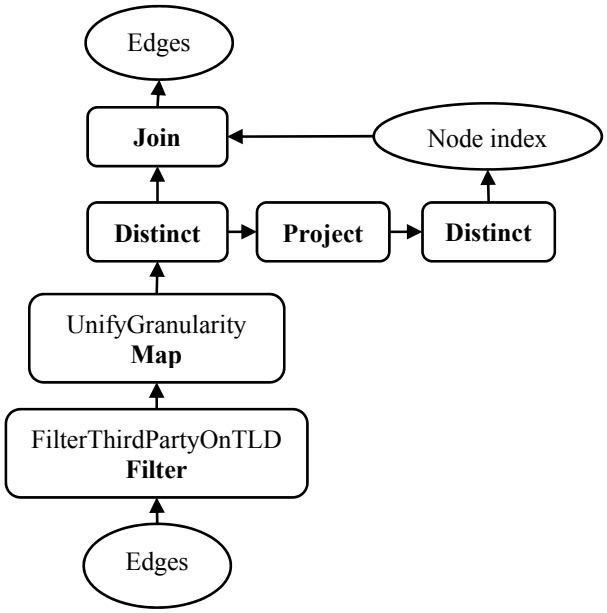

Figure 27: Generated edge file and node file

4.1.2 Popular Third Parties for WHOIS Crawler

For efficiently crawling WHOIS information to recognize domain's company, we only access third party who has higher sum of centrality because WHOIS server forbids massive crawling. Thus, the program seizes the top third parties in descending order based on their sum of centrality.

Figure 28 displays the complete process. The inputs are first party's centrality that are 4.5 GB PageRank, 3.3 GB harmonic centrality, 1.3 GB edge file and 423 MB node file. The output are 3.8 MB top PageRank third parties and 3.1 MB top harmonic centrality third parties. This task takes around 38 minutes for 2 times that one for computing PageRank and the other for harmonic closeness.

We first joins edge index with first-party centrality computed and published by WebDataCommons. This gives us tuples consist of third party, first party and first party's centrality. Then, the program aggregates the centrality group by third party. We require the top third parties by centrality. Before utilizing FirstN, CentralityFilter filters third parties who have lower centrality which is a tuned parameter. This filtering accelerates the processing.

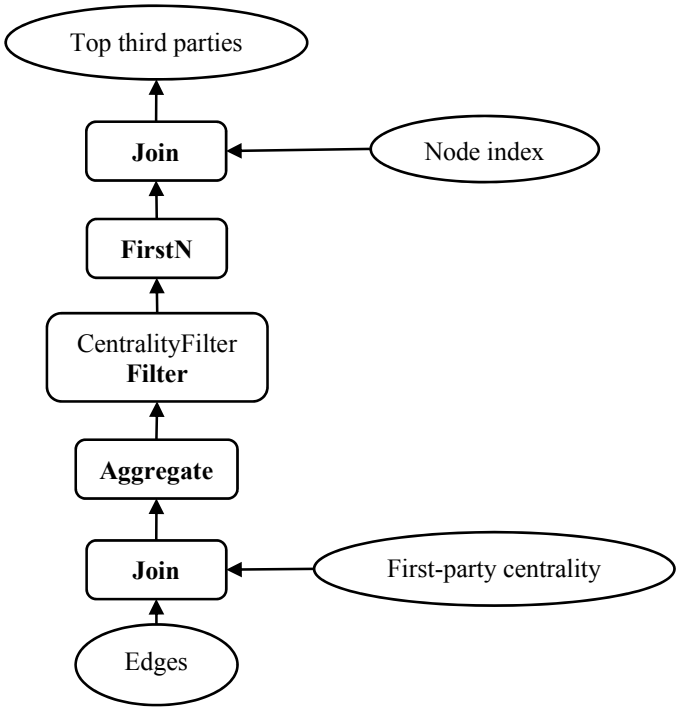

Figure 28: Gathering top third parties

4.1.3 Aggregating to Company Level

Aggregating the edges to company level is the most expensive distinct() among other process. This is extremely skew containing many remarkably high-degree nodes, for example, Google and Facebook in our case. The data is already skew at domain level, and we are going to aggregate to company level and make the data more skew.

Especially, Flink adopts vertex-parallel computation and high-degree nodes is a pain for such computation. Without writing the file before exploiting distinct() the program encounters too many duplicate keys exception. The program also applies JoinWithTiny() and different direction of join[39] to avoid the exception. We also build node index which is the company index for these edges. The inputs are 1.3 GB edge file and 2.5 MB domain's company mapping, and the output is 372 MB edge file. This task takes 9 minutes. The whole process is as Figure 29.

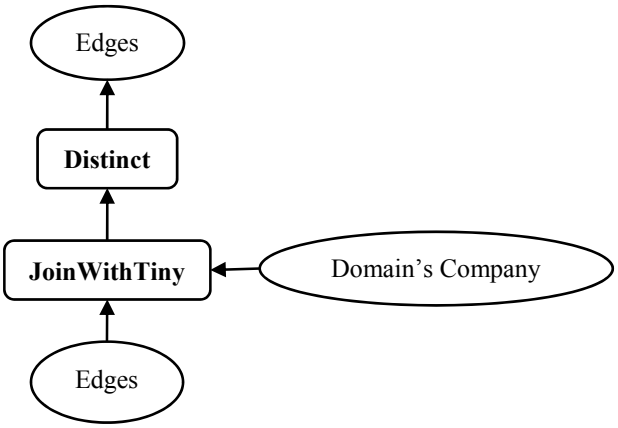

Figure 29: Aggregating to company level

4.1.4 One-mode Projection

Adopting one-mode projection projects an undirected weighted graph from a bipartite graph for computing run of network. The overall process requires two steps. The first step is to compute the node resource, and then utilizing the node resource and the bipartite graph to create undirected weighted graph. We exploit resource allocation algorithm for one-mode projection, and executing resource allocation is the first step. This node resource also represents as one kind of graph statistics of run of network. Figure 30 demonstrates the overall process of the first step which is resource allocation. The input is 372 MB edge file and generates 199 KB node resource file, and it takes about 18 minutes.

First, from a bipartite graph (X, Y) the goal is to project an undirected weighted graph composed of only X and without any Y. We want to distribute weights from X to Y, and then from Y back to X. In the implementation we generate node x's neighbors Y for each node x, and node y's neighbors X for each node y for distributing resource. Next, we assign initial weight and distribute the weights to each node x's neighbor which are nodes Y only. Then, aggregating the weights based on nodes Y to represent the resource that nodes Y contain at the first phase.

In the second distributing phase, we exploit node Y's weights and distribute it back to nodes X by joining node Y's neighbors with Y's resource. We again aggregate based on node X and this is the resource of X which we need for the next process.

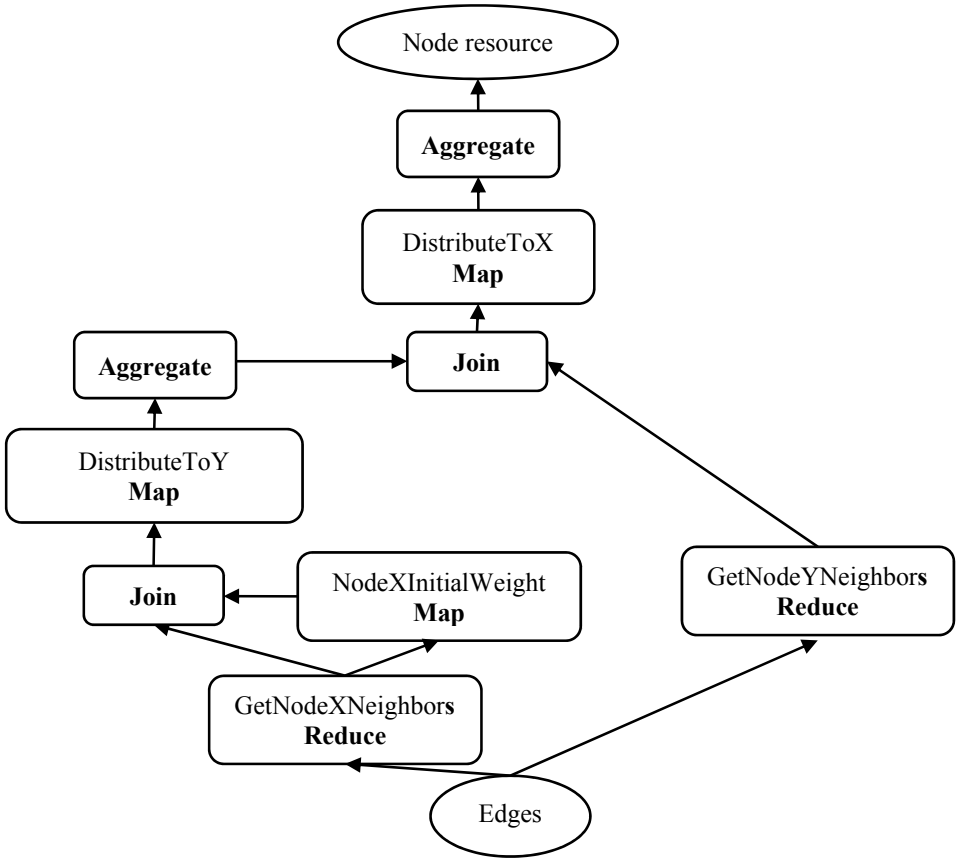

Figure 30: Implementation of resource allocation

After obtaining the node resource, we start to use the node resource to generate edges significantly. Figure 31 displays the whole process. It takes 372 MB edge file and 199 KB node resource file as input and generates 3.5 MB file as output for 3 minutes. According to the definition of our significant one-mode projection with resource allocation, an edge exists if the edge resource is significant. We first generate node Y's neighbors which are the nodes won't appear in the undirected weighted graph, and they are first parties. Next, Addsignificantedges is the most important step, and we add edges for each node Y's neighbors significantly using Broadcast Join with node resources. Then, some null edges appear because some nodes Y doesn't produce any edge. Finally, we aggregate the weight of edges and obtain an undirected weighted graph.

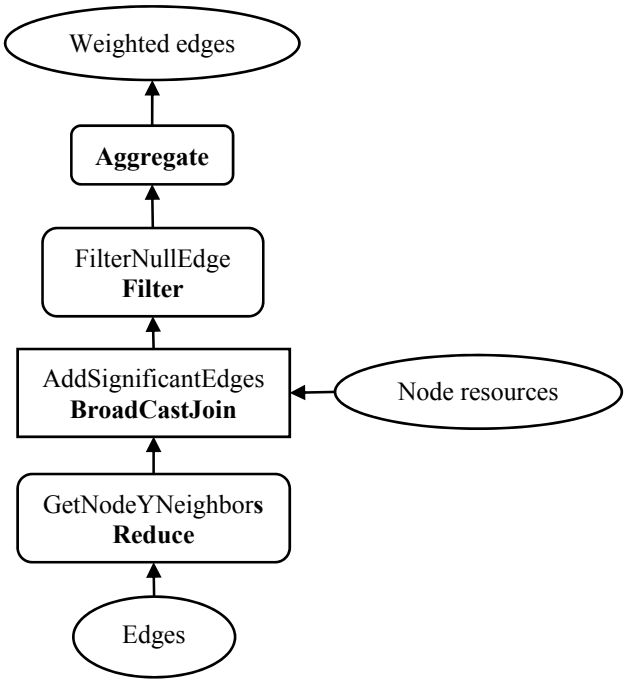

Figure 31: Significant one-mode projection with resource allocation

4.2 Graph Statistics Computation in Flink

After having the edges at company level in adequate format, we start to compute graph statistics for each third-party company.

4.2.1 Web Traffic

The web traffic computation is similar to the process gathering top third parties by sum of centrality. Figure 32 shows the process of computation. The inputs are first party's centrality that are 4.5 GB PageRank and 3.3 GB harmonic closeness, 372 MB edge file and 216 KB company index file. The outputs are 312 KB company's sum of harmonic closeness and 352 KB sum of PageRank. This task takes about 19 minutes for 2 times that one for computing PageRank and the other for harmonic closeness.

Edges join with first-party centrality and aggregate the centrality group by the company. PageRank and harmonic closeness are two kinds of centrality we exploit. The execution time at company level is much quicker than the same computation at domain level.

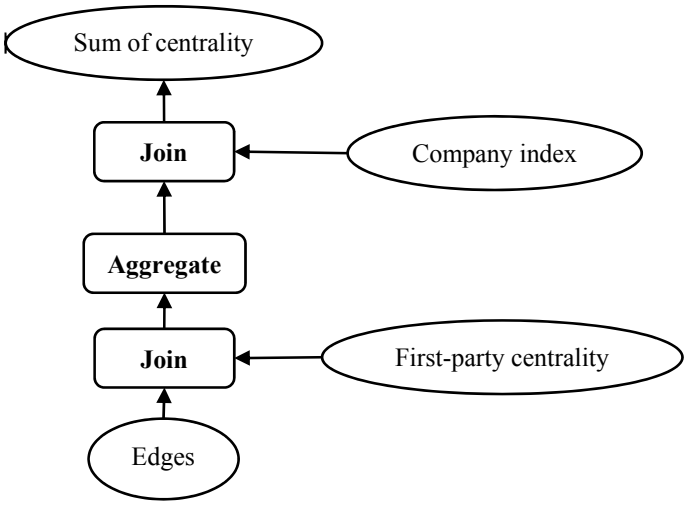

Figure 32: Implementation of web traffic statistics

4.2.2 Run of Network

Computing run of network requires one-mode projection. After having undirected weighted graph, we compute weighted PageRank on undirected weighted graph. Figure 33 demonstrates the whole process. The input is 3.5 MB weighted edge file and the output is 187KB, and it takes 20 seconds.

The implementation reads weighted edges and project nodes from edges. Normalizing edges weight and initializing PageRank for each node is the pre-process for running iterative computation. The implementation of weighted PageRank exploits Flink's graph library that sending the PageRank as message and the nodes who receive message must update their rank, and

then they send message to their neighbors. The maximum iterations is 20.

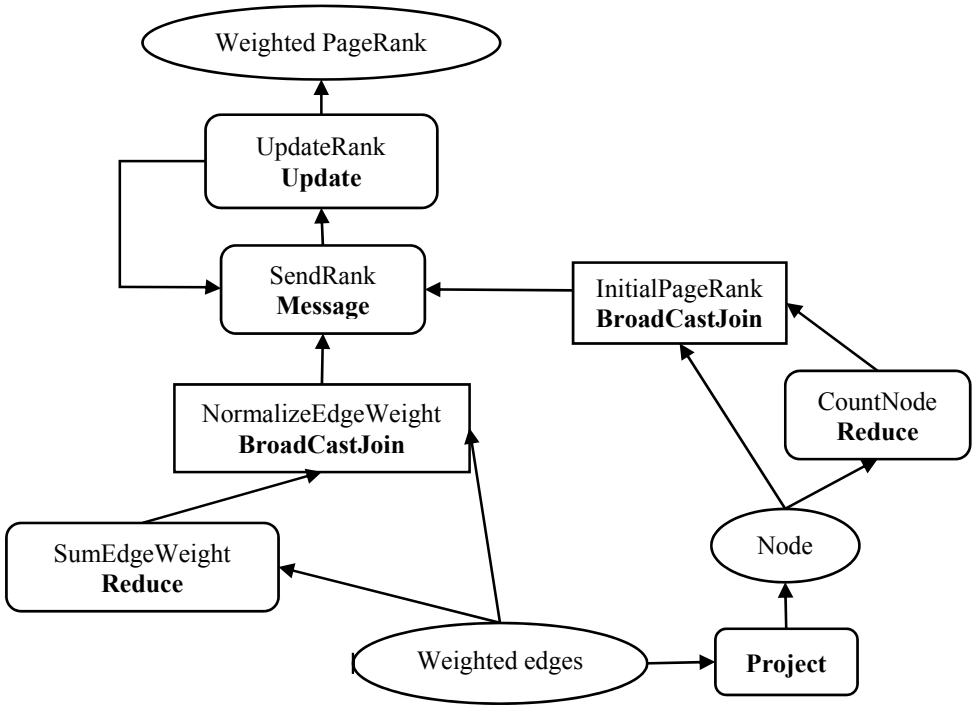

Figure 33: Implementing weighted PageRank

4.2.3 User Intent

The user intent computations read data from Alexa and Google AdWords and utilize many join operators. The size of joining data is extremely different and the implementation uses JoinWithHuge and JoinWithSmall as a hint to join.

The whole process executes as Figure 34. First, category of first party joins with first-party index and the latter is much larger than the former. Then, edges join with category of first party and distinct() is executed if considering breadth, and considering depth the distinct() is skipped. After that the implementation starts to join with tiny file of category value which are suggested bid from Google AdWords and sensitivity of data. Next, aggregating third-party value generates user intent.

The input is 372MB edge file, 1.1 GB first-party node index file, 159KB Alexa top 525 web category file, 46 KB Google AdWords keywords value file. The output is three 99 KB user intent files including user intent breadth and depth and privacy hazard. This takes about 2 minutes for one kind of user intent.

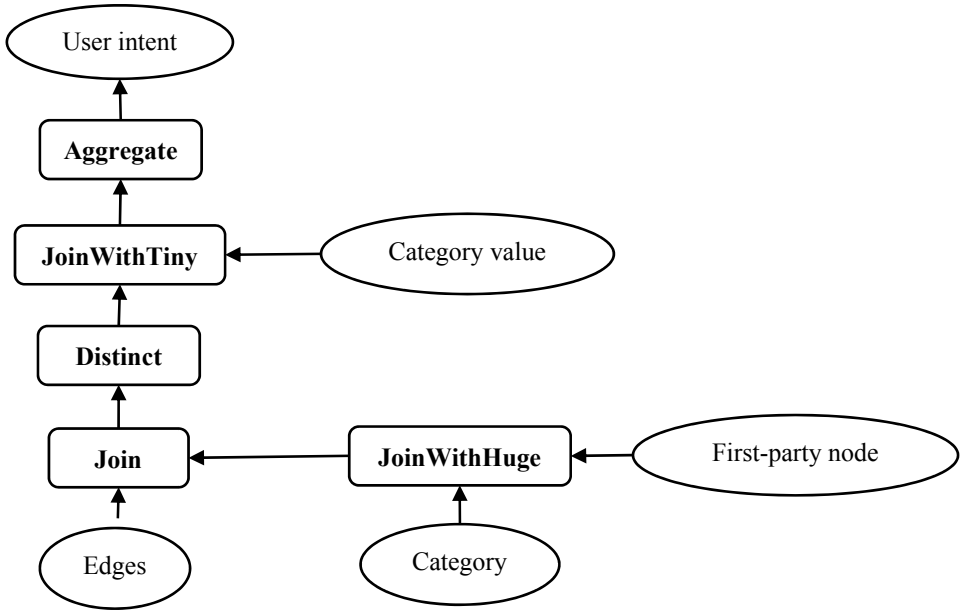

Figure 34: Implementing user intent

5. RESULTS

Here we show the evaluation and the results we obtained based on the approach and the implementation in chapter 3 and chapter 4. We evaluate our result with the other data by correlation and regression analysis, and conduct regression analysis for explaining the relationship between web traffic, run of network and user intent symbolized by sum of PageRank, sum of harmonic closeness, node resource, weighted PageRank, user intent breath, user intent depth and privacy hazard.

5.1 Graph Statistics

There are three kinds of computation which are based on web traffic, run of network and user intent. PageRank and harmonic closeness are two measures in terms of web traffic, node resource and weighted PageRank are two statistics related to run of network, and user intent depth, user intent breath and privacy hazard are three statistics regarding user intent.

5.1.2 Web Traffic

Usually daily page views represent the web traffic of the website, and due to the unavailability of massive page views information, we use the centrality in hyperlink graph to represent the web traffic of website and aggregate the centrality for each third party in bipartite graph.

Table 3 lists the top 20 third parties by their sum of PageRank and sum of harmonic closeness. Google dominates web traffic in terms of two measures and Facebook is at the second place in these two measures. However, other third parties can't hold their position in the two measures. For example, Twitter wins Adobe by sum of PageRank and loses by sum of harmonic closeness. eBay also has high sum of harmonic closeness and low sum of PageRank comparing to Microsoft and Amazon. The unit of measurement is extremely different that sum of PageRank is probability and sum of harmonic closeness counts the average hops.

After knowing top companies having web traffic, we begin to look at the distribution of web traffic by scatterplot. Figure 35 shows the distribution is remarkably skew from the view of sum of PageRank and sum of harmonic closeness. Google locates at the top right and has the remarkably highest sum of PageRank and sum of harmonic closeness. Most points locate at the bottom left and Google dominates web traffic.

Company	Sum of PageRank	Sum of harmonic closeness
Google Inc.	0.32866	1.36E+14
Facebook Inc.	0.145993	2.85E+13
Twitter Inc.	0.109653	1.43E+13
Adobe Systems Incorporated	0.103175	2.46E+13
Yahoo! Inc.	0.089169	1.13E+13
Automattic Inc.	0.079569	1.03E+13
AddThis Inc.	0.077985	1.02E+13
Microsoft Corporation	0.056852	2.64E+12
Amazon.com Inc.	0.056379	2.93E+12
Photobucket Inc.	0.050008	2.83E+12
eBay Inc.	0.049518	6.02E+12
Vimeo LLC	0.049375	2.67E+12
Amazon Technologies Inc.	0.048152	3.94E+12
AOL Inc.	0.045846	1.13E+12
Wikimedia Foundation Inc.	0.040324	1.04E+12
LinkedIn Corporation	0.040307	2.22E+12
Akamai Technologies inc.	0.039943	3.23E+12
Apple Inc.	0.039615	1.07E+12
Tumblr Inc.	0.036862	1.10E+12
DNStination Inc.	0.03681	6.72E+12

Table 3: Top 20 third parties by sum of PageRank and sum of harmonic closeness

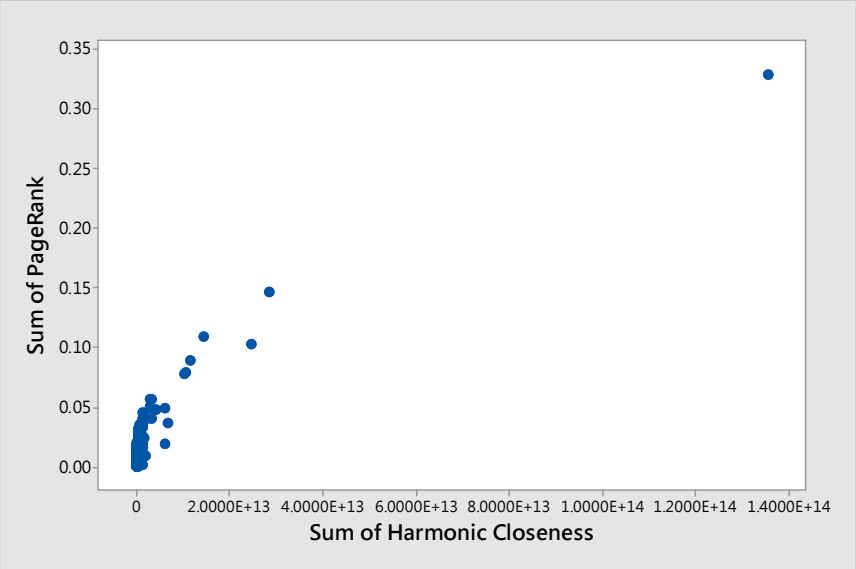

Figure 35: Scatterplot of sum of PageRank vs sum of harmonic closeness

5.1.2 Run of Network

Run of network exemplifies the ability of third party being embedded by first party, and we exploit the structure in bipartite graph to illuminate the effect of run of network. During one-mode projection we obtain node resource, and after one-mode projection we compute weighted PageRank in undirected weighted graph.

Table 4 lists the top 20 third parties by weighted PageRank and node resource. Google monopolizes in terms of the two measures. Facebook is at the second position in both measures. Go Daddy acquires high weighted PageRank but low node resource which means it connects to high PageRank node results of PageRank computation. In fact, Go Daddy is a company provides WHOIS privacy and isn't a third party, as well as DNStination Inc. Same situation also happens to BlueHost which is a company helps to register the domain name in WHOIS.

Figure 36 shows the distribution of node resource and weighted PageRank. Again, both the distributions of node resource and weighted PageRank are remarkably skew. We can also find Google dominates run of network and locates at top right. Many other companies have comparatively smaller weighted PageRank and node resource.

Company	Weighted PageRank	Node resource
Google Inc.	0.424094832	884.3132196
Facebook Inc.	0.080253952	248.912805
Adobe Systems Incorporated	0.057113908	134.1901137
Go Daddy Operating Company LLC	0.041158264	15.13244964
Twitter Inc.	0.026321404	143.9730044
AddThis Inc.	0.025352281	110.162562
Automattic Inc.	0.020985661	135.3833975
Yahoo! Inc.	0.020614908	154.9487428
DNStination Inc.	0.011261429	35.93110621
eBay Inc.	0.010976315	74.13241644
Amazon Technologies Inc.	0.007328462	86.89461556
Akamai Technologies inc.	0.007234329	37.10946807
Magnetic Media Online Inc.	0.005527368	2.376917048
Microsoft Corporation	0.004155381	58.56836364
Amazon.com Inc.	0.003847653	54.23229462
Vimeo LLC	0.003649264	34.02081414
Photobucket Inc.	0.00334446	80.24802485
BLUEHOST INC	0.002799294	3.027114137
LinkedIn Corporation	0.002024063	29.16576996
OCLCOnlineComputer Library Center Inc.	0.001960293	4.848189323

Table 4: Top 20 third parties by weighted PageRank and node resource

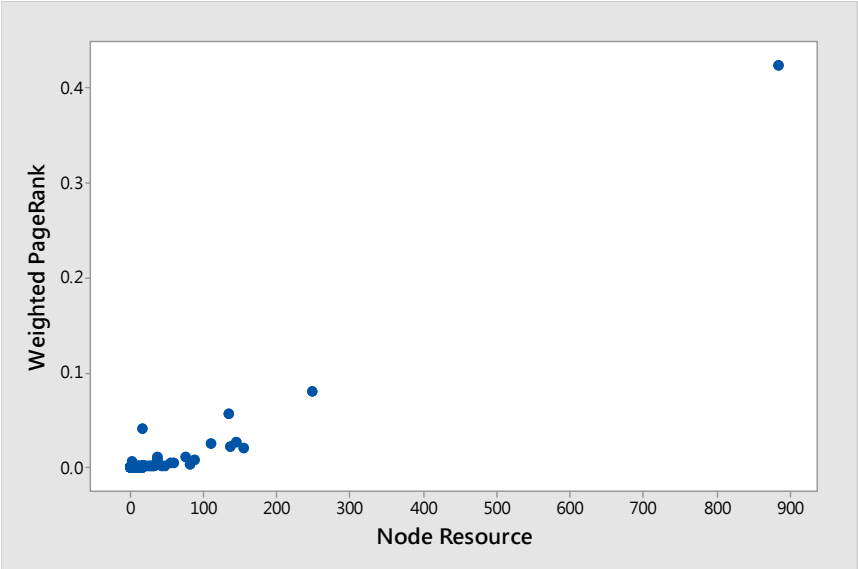

Figure 36: Scatterplot of node resource vs weighted PageRank

5.1.3 User Intent

User intent illustrates the ability that third party knows people's interest. We compute user intent according to the first party's category and the values of category. The values are the power driving real purchase and the sensitivity of data.

Table 5 lists the top 20 third parties by user intent breadth, user intent depth and privacy hazard. All top 20 third parties have the same value of user intent in breadth that they can see all first parties' categories. Google triumphs user intent in depth and privacy hazard. Akamai and Amazon has lower user intent depth but higher privacy hazard.

Company	User intent breadth	User intent depth	Privacy hazard
Google Inc.	13.47	5880.48	9162
Facebook Inc.	13.47	4234.29	6376
Twitter Inc.	13.47	3241.4	4786
Adobe Systems Incorporated	13.47	2892.09	4342
AddThis Inc.	13.47	2644.2	4165
Yahoo! Inc.	13.47	2245.81	3328
Automattic Inc.	13.47	1735.26	2477
Amazon.com Inc.	13.47	1478.97	2157
Microsoft Corporation	13.47	1425.88	2036
Akamai Technologies inc.	13.47	1218.95	1660
TMRG Inc	13.47	1202.16	1668
AOL Inc.	13.47	1199.67	1641
Amazon Technologies Inc.	13.47	1161	1646
Photobucket Inc.	13.47	1109.78	1614
Vimeo LLC	13.47	1020.82	1429
eBay Inc.	13.47	887.08	1346
LinkedIn Corporation	13.47	862.22	1114
Apple Inc.	13.47	763.7	1030
Brightcove Inc.	13.47	732.8	1023
Wikimedia Foundation Inc.	13.47	727.11	1033

Table 5: Top 20 third parties by user intent breadth, user intent depth and privacy hazard

We first show the distribution of user intent breadth by histogram. 9% companies has the same highest value because those companies can see all categories of first party in our data. User

intent breadth is not as skew as other graph statistics. We can also observe that 11% companies focus on certain categories of first party.

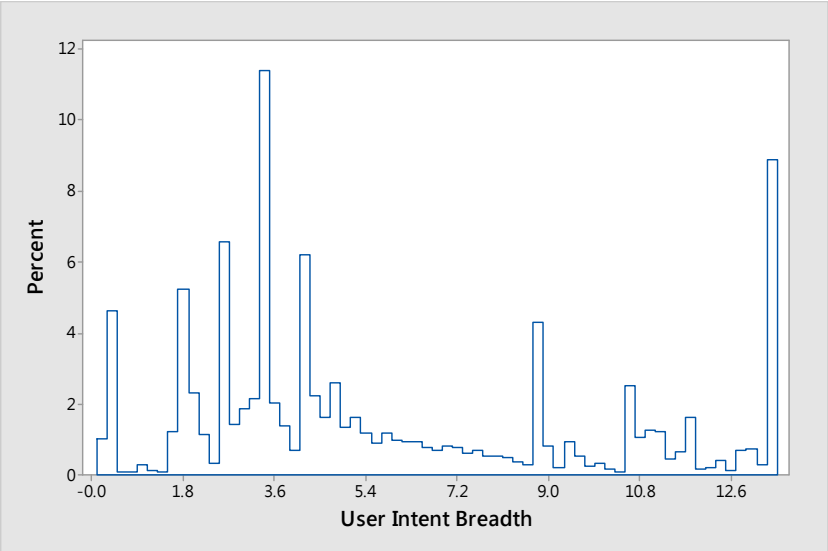

Figure 37: Histogram of user intent breadth

Next, we start to investigate the distribution of user interest depth and privacy hazard. In Figure 38 both distributions are still skew but not as skew as web traffic and run of network because the distances between the top points are much smaller. Google still dominates user intent depth and privacy hazard, but doesn't have the same dominating power as web traffic and run of network.

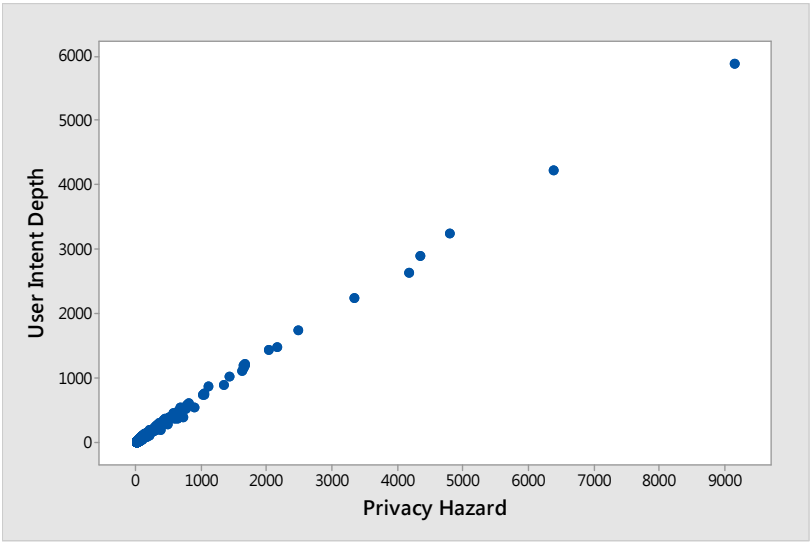

Figure 38: Scatterplot of user intent vs privacy hazard

5.1.4 Correlation of Graph Statistics

To examine the relationship we compute Pearson correlation of all statistics with correlation matrix. Before computing correlation we transform the data into log scale because Pearson correlation is highly sensitive for skew data and it's absolutely our case.

Table 6 shows the correlation matrix for seven graph statistics in three main categories which are web traffic, run of network and user intent. Reading this table from row and then from column allows us to know the correlation. For example, the correlation between node resource and sum of harmonic closeness is 0.815 which means they are highly correlated to each other in log scale. Although the correlation is not particularly high, the hypothesis test states that those statistics are correlated due to p-value is 0. The null hypothesis is two factors are independent and is rejected due to p-value is 0.

The highest correlation appears in privacy hazard and user intent (0.947), and the lowest correlation appears in sum of PageRank and sum of harmonic closeness (0.273). All user intent related statistics are highly correlated (greater than 0.7), and run of network related statistics have the same phenomena except that sum of PageRank and sum of harmonic closeness are not that highly correlated (between 0.3 and 0.7).

	Web traffic		Run of network		User intent		
	Sum of PageRank	Sum of harmonic closeness	Weighted PageRank	Node Resource	Breadth	Depth	Privacy Hazard
Sum of harmonic closeness	0.44	1					
Weighted PageRank	0.273	0.681	1				
Node resource	0.479	0.815	0.718	1			
Breadth	0.756	0.542	0.274	0.473	1		
Depth	0.78	0.691	0.461	0.656	0.922	1	
Privacy hazard	0.655	0.731	0.503	0.706	0.832	0.947	1

Table 6: Correlation matrix of all graph statistics at log scale

5.2 Evaluating Graph Statistics

To evaluate the graph statistics computed, we evaluate the graph statistics with revenue that all graph statistics except for privacy hazard should be highly correlated to the revenue and explain the revenue from web traffic, run of network and user intent which are effective factors for estimating revenue. Furthermore, if those graph statistics can explain the revenue with the regression model accurately, this indicates that we can further utilize them to predict the revenue with regression.

5.2.1 Comparing with Real World Revenue

We want to compare our graph statistics of third party with others' collected statistics. Our graph statistics represents the status of 2012 since the extracted graph is from the web crawl data that Common Crawl collected in 2012[1]. Many companies devote to release the information for top third parties, and charge people who want to access more information of third parties. Builtwith publishes recent months top popularity of third parties[40] and ValueClick reports the revenue of third parties each year[10]. None of the above two cases fit our requirements since we need the statistics in 2012 and not only for top third parties.

eMarketer fits our requirements the best that it publishes the digital advertising revenue each year for top 12 companies as shown in Figure 39[41]. The main profit of third-party web tracking is selling advertisement as the famous cases including Google and Facebook, therefore, comparing our graph statistics with the digital advertising revenue rather than the whole revenue of the company is more suitable. The information eMarketer publishes doesn't report many companies and only contain the information for top companies, however, through the comparison between our graph statistics and the revenue they release, we can understand how our graph statistics perform and evaluate them.

Figure 39 contains the information of the specific revenue but Pandora and Millennial Media are not in our data due to we only have the information for those companies obey WHOIS policy and keep clean information. This means we evaluate the graph statistics by ten instances.

Net Digital Ad Revenue Share Worldwide, by Company, 2012-2014
% of total digital ad revenues

	2012	2013	2014
Google	31.30%	31.92%	31.45%
Facebook	4.09%	5.82%	7.79%
Microsoft	2.44%	2.45%	2.54%
Yahoo	3.36%	2.86%	2.52%
IAC	1.34%	1.27%	1.04%
AOL	1.02%	0.94%	0.88%
Twitter	0.26%	0.50%	0.79%
Amazon	0.53%	0.63%	0.75%
LinkedIn	0.37%	0.47%	0.54%
Pandora	0.34%	0.43%	0.52%
Yelp	0.12%	0.18%	0.25%
Millennial Media	0.07%	0.09%	0.09%
Other	54.76%	52.44%	50.82%
Total digital (billions)	**$104.57**	**$120.05**	**$140.15**

Note: includes advertising that appears on desktop and laptop computers as well as mobile phones and tablets, and includes all the various formats of advertising on those platforms; net ad revenues after companies pay traffic acquisition costs (TAC) to partner sites; numbers may not add up to 100% due to rounding
Source: company reports; eMarketer, June 2014

174629 www.eMarketer.com

Figure 39: Digital advertising revenue of top companies

5.2.2 Correlation between Graph Statistics and Revenue

Before computing Pearson correlation we transform the data into log scale because Pearson correlation is highly affected by skew data. We first look at Pearson correlation between the graph statistics and the digital advertising revenue, and also consider privacy hazard to recognize the correlation as shown in Table 7. The value in the cell is the Pearson correlation and the value inside the parenthesis is the p-value testing whether two factors are significantly correlated. If the p-value is smaller than the threshold 0.1 we conclude that two factors are significantly correlated.

In the first column, we discover that sum of harmonic closeness, weighted PageRank and node resource are significant correlated to revenue because p-value is smaller than 0.1. All statistics of user intent are not significantly correlated to revenue.

The highest correlation exists in privacy hazard and user intent in depth which is equal to one. The factors correlated to revenue, if we take the threshold as 0.3, are sum of PageRank, sum of harmonic closeness, weighted PageRank, node resource, user intent depth and privacy hazard.

		Web traffic		Run of network		User intent		
	Revenue	Sum of PageRank	Sum of harmonic closeness	Weighted PageRank	Node Resource	Depth	Breath	Privacy Hazard
Sum of PageRank	0.51 (0.109)	1						
Sum of harmonic closeness	0.572 (0.066)	0.959 (0)	1					
Weighted PageRank	0.606 (0.048)	0.977 (0)	0.984 (0)	1				
Node resource	0.534 (0.091)	0.949 (0)	0.974 (0)	0.949 (0)	1			
Depth	0.362 (0.274)	0.971 (0)	0.914 (0)	0.915 (0)	0.932 (0)	1		
Breadth	0.051 (0.881)	0.631 (0.037)	0.704 (0.016)	0.615 (0.044)	0.653 (0.029)	0.66 (0.027)	1	
Privacy hazard	0.353 (0.286)	0.970 (0)	0.914 (0)	0.914 (0)	0.928 (0)	1 (0)	0.673 (0.023)	1

Table 7: Correlation between digital ad revenue and graph statistics at log scale

5.2.3 Regression Analysis for Revenue

To understand how our graph statistics explain the revenue, conducting regression analysis delivers it by demonstrating a formula. We can further utilize this formula to predict the change of revenue by increasing or decreasing 1 unit of the graph statistics. The dependent variable is revenue and the independent variables are the graph statistics which is shown as a formula below.

$$Revenue = c + b_1\, Sum\ of\ PageRank + b_2\, Sum\ Of\ harmonic\ closeness + b_3\, Weighted\ PageRank + b_4\, Node\ resource + b_5\, User\ intent\ depth + b_6\, User\ intent\ bredth + b_7\, Privacy\ Hazard$$

All b_i are coefficients serves as the change of 1 unit brings to the revenue, for example, 1 unit increasing or decreasing of Sum of PageRank will cause b_1 increasing or decreasing of revenue, and the intercept c leads to the regression predicts more accurately.

By examining our graph statistics and revenue, all of them are extremely skew especially Google dominates all statistics. Without any data transformation the regression predicts Google's revenue accurately whereas predicts others' revenue inaccurately. Figure 40 reveals it by showing

the residual and the fitted value. Residual is the difference between the real value and the predicted value and the fitted value is the predicted value. Google is the data point at fitted value 30 and residual 0 which means the regression predicts Google's revenue accurately and predicts others' revenue inaccurately.

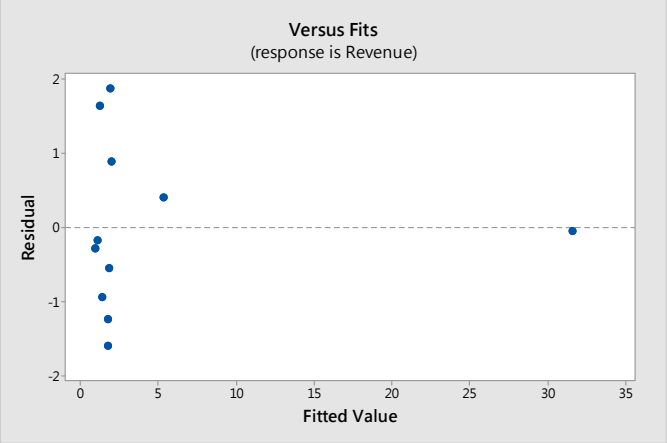

Figure 40: Residual versus fitted value

Logarithm transformation solves the above problem by taking logarithm function for each variables. It's called regression at log-log level and the regression formula becomes as below.

$$Ln(Revenue) = c + b_1\, Ln(Sum\ of\ PageRank) + b_2\, Ln(Sum\ of\ harmonic\ closeness) + b_3\, Ln(Weighted\ PageRank) + b_4\, Ln(Node\ resource) + b_5\, Ln(User\ intent\ depth) + b_6\, Ln(User\ intent\ bredth) + b_7\, Ln(Privacy\ Hazard)$$

Overfitting might exist when using all seven variables to explain the dependent variables. Therefore, we inspect the regression using different number of variables to discover the best regression model by determining R^2, R^2_{adj} and $\sigma^2_{residual}$ as shown in Table 8. When using more variables R^2 alwasys increases and overfitting might take place. The property of best regression model retains the highest R^2_{adj} and lowest$\sigma^2_{residual}$. The former means the regression explains appropriately and the latter indicates the regression predicts accurately. The best regression is using three variables that are Sum of harmonic closeness, user intent depth and breadth, and it contains the highest R^2_{adj} (58.1) and the lowest $\sigma^2_{residual}$ (0.93515).

				Web traffic		Run of network		User intent		
Number Of Variables	R^2	R^2_{adj}	$\sigma^2_{residual}$	Sum of PageRank	Sum of harmonic closeness	Weighted PageRank	Node Resource	Depth	Breath	Privacy Hazard
1	36.7	29.6	1.2121			X				
1	32.7	25.2	1.2496		X					
2	60.9	51.2	1.0099			X				X
2	59.6	49.5	1.0272	X						X
3	70.7	58.1	0.93515		X			X	X	
3	70.5	57.8	0.93828		X				X	X
4	72.3	53.9	0.98120	X	X			X	X	
4	72.3	53.8	0.98233	X	X				X	X
5	73.4	46.7	1.0548	X	X	X		X	X	
5	72.9	45.9	1.0632	X	X	X			X	X
6	73.6	34.1	1.1733	X	X	X	X	X	X	
6	73.5	33.8	1.1756	X	X	X		X	X	X
7	73.8	12.7	1.3500	X	X	X	X	X	X	X

Table 8: Evaluation of regression model for revenue

Before interpreting the regression, now we check again the residual plot with three variables regression at log-log level. Figure 41 conveys the regression at log-log level is no longer only predict Google's revenue accurately which was the data point at the rightmost before. The regression also predict others accurately.

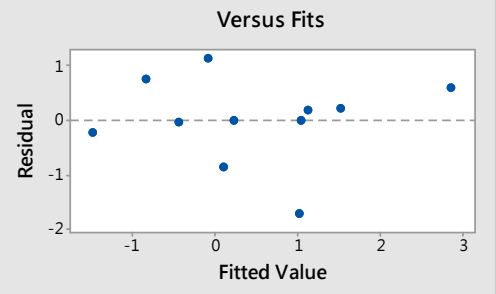

Figure 41: Residual plot for revenue regression at log-log level

Using those three variables, the regression becomes as the following formula. The value inside parenthesis is the value which reveals the coefficient is significant or not.

$$Ln(Revenue) = -14.64\ (0.0032) + 1.008\ (0.01)\ Ln(Sum\ of\ harmonic\ closeness) - 0.717(0.115)\ Ln(User\ intent\ depth) - 3.49(0.054)\ Ln(User\ intent\ bredth)$$

Ln (User intent depth) is not significant (0.115) if we set the threshold as 0.1 whereas the other two coefficients and the intercept are significant that they are less than 0.1. The regression conveys that the top third-party companies can increase 1.008 unit of revenue when increasing 1 unit of sum of harmonic closeness at log level. User intent doesn't help the company to increase their revenue due to the minus coefficient. Furthermore, we can infer that increasing web traffic helps increase revenue whereas increasing user intent doesn't help. Also, increasing run of network does not improve the revenue significantly.

5.3 Regression Analysis for Privacy Hazard

In contrast to revenue which are the benefit people support third-party web tracking, privacy hazard is the dark side of web tracking. Thus, we exploit regression to analyze privacy by other six graph statistics.

5.3.1 Regression

Due to skewness of data, we take the regression at log-log level as below. All the coefficients from b_1 to b_6 represents one unit increases or decreses the variblabe leads to b_i unit increasing or decreasing of dependent variables *Ln (Privacy hazard)*.

$$Ln(Privacy\ hazard) = c + b_1\ Ln(Sum\ of\ PageRank) + b_2\ Ln(Sum\ of\ harmonic\ closeness) + b_3\ Ln(Weighted\ PageRank) + b_4\ Ln(Node\ resource) + b_5\ Ln(User\ intent\ depth) + b_6\ Ln(User\ intent\ bredth)$$

Before examining the coefficients, we first look how the regression predicts the real value. Figure 42 conveys that when fitted value increases, the residual decreases and when fitted value

decreases, the residual increases. This leads to the significance of coefficient biased which is called heteroscedasticity.

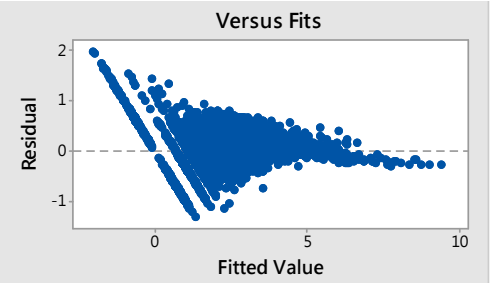

Figure 42: Residual plot for privacy regression

The formula is shown as below. It indicates one factor X affects the residual that we don't recognize. $\sigma^2_{residual}$ is constant but there is a factor X that causes the variable of residual. When X increases or decreases, variance of residual increases or decreases.

$$Variance(Residual) = X\ \sigma^2_{residual}$$

A solution to this problem is called Generalized Least Squares (GLS) which makes the variance of residual constant as the formula shown below.

$$Variance(Residual) = \sigma^2_{residual}$$

5.3.2 Generalized Least Squares

If we know which factor affects the residual, we can conduct weighted least squares to eliminate the factor x in variance of residual. For example, assume factor x is *sum of PageRank*.

$$Variance(Residual) = Ln(Sum\ of\ PageRank) \times \sigma^2_{residual}$$

We divide each variable by the variable *sum of PageRank* as the formula below.

$$\frac{Ln(Privacy\ hazard)}{Ln(Sum\ of\ PageRank)} = c + b_2 \frac{Ln(Sum\ of\ harmonic\ closeness)}{Ln(Sum\ of\ PageRank)} + b_3 \frac{Ln(Weighted\ PageRank)}{Ln(Sum\ of\ PageRank)} + b_4 \frac{Ln(Node\ resource)}{Ln(Sum\ of\ PageRank)} + b_5 \frac{Ln(User\ intent\ depth)}{Ln(Sum\ of\ PageRank)} + b_6 \frac{Ln(User\ intent\ bredth)}{Ln(Sum\ of\ PageRank)}$$

Then, we can obtain the constant variance of residual as the formula below. The computation of variance of residual is derived from the above formula. This method is called weighted least squares.

$$\frac{Variance(Residual)}{Ln(Sum\ of\ PageRank)} = \sigma^2_{residual}$$

The problem is we don't know which variable is the factor X. GLS discovers the factor X. The idea basically uses regression to estimate factor X.

Let the residual be u and the regression becomes as the formula below. The exponential function arises from log-log level.

$$Var(u) = \sigma^2_{residual} exp\ (c + b_1\ Ln(Sum\ of\ PageRank) + b_2\ Ln(Sum\ of\ harmonic\ closeness) + b_3\ Ln(Weighted\ PageRank) + b_4\ Ln(Node\ resource) + b_5\ Ln(User\ intent\ depth) + b_6\ Ln(User\ intent\ bredth))$$

Thus, we conduct regression for residual. We take logarithm to handle skewness and square function to not let the negative value and positive value eliminate each other. Conducing a regression for the residual applies the fact that the variance of residual is affected by a factor.

$$Ln(u^2) = c + b_1\ Ln(Sum\ of\ PageRank) + b_2\ Ln(Sum\ of\ harmonic\ closeness) + b_3\ Ln(Weighted\ PageRank) + b_4\ Ln(Node\ resource) + b_5\ Ln(User\ intent\ depth) + b_6\ Ln(User\ intent\ bredth)$$

Then, the above regression generates the fitted value g and we transform it back to its original form by exp(g) as h which is absolutely and factor x that affects the variance of residual. After that we apply the idea of weighted least squares that dividing each variable by h as the formula below.

$$\frac{Ln(Privacy\ hazard)}{h} = c + b_1\ \frac{Ln(Sum\ of\ PageRank)}{h} + b_2 \frac{Ln(Sum\ of\ harmonic\ closeness)}{h} + b_3 \frac{Ln(Weighted\ PageRank)}{h} + b_4\ \frac{Ln(Node\ resource)}{h} + b_5\ \frac{Ln(User\ intent\ depth)}{h} + b_6\ \frac{Ln(User\ intent\ bredth)}{h}$$

We check heteroscedasticity by applying White Heteroscedasticity F test. The null hypothesis is the regression has no heteroscedasticity. In GLS the White Heteroscedasticity F test is non-reject H0. In Ordinary Least Squares (OLS) White Heteroscedasticity F test rejects H0. This indicates GLS solves heteroscedasticity problem and the significance of coefficients now are available to test.

5.3.3 Regression Models Comparison

We conduct GLS and OLS and also want to explore the best regression model for GLS. Table 9 shows four regression models including OLS, GLS1, GLS2 and GLS3. The value inside parenthesis is p-value. All coefficients in OLS are significant. The power of explanation is also high and variance of residual is small. However, it contains heteroscedasticity and all the coefficients are biased. GLS1 is the first GLS model using all graph statistics as variables.

The problem of GLS1 is that the intercept is not significant, and it makes our regression conveying strong assumption that the mean of response variable is 0. We cannot eliminate intercept since it's the same that saying the mean of response variable is 0. Thus, we remove the insignificant variable weighted PageRank and derive GLS2. Again, the intercept is still not significant and we remove the insignificant variable sum of harmonic closeness and obtain GLS3. In GL3 all coefficients are significant and the evaluation criteria are similar to GLS1 and GLS2.

The coefficients provide the way to explain the dependent variable privacy hazard. User intent breadth doesn't increase privacy hazard (-0.879). This is because in our data many third-party companies can know all user intent by 14 categories. Web traffic doesn't increase privacy hazard (-0.056). The most important variable for privacy hazard is user intent depth and it increases privacy hazard (1.367), and run of network increases privacy hazard (0.1611).

Dependent variables : Privacy hazard, log-log level, #sample = 6082					
Independent variables		OLS	GLS1	GLS2	GLS3
Web traffic	Sum of PageRank	-0.1487 (0)	-0.0553 (0)	-0.0556 (0)	-0.056 (0)
	Sum of harmonic closeness	0.0242 (0)	0.0088 (0.0570)	0.0098 (0.0217)	
Run of network	Weighted PageRank	-0.0885 (0)	0.0109 (0.5756)		
	Node resource	0.1837 (0)	0.1418 (0)	0.1438 (0)	0.1611 (0)
User intent	Depth	-0.216 (0)	-0.8135 (0)	-0.8143 (0)	-0.8179 (0)
	Breadth	1.1551 (0)	1.3527 (0)	1.3534 (0)	1.3607 (0)
Intercept		-2.3323 (0)	0.0351 (0.8939)	-0.1018 (0.2977)	0.1057 (0.0047)
R^2		0.9281	0.8548	0.8549	0.8547
R^2_{adj}		0.9281	0.8547	0.8547	0.8546
$\sigma_{residual}$		0.3687	0.4384	0.4384	0.4385

Table 9: Regression for privacy hazard

5.4 Summary

We list our important and interesting findings in this section. All the distribution of computed graph statistics is skew including the graph statistics representing web traffic (sum of PageRank and sum of harmonic closeness), the graph statistics speak for run of network (node resource, weighted PageRank) and the graph statistics for user intent (user intent depth, user intent breadth and privacy hazard). User intent in breadth is not as skew as other statistics because most top third parties can know all categories of user intent in our data. Google dominates the revenue in web-tracking industry due to excellent performance in web traffic, run of network and user intent by investigating its comparatively huge sum of PageRank, sum of harmonic closeness, node resource, weighted PageRank, user intent depth and. Google also has highest privacy hazard.

The correlation of the computed graph statistics is significant correlated by hypothesis test that all the p-value is 0 as shown in Table 6: Correlation matrix of all graph statistics at log scale. This indicates those statistics are not independent. The correlation between the real revenue and our graph statistics is shown in Table 7: Correlation between digital ad revenue and graph statistics at log scale. Revenue is correlated to web traffic and run of network. More specifically, they are significant correlated (p-value < 0.1) except for the statistics of user intent related statistics.

The regression analysis reveals that the best regression model to predict the revenue is to use sum of harmonic closeness, user intent depth and user intent breadth as the following shown. We find the best model with the best explaining and predicting power.

$$Ln(Revenue) = -14.64\ (0.0032) + 1.008\ (0.01)\ Ln(Sum\ of\ harmonic\ closeness) - 0.717(0.115)\ Ln(User\ intent\ depth) - 3.49(0.054)\ Ln(User\ intent\ bredth)$$

This indicates that increasing or decreasing 1.008 unit of sum of harmonic closeness will increase or decrease one unit of revenue at log scale. For top companies to raise their revenue, they should focus on sum of harmonic closeness, generally speaking, increasing web traffic.

The best regression model for privacy is as follows where *h* is the weighted factor to handle heteroscedastic. The regression model indicates that increasing or decreasing 0.1611 unit of node resource will increase or decrease one unit of privacy hazard at log scale. For privacy concern, they should focus on decreasing user intent depth.

$$\frac{Ln(Privacy\ hazard)}{h} = 0.1057\ (0.0047) + -0.056\ (0)\ \frac{Ln(Sum\ of\ harmonic\ closeness)}{h} + 0.1611\ (0)\frac{Ln(Node\ resource)}{h} - 0.8179(0)\ \frac{Ln(User\ intent\ depth)}{h} + 1.3607\ \frac{Ln(User\ intent\ bredth)}{h}$$

6. DISCUSSION

This section describes the contribution of this thesis, and discuss the future work.

6.1 Contributions

This thesis successfully adopts computer science methods to discover insight in real world business. We use massive open data and open big data framework to provide the economic understanding of third-party web tracking. This thesis is the most comprehensive research examining web tracking with open data from the company level instead of domain level.

There are technical and business contributions in our work. In terms of technical contributions, this thesis contributes to the usage of Common Crawl and Apache Flink. We use open data and open source to discover interesting and important insights, and mining massive web crawl data with parallel programming. Efficiently accessing WHOIS information aids us to investigate the web crawl data in terms of company rather than domain.

The approach this thesis invent to compute graph statistic demonstrates the performance computation for bipartite graph. Most researches, algorithms and approach are designed for one-mode graph containing only one kind of node. We invent significant one-mode projection with resource allocation to transform the bipartite graph into one-mode graph. This approach keeps the original information structure in bipartite graph and processes it efficiently by pruning less significant edges.

We look at centrality measures from different perspectives endowed with business meanings. Therefore, we process the bipartite graph with several heterogeneous data including WebDataCommons, WHOIS, Alexa, and Google AdWords.

Regarding business contributions, before our work those understandings about third-party revenue are not publicly accessible. We reveal the main factors affecting the revenue and display web traffic, run of network and user intent for third party. The distributions of those statistics are extremely skew and Google dominates those three factors. Besides, we evaluate the relationship between those revenue factors with the real revenue and privacy hazard. This unveils which

revenue factors influence the privacy and the real revenue significantly higher and which influence them significantly lower. The knowledge provides a guideline for company who wants to raise revenue or ease privacy concerns. We apply graph algorithms with concepts of online business. This allows us to bridge the concept of business and computer science

6.2 Future Work

For future improvement, there are four aspects in this thesis. First, we aggregate the data to company level, and the company information of domains from WHOIS is not clean, consistent and easy to access although it's the current best way recognizing domain's company. Some companies sell massive processed domain's information and it's reasonable that their data quality is higher than directly accessing WHOIS. Besides, we hope WHOIS change their term of use and allows massively automatically accessing for popular domains at least.

Second, the other improvement is the time issue because our data is in 2012. The revenue and other information for each web tracking company of 2012 is mostly unavailable for free to access. Most companies provides the information for recent weeks, and asks for paying when accessing historical data. Also, the category information from Alexa and keyword price from Google AdWords are in 2015 but not 2012. Although price information is highly autoregressive, consistently using the information in 2012 is definitely more accurate for analysis. The category information we extract from Alexa for free to access is the top 525 domains for each category. A better study would be using all domains and including subcategories.

Third, in our regression analysis logarithm transformation has been used for ensuring the same unit of measurement. This is s a common technique for human or society related data such as wage, inflation and stock price. However, our graph statistics are not that kind of human or society related data and some statistics are extremely high (*sum of harmonic closeness*) and some are remarkably low (*sum of PageRank*). A better solution would be using Box-Cox transformation[42] for each variables to eliminate this effect of unit of measurement. Box-Cox represents a best practice where normalizing data or equalizing variance is desired proof by statistician.

Fourth, we have computed seven graph statistics to represent three main factors influencing revenue. Of course there are more factors affecting revenue. It's possible to compute more statistics, and analyze them with regression to recognize which factors are significant.

REFERENCES

[1] (2015). *Common Crawl*. Available: http://commoncrawl.org/

[2] D. Easley and J. Kleinberg, *Networks, crowds, and markets: Reasoning about a highly connected world*: Cambridge University Press, 2010.

[3] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata*, et al.*, "Graph structure in the web," *Computer networks,* vol. 33, pp. 309-320, 2000.

[4] C. Olston and M. Najork, "Web crawling," *Foundations and Trends in Information Retrieval,* vol. 4, pp. 175-246, 2010.

[5] A. Rana. (2010, CommonCrawl - Building an open Web-Scale crawl using Hadoop.

[6] R. Meusel, S. Vigna, O. Lehmberg, and C. Bizer, "Graph structure in the web---revisited: a trick of the heavy tail," in *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, 2014, pp. 427-432.

[7] O. Lehmberg, R. Meusel, and C. Bizer, "Graph structure in the web: aggregated by pay-level domain," in *Proceedings of the 2014 ACM conference on Web science*, 2014, pp. 119-128.

[8] S. Spiegler, "Statistcs of the common crawl corpus 2012," Technical report, SwiftKey2013.

[9] C. Hornbaker and S. Merity. (2013). *Measuring the impact of Google Analytics - Efficiently tackling Common Crawl using MapReduce & Amazon EC2*. Available: http://smerity.com/cs205_ga/

[10] A. Hunter, M. Jacobsen, R. Talens, and T. Winders, "When money moves to digital, where should it go," *Identifying the right media-placement strategies for digital display. Comscore White Paper,* 2010.

[11] G. Kovacs. (2012). *Tracking our online trackers*. Available: https://www.ted.com/talks/gary_kovacs_tracking_the_trackers

[12] P. Gill, V. Erramilli, A. Chaintreau, B. Krishnamurthy, K. Papagiannaki, and P. Rodriguez, "Follow the money: understanding economics of online aggregation and advertising," in *Proceedings of the 2013 conference on Internet measurement conference*, 2013, pp. 141-148.

[13] S. Dudoladov, A. Katsifodimos, C. Xu, S. Ewen, V. Markl, S. Schelter*, et al.*, "Optimistic Recovery for Iterative Dataflows in Action," 2015.

[14] J. R. Mayer and J. C. Mitchell, "Third-party web tracking: Policy and technology," in *Security and Privacy (SP), 2012 IEEE Symposium on*, 2012, pp. 413-427.

[15] B. Krishnamurthy, K. Naryshkin, and C. Wills, "Privacy leakage vs. protection measures: the growing disconnect," in *Proceedings of the Web*, 2011, pp. 1-10.

[16] B. Krishnamurthy and C. Wills, "Privacy diffusion on the web: a longitudinal perspective," in *Proceedings of the 18th international conference on World wide web*, 2009, pp. 541-550.

[17] P. Eckersley, "How unique is your web browser?," in *Privacy Enhancing Technologies*, 2010, pp. 1-18.

[18] F. Roesner, T. Kohno, and D. Wetherall, "Detecting and defending against third-party tracking on the web," presented at the Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation, San Jose, CA, 2012.

[19] B. Krishnamurthy and C. E. Wills, "Generating a privacy footprint on the internet," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, 2006, pp. 65-70.
[20] P. B. S. Vigna, "Axioms for Centrality," 2013.
[21] P. Boldi and S. Vigna, "In-core computation of geometric centralities with hyperball: A hundred billion nodes and beyond," in *Data Mining Workshops (ICDMW), 2013 IEEE 13th International Conference on*, 2013, pp. 621-628.
[22] U. Kang, S. Papadimitriou, J. Sun, T. Watson, and H. Tong, "Centralities in large networks: Algorithms and observations," 2011.
[23] M. Newman, *Networks: An Introduction*: Oxford University Press, 2010.
[24] F. Hüske. (2015). *Peeking into Apache Flink's Engine Room*. Available: https://flink.apache.org/news/2015/03/13/peeking-into-Apache-Flinks-Engine-Room.html
[25] B. Elser and A. Montresor, "An evaluation study of bigdata frameworks for graph processing," in *Big Data, 2013 IEEE International Conference on*, 2013, pp. 60-67.
[26] ICANN. (2015). *WHOIS*. Available: http://whois.icann.org/en
[27] I. J. Block, "Hidden Whois and Infringing Domain Names: Making the Case for Registrar Liability," *U. Chi. Legal F.,* p. 431, 2008.
[28] J. Woodridge, "Introductory Econometrics. 3rd," *Mason, Ohio: Thomson Higher Education,* p. 199, 2006.
[29] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*: Morgan Kaufmann, 2005.
[30] K. Benoit, "Linear regression models with logarithmic transformations," *London School of Economics, London,* 2011.
[31] U. o. Mannheim. (2015). *Webdatacommons*. Available: http://webdatacommons.org/
[32] Alexa. (2015). *The top ranked sites in each category*. Available: http://www.alexa.com/topsites/category
[33] Google. (2015). *Google Adwords*. Available: https://www.google.com/adwords/
[34] I. L. Wiki. (2015). *Sensitive Data*. Available: http://itlaw.wikia.com/wiki/Sensitive_data#cite_note-1
[35] T. Zhou, J. Ren, M. Medo, and Y.-C. Zhang, "Bipartite network projection and personal recommendation," *Physical Review E,* vol. 76, p. 046115, 2007.
[36] K. A. Zweig and M. Kaufmann, "A systematic approach to the one-mode projection of bipartite graphs," *Social Network Analysis and Mining,* vol. 1, pp. 187-218, 2011.
[37] W. Xing and A. Ghorbani, "Weighted pagerank algorithm," in *Communication Networks and Services Research, 2004. Proceedings. Second Annual Conference on*, 2004, pp. 305-314.
[38] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive datasets*: Cambridge University Press, 2014.
[39] S. Ewen. (2015). *Hash join failing*. Available: http://apache-flink-user-mailing-list-archive.2336050.n4.nabble.com/Hash-join-failing-td1389.html
[40] (2015). *Builtwith*. Available: http://builtwith.com/
[41] eMarketer, "Company reports," eMarketer, Ed., ed, 2014.
[42] J. W. Osborne, "Improving your data transformations: Applying the Box-Cox transformation," *Practical Assessment, Research & Evaluation,* vol. 15, pp. 1-9, 2010