DATA ANALYTICS OF LIBRARY RESOURCES

by

Debashish Roy

M.Sc. in Computer Science, Uppsala University, Sweden, 2007

A thesis

presented to Ryerson University

in partial fulfillment of the
requirements for the degree of

Master of Science

in the Program of

Computer Science

Toronto, Ontario, Canada, 2015

©Debashish Roy 2015

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my dissertation may be made electronically available to the public.

Data Analytics of Library Resources

Debashish Roy

Master of Science, Computer Science, 2015

Ryerson University

Abstract

Library Analytics is used to analyze the huge amount of data that is collected by most colleges and universities when the library website is browsed. It is really useful to assess and evaluate the usage of different types of library e-resource items such as books, journals etc. In this thesis work we have designed and implemented a multi-threaded Library Analytics system specifically for the Ryerson Library. We have analyzed the library usage data to generate different types of reports through a data visualization engine based on requirements from the librarian. Furthermore, we have paid special attention to the task of e-resource item clustering, which could better organize the resources and potentially be used to recommend interesting items to users. We have compared different clustering algorithms and found that association-rule based clustering is 20% more accurate than others and it also identifies the hidden relationships between different browsed articles.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Dr. Cherie Ding for her valuable support and guidance in helping me to go through all the difficulties in my work. Her suggestions have greatly enhanced my knowledge and skills in research and have significantly contributed to the completion of this thesis.

In addition, I would like to thank Dr. Ali Miri, Dr. Alireza Sadeghian, and Dr. Isaac Woungang who have reviewed my thesis and have given me valuable comments to improve my thesis. Furthermore I want to thank Lei Jin and Dana Thomas of Ryerson Library for their continuous support.

Also, I would like to acknowledge the support of the Computer Science Department of Ryerson University.

Finally, I would like to express my deep appreciations to my family, relatives, and friends who have motivated and supported me during these years of study.

Contents

<i>Declaration</i>	ii
<i>Abstract</i>	iii
<i>Acknowledgements</i>	iv
<i>List of Figures</i>	vii
List of Acronyms	viii
1 Introduction	1
1.1 Background	1
1.2 Problem Statement	3
1.3 Objectives	5
1.4 The proposed approach	6
1.5 Thesis Outline	7
2 Related Works	8
2.1 Introduction	8
2.2 Clustering Algorithms	10
2.2.1 Content-based clustering	10
2.2.1.1 Hierarchical Clustering	10
2.2.1.2 Partitional Clustering	12
2.2.1.3 Expectation-Maximization Clustering	14
2.2.1.4 Density Based Clustering	15
2.2.2 Clustering on Association Rule Mining	16
2.3 Library Analytics	18
3 Library Analytics	21
3.1 Introduction	21

3.2	Web Log Analyzer	22
3.2.1	Pre-processing of the Log	25
3.2.2	URL Categorization	27
3.2.3	Log Analyzer and Visualization	28
3.3	Clustering of e-resource items	33
3.3.1	Content Based Clustering	33
3.3.1.1	Introduction	33
3.3.1.2	K-means Clustering	34
3.3.1.3	Farthest First Clustering	34
3.3.1.4	Filtered Clustering	34
3.3.1.5	Density Based Clustering	35
3.3.1.6	Hierarchical Clustering	36
3.3.1.7	Expectation and Maximization Clustering	36
3.3.2	Association Rule Based Clustering	37
3.3.2.1	Introduction	37
3.3.2.2	Generating Association Rule	38
3.3.2.3	Hypergraph Generation	39
3.3.2.4	Partitioning the Hypergraph	40
4	Experiments	42
4.1	Prototype Implementation	43
4.2	Experiment on Clustering Algorithms	48
4.3	Results Analysis for Clustering Algorithms	55
5	Conclusion and Future Work	59
5.1	Conclusion	59
5.2	Future Work	60
	Bibliography	61

List of Figures

2.1	Hierarchical Clustering	11
2.2	Density Based Clustering	16
3.1	A Sample Log File	24
3.2	Library Log Analyzer System Design	30
3.3	Reports generated by the Data Visualization Tool	31
4.1	GUI for Log file preprocessing	44
4.2	GUI for Log Visualization	45
4.3	Database Accessed by Unique Users	46
4.4	Download Chart for a Single Database	46
4.5	Server Usage by users based on Hits	47
4.6	Top 10 Departments based on a single database usage	47
4.7	Manual Clustering	48
4.8	HMETIS Clustering	49
4.9	K-means Clustering	49
4.10	FarthestFirst Clustering	50
4.11	Filtered Clustering	50
4.12	MakeDensityBased Clustering	51
4.13	Expectation Maximization Clustering	51
4.14	Hierarchical Clustering	52
4.15	Cluster Evaluation	56
4.16	Running time of algorithms	57

List of Acronyms

ARM Association Rule Mining

EM Expectation and Maximization

FAM Federated Access Management

FTC Frequent Term Based Clustering

HFTC Hierarchical Frequent Term Based Clustering

IDF Inverse Document Frequency

LAMP Library and Metrics Project

LCA Latent Class Analysis

TF Term Frequency

URL Uniform Resource Locator

Chapter 1

Introduction

1.1 Background

In the current world data is generated from various sources. Most of the organizations have growing interest to understand these data properly because a better understanding will help the organizations to come up with some attractive and profitable ideas. To come up with those ideas different organizations are developing or applying different data analytics techniques. Analytics is defined in [13] as the process of developing actionable insights through problem definition and the application of statistical models and analysis against existing and/or simulated data. By saying actionable it means analytics deals with different types of practical solutions rather than theoretical descriptions. To develop a successful approach towards analytics different analytics tools and methods are developed and they are used to understand the nature of data. These tools generate different types

1.1. BACKGROUND

of valuable patterns and these patterns help organizations to make different well informed decisions.

Different types of analytics can be applied to hardware or network troubleshooting, to optimize performance of hardware or network tools, to increase usability of various websites etc. When the analytics software generate different valuable patterns from the commercial websites by mining web usage data then this type of analytics approach is known as Web Analytics [22]. Different types of Web Analytics tools are used to improve websites profitability. In [51] Web Analytics is defined as “the act of increasing a website’s persuasion and relevancy to achieve higher conversion rates which will provide a way for website optimization”. So with the help of big data and advanced analytical technologies organizations can make their business profitable. Usage of Web Analytics by different companies is increasing day by day. In [28] it shows that the number of organizations using analytics to improve their business is increasing every year. To apply Web Analytics most of the websites store users’ browsing information in the web server logs. Then they use different Web Analytic tools to better understand user browsing patterns. Later these patterns are analyzed to increase the usability of the websites.

A huge amount of data is collected by most colleges and universities. A part of this huge data is the library usage data which is collected when the library website is browsed. To analyze the data properly different analytics tools are applied on the library usage data. So library analytics is to apply analytics on the library usage data of any college or

university. The analysis is useful to assess and evaluate different types of library e-resource items and you can also monitor activity of library users. The analytic tools can apply different data mining techniques and prepare attractive visualizations to reveal hidden patterns and trends. In old days [7] usage data was gathered by hand. Counting was done each time a print journal volume or issue was reshelfed. Citation studies were done by doing a random sampling of the faculty articles. But today we have more quantitative data. Publishers or vendors generate mostly standardized usage statistics for databases, books and journals. Now we have statistics for owned, subscribed, and unowned content. Along with everything online citation indexes are also available. Analyzed results of this huge data help to evaluate the usage at the libraries. By evaluating the usage statistics one can find out the least browsed document sets, the department that uses the library the most, spamming websites that create unusual traffic, how people read books online [9] etc. So proper analysis will help the librarians to find out the weak points in the system and hence it helps the library to increase the quality of library service while reducing the manual labor costs and hazards.

1.2 Problem Statement

Library Analytics has been an active research area in recent years. Various analytic decisions have been made through Library Analytics and different types of problems are solved. Analytics work in [10] demonstrates that pattern of library usage by different

1.2. PROBLEM STATEMENT

undergraduate students from various disciplines can be determined. Then the library can identify how the students are using the library. In [53] it has been shown that Library Analytics is also able to correlate student academic performance with library material usage. Research on Library Analytics can now answer to the question -“Who is not studying in the university?”. The research work in [8] compares under-graduate academic disciplines and library use and they are able to group the students that do not study.

The researchers have also used Library Analytics to predict the requirements of future library user. [52] demonstrates an approach which can be used to predict the future usage. Then based on the predicted usage they can redesign different library services which will make the services more usable. Library Analytics tools can use web site usage statistics to provide attractive site development ideas [4] and it can also be used to discover usage patterns from web data [44].

To analyze library data standard web analytic tools are not enough because URL pattern for library data is unique. The library has subscription to various journals, e-books etc. so when any resource is accessed the pattern of the URL is also changed based on the vendors. There are also many research projects on library analytics, but none of them has become a standard and many of them are still on-going. Researchers from University of Pennsylvania are working on Metridoc [16] which is a data integration framework to assist libraries with data ingestion and normalization. University of Illinois at Chicago are working with their library analytic tool Counter4 [16]. Cardiff University

has developed RAPTOR [27] which is a software toolkit for reporting e-resource usage statistics in a user-friendly manner. All these projects are working on to provide e-resource usage statistics but they do not address the problem of identifying similarity among different e-resource items such as books, journals, articles etc. So we cannot use them in our project directly.

Library has a huge collection of books, journals etc. Sometimes identifying useful or interesting information could be hard. If we could find relationship between books or journals or papers, and put similar ones in a group, then it is easier for users to browse the whole collection, and it is also possible to recommend new books/journals/articles from the same group to users which might be of users' interests as well. In this research we would like to perform general analytic tasks identified by librarians and present visualized results to them. Specifically, we also want to study the problem of clustering of books/journals/articles in the context of library analytics, by investigating different clustering approaches, we want to find an approach which can identify similarity, not only based on their contents but also based on users' browsing behavior.

1.3 Objectives

The objectives of this thesis work are: 1) Design and Implement a Library Analytics System which is a multi-threaded Web Log Analyzer specifically for Ryerson Library; 2) Generate different types of reports for the Library Analytics System using a visualization

tool; 3) As there are many ways to cluster documents we would like to compare their performances, we mainly focus on content-based approach and association rule based approach. The former might be more common whereas the latter might be able to find hidden results, especially the relation between content-wise non-similar documents.

1.4 The proposed approach

A library has thousands of articles to browse. With the use of web usage mining we can easily get the top browsed e-resource items. There are many different ways available to cluster these items. The first way of clustering is based on the content of the e-resource items. Every item is represented as a vector of term weights. The most popular weighting scheme is TF-IDF. Since we do not have access to full text, only title is available and there are usually not many repeating terms in the title, we only use IDF weight. Once the IDF calculation is done we will have a big document-term matrix. Now to cluster the titles we can apply different clustering algorithms on the matrix. We will apply a few commonly used clustering algorithms: Hierarchical Clustering, Centroid Based Clustering, EM Clustering, Density Based Clustering etc.

The second approach of clustering is based on access patterns, if two books are always visited together they are similar even if their contents might be different. Here clustering is done based on users' browsing patterns which will provide frequent itemsets. On the top browsed e-resource items association rule mining algorithm is applied to get the frequent

itemsets. Then we apply a partitioning algorithm HMETIS to cluster the items.

1.5 Thesis Outline

The rest of the thesis is organized as follows:

Chapter 2 reviews and analyzes the existing research work in the field of Library Analytics. It explains different types of clustering algorithms like Hierarchical Clustering, Centroid Based Clustering, EM Clustering and Density Based Clustering and describes how clustering is done based on Association Rules.

Chapter 3 provides the details of our proposed approach. To analyze the library web log we have developed a Web Log Analyzer. To design a Web Log Analyzer we had to follow steps such as: Log pre-processing, URL Categorization and development of a tool to visualize the results. Here all these steps are described. In our approach we selected Association Rule Based clustering algorithm based on HMETIS, a Hypergraph Partitioning Algorithm. We have applied the HMETIS partitioning algorithm to figure out the best cluster. It also explains different content-based clustering algorithms.

Chapter 4 explains our experiment design and discusses the results of different clustering algorithms with the implementation details of Library Analytic Prototype.

Finally, in Chapter 5, we conclude our thesis with a summary of results and analysis. Our future research directions are also discussed in this chapter.

Chapter 2

Related Works

2.1 Introduction

Everyday the digital world that we are living in generates enormous amount of data. Data are generated from many sources like transactions related to item purchase, web server logs, satellite imagery etc. Using these data one can describe the characteristics of a living species, identify different attributes of natural phenomenon [56] etc. Most educational institutes like colleges and universities collect and store vast amounts of data. These data are related to students' admission records, library records etc. To analyze these data different analytics applications are used. These applications analyze the data and come up with different reports and data visualization to reveal different patterns, trends and exceptions [22]. To perform analytics different data mining algorithms are used such as: Clustering, Classification, Association Rule Mining etc. The benefit of

using these algorithms in a university library is you can predict upcoming challenges both for the university and for the students. You can also get an understanding of how students are using the library.

When any website is browsed that browsing information is stored in a web server log file. This file is created automatically and it is maintained by the server. It stores all the activities that the server has performed to complete different types of requests. From the log files we can generate browsing history. Analytics are applied on these files to find different browsing patterns and to figure out trends in the data. When analytics is done on a server log of a university library then from the browsing pattern we can create clusters of similar documents. When we cluster the similar documents then a large document collections can be browsed very quickly, an efficient recommendation system can be designed and in an university environment we can also figure out which user groups are using the clusters.

To cluster the documents a various types of clustering approaches have been proposed. The researchers have used clustering algorithms like: Hierarchical Clustering, Partitional Clustering, EM Clustering, Density Based Clustering etc. to cluster the documents. Again clustering can also be done based on the frequent itemsets. Here association rule mining is applied to generate a set of frequently browsed documents. Then based on the generated association rules clusters are formed. In this chapter, we will first review various clustering algorithms that are proposed in the past researches for web usage mining, text document

mining etc. Then, we will review some of the literatures that are more closely related to our work.

2.2 Clustering Algorithms

Clustering is a data mining approach which is used to classify or group data into a set of categories or clusters. Clusters are formed by analyzing various features of data items. Items that are kept in the same cluster should have some common properties based on different types of criteria. Clustering is also known as unsupervised classification because this process does not use a classifier or any labeled data to create clusters. The unknown nature of data is explored by clustering and to perform clustering you need little or no prior information. There are different types of clustering algorithms available. All these algorithms partition data objects into a certain number of clusters [56]. In the following subsections we will focus on different document clustering methods which are proposed and used by different researchers and are used in our experiment.

2.2.1 Content-based clustering

2.2.1.1 Hierarchical Clustering

Generally there are two types of clustering, i.e. partitional clustering and hierarchical clustering [21]. In hierarchical clustering data is grouped with a sequence of nested partitions. There are two types of hierarchical clustering algorithms: agglomerative and

Figure 2.1: Hierarchical Clustering [57]

divisive. Many different approaches to hierarchical clustering from divisive to agglomerative clustering have been suggested in [48][39] to categorize the documents. A binary tree or a dendrogram is used to represent the results of hierarchical clustering. In Fig. 2.1 we see the direction for agglomerative clustering and divisive clustering is opposite to each other. Agglomerative clustering is a bottom-up approach. It starts with all points in their own group and then at each step of the algorithm, the pair of clusters with the shortest distance are combined into a single cluster and the algorithm stops when all the points are combined into a single cluster. The divisive clustering starts with all points in a single cluster, then at each step of the algorithm clusters are partitioned into two sibling clusters having maximum dissimilarity, the algorithm stops when points are partitioned into n -clusters.

Efficient document clustering algorithms analyze large amounts of information and form a small number of meaningful clusters. Using these clusters navigation and browsing mechanisms can be improved. Any ideal clustering algorithm provides meaningful hierarchies of large document collections. In [60][59] the authors focus on document clustering algorithms that build such meaningful hierarchical solutions. In a comprehensive study they have shown that performance of partitional clustering algorithms are always better and they provide better solutions than agglomerative algorithms. In their research they have designed a new class of clustering algorithm known as constraint agglomerative algorithm. Here they have combined the features from both partitional and agglomerative clustering algorithms.

2.2.1.2 Partitional Clustering

As we have seen in hierarchical clustering clusters are determined through a successive levels of clusters through an iterative approach. Partitional clustering assigns a set of points into K clusters [58] and there is no hierarchical structure. Based on a criterion function the partitional clustering algorithms organize the datasets into K clusters. Different types of partitional clustering algorithms have been designed: i) K -means clustering algorithm which is based on the sum-of-squared-error criterion, ii) partitional clustering algorithm under a probabilistic framework which is based on a probability density function, iii) k -medoids clustering algorithms where k points from the original dataset are selected as the medoids and using these medoids as anchors k number of clusters are built

[1], and iv) Bisecting K-means clustering algorithm starts with a single cluster of all the documents, then they are splitted into two sub-clusters using the basic K-means algorithm [45]. The authors in [45] have compared different types of clustering algorithms. In their result they have found that the bisecting K-means clustering algorithm is better than the standard K-means approach and it is also better than the hierarchical clustering algorithms.

Document clustering algorithms perform an unsupervised classification of documents and similar types of documents are kept in the same group. For text document clustering the authors in [43] have applied K-means clustering algorithm on different representation of data such as tf, tf-idf and boolean. Along with various types of representations they have also used different feature selection schemes such as with or without stop word removal and with or without stemming. They have found that tf-idf representation and use of stemming provides better clustering. [24] analyzes Lloyd's algorithm based K-means clustering algorithm and describes a filtering algorithm. The algorithm is based on a stored Kd-tree where multidimensional datapoints are saved.

The authors in [5] propose a frequent term based text clustering approach. Frequent item sets are generated using the algorithms for association rule mining. The mutual overlap of frequent term sets are used to cluster the text documents. They present two clustering algorithms, one is Frequent Term-based Clustering (FTC) which is based on K-partitions and another is Hierarchical Frequent Term-based Clustering (HFTC) which

is based on hierarchical clustering.

In [54] clustering has been examined as an educational data mining method. The authors have compared K-means clustering and the model-based Latent Class Analysis (LCA). To compare they have used usage data from an educational digital library service. Their research result shows that LCA is better than K-means clustering algorithm.

The authors in [55] use K-means clustering algorithm to cluster web users based on web user log data. They work with a set of web users and their associated historical web usage data. Over here they study users' behavior characteristics and cluster them based on K-means clustering.

2.2.1.3 Expectation-Maximization Clustering

EM that is expectation maximization algorithm which enables parameter estimation in probabilistic models with incomplete dataset [17]. EM clustering algorithm is based on an interactive approach and it works on a probability distribution function that finds the parameters that have the maximum probability of its attributes. Every iteration has got one expectation step and one maximization step. The expectation step estimates the probability of every point that belongs to different cluster. Then in the maximization step estimation is done again on the parameters of the probability distribution of each cluster.

The authors in [3] compare expectation maximization clustering algorithm and K-means clustering algorithm on different types of dataset. Their result shows that K-

means performs comparably to EM. For low dimensional dataset EM performs better than K-means but for high dimensional dataset EM fails. In [38] the authors have used web usage data to compare CFuzzy means and EM clustering. Their result shows that EM clustering algorithm shows 5 to 8 percent better performance than CFuzzy means clustering algorithm.

2.2.1.4 Density Based Clustering

DBSCAN is a density based clustering algorithm. In DBSCAN high density regions are separated from each other by low density regions [49]. In a center-based approach of DBSCAN clustering density for a particular point is determined by counting the number of points within a specified radius. Fig. 2.2 shows the concept of center based approach of DBSCAN clustering. DBSCAN works with three types of points: core point, border point and noise point. Data objects that are located in the low-density regions are known as noise or outliers [30]. To cluster web usage data DBSCAN clustering is also used. The authors in [40] have used DBSCAN clustering algorithm to mine web usage data. Over here a comparison was done between DBSCAN clustering and agglomerative clustering. They found that DBSCAN clustering algorithm has better efficiency and superior performance. The authors in [31] use DBSCAN clustering algorithm to mine web usage data to discover visitor group with common behavior.

Figure 2.2: Center based approach for DBSCAN clustering [49]

2.2.2 Clustering on Association Rule Mining

In a database of transactions we can cluster the items that are present in the transactions. It is very effective if we can cluster related items [20]. For example every sale record in a grocery store is a transaction and every transaction consists of sold items so if clustering is applied on these transactions then we will have clusters of items sold together. Then this knowledge can be used to perform effective shelf-space organization which will help the consumers to find out the related items quickly. So to get the frequent itemsets we can apply Association Rule Mining (ARM) algorithms. Based on given support and confidence ARM will provide us frequent itemsets. Support and confidence are two user defined parameters. Minimum support and minimum confidence are set by the user to generate the most frequent itemsets. From the ARM rules hidden relation-

ships among the items can be figured out so if ARM based clustering is applied then we may see surprise items in the clusters. To cluster the frequent itemsets the authors in [20] considered every single association rule as an edge of an hypergraph. So with all the association rules a hypergraph was formed. Then they applied a graph partitioning algorithm, hMetis, to cluster the items.

The authors in [32] use ARM based clustering on Web Usage Data for pattern discovery. Web usage mining is a collection of three phases: pre-processing, pattern discovery and pattern analysis. In the pattern discovery step they have used association rule mining to find out the frequently browsed pages and then they applied DBSCAN algorithm on those association rules to find out the clusters. The authors in [19] applied hierarchical clustering to design an efficient decision support system. In [29] to cluster the dataset hierarchical clustering is used on the frequent itemsets that are generated by Apriori Association Rule.

The authors in [34] apply association rule based clustering on web images. From the association rules hypergraph is generated. Different features of the images are used as vertices and association rules are used as hyperedges. Then a hypergraph partitioning algorithm, hMetis, is used to generate clusters of features and using a scoring function they assign images to different clusters. Both the research work in [32] and [34] have used the same approach on different types of dataset. [32] uses market basket data and [34] uses web images as their datasets.

The authors in [33] use association rule based text document clustering. In their approach to cluster the association rules they have used K-means clustering algorithm. In [37] clustering is done based on association rule to find relevant links between binary rare attributes in a large data set. In [15] document clustering is done based on entity association rules. The authors in [36] investigate the performance of document clustering approach based on Association Rules Mining. In their research they have found that clustering based on association rule mining has higher efficiency, scalability and accuracy.

2.3 Library Analytics

Data analytics in the library is useful to understand the behavior of all the users of the library web site. Using library analytics we can answer questions about the behavior of website users like when the website is visited, the duration of the visit and how they come to the website. The author in [50] applies analytics on library server log to understand the behavior of its users. In [35] Google analytics is used on library web usage data to find out statistical information of different page usage, users movement in the website etc.

In [23] library analytics is applied to figure out the effect of library usage for successful students. It is found that students who do well in the exams they use library more than the students who do poorly in the exams. The author in [14] applies library analytics to discover the impact of library use in evaluating student performance. They found out

that there is a strong relationship between students grades and electronic resource usage. The more the students use the library the better are their grades. The authors in [46] apply library analytics on library usage data to explore the effect of undergraduate library usage based on demographic characteristics such as age, gender, ethnicity and country of origin. Their study has found that there is a strong relationship between demographic characteristics and library usage.

The authors in [11] have designed a Library Analytics and Metrics Project which is known as LAMP and the main target of their design is to facilitate the statistically significant correlation across a number of universities between library activity data and student attainment. The Jisc Library and Metrics Project (Jisc LAMP) [41] is working for the libraries to work with many types of data they receive everyday. When these data are analyzed properly libraries can get insight of the usage of their collections across the institutions. This will help them for the strategic development of the resource collections. So the libraries are considering to use data for informed decision making. From JiscLamp project it is found that there is a lack of tools and services for libraries to analyze these data properly. The authors in [42] explore the potential of a shared analytics service for UK academic libraries. To accomplish the task they used Jisc LAMP. Metridoc [16], which is a data integration framework to assist libraries, facilitate the collection, transport and the use of library activity data. It provides simple tools to extract useful information from various data sources and after different transformations data is stored in the repository.

2.3. LIBRARY ANALYTICS

The MetriDoc framework is currently undergoing test implementation at the University of Chicago and North Carolina State University. The library analytic tool Counter4 [16] provides reports on e-resources in one-single, integrated code of practice, including journals, databases, books, reference works and multimedia. Counter4 mainly works on content ranking based on number of downloads and citations. Now they are working on to share data for cross-institutional analysis. The RAPTOR [27] software toolkit integrates Federated Access Management (FAM) technology and proxy technologies as well as other existing resource usage reports. For FAM it has got Shibboleth Identity Provider Service. This identity technique is widely used in the academic context and it is also a big part of FAM paradigm. RAPTOR delivers statistics in a graphical and textual form. RAPTOR uses two distinct categories of raw information that is i) Authentication information and ii) Usage information. Authentication information can be derived from the log files of authentication systems. It can gather usage information by consuming existing usage reports produced by information providers or by analyzing the log files of proxy software such as EZproxy. Right now RAPTOR is an open source, free to use software suite. RAPTOR can be deployed by institutions that have a need for statistical usage information of their authentication systems. All these tools provide statistics on usage information but they do not identify the similarities among different e-resource items. To use them in our project further customization are required. So we cannot use these tools directly in our project.

Chapter 3

Library Analytics

3.1 Introduction

We have applied Library Analytics on the web server log of Ryerson University's Library. This server log is basically a web usage log. So as a first step we have to apply different steps of web log processing. To accomplish this task we have designed a prototype application to analyze the web log. This prototype has two parts: first part is used to pre-process the server log, to categorize different URLs and it also includes a tool to visualize the data. The second part applies association rule based clustering algorithm on the processed data. In this chapter we will describe all the steps involved in the prototype. In the Web Log Analyzer section we will describe how the prototype is designed, the Clustering section will focus on different steps involved to cluster data based on different clustering algorithms.

3.2 Web Log Analyzer

To extract the knowledge and to figure out different patterns from Web data different techniques of web mining are applied. The web mining approach that is applied on web usage data or on web log data is known as web usage mining. Here browsing data is analyzed and target of web usage mining is to discover interesting and frequent user access patterns. The browsing data or the usage data is stored in web server logs. When a user browses any website that user's behavior is stored in the server log. So proper mining will reveal the browsing pattern of that user.

When a user browses the library web site of Ryerson University, that browsing information is stored in the server log. There are different types of server logs available such as: access log file, error log file, agent log file and referrer log file. Access log files store all the requests that are processed by the server. We have worked mainly with the access log files of the library's website. In Ryerson the access data is collected through the EZproxy server. EZproxy is a middleware solution for remote user authentication. EZproxy server maintains a standard web server log file with usage information [47]. It is used by the libraries to give access from outside the library's regular network. EZproxy records usage information using common log format. EZproxy server for Ryerson's library uses the following log format:

```
%h %u %l %{ezproxy-session}i %t "%r" %s %b
```

Here %h is the host accessing the EZproxy server and it is always an IP address,

%u is username used to log into the EZproxy server, %l is the remote username but it is always a -, %i is EZproxy's identifier for the user's current session, %t captures the timestamp, %r gets the complete request with server information, %s collects the status codes returned by the server and %b provides a statistics of number of bytes transferred. Here is an example log entry created by EZproxy server:

```
"10.0.0.1 X2bFdM1R3txwlv - 13d8f72f08d1a4e1c418a7cb8fc31437 [01/Jun/2014: 00:47:10  
-0500] "GET http://site.ebrary.com:80/lib/oculryerson/docDetail.action?  
docID=10251051 HTTP/1.1" 200 29732"
```

The entry says user 'X2bFdM1R3txwlv' from IP address '10.0.0.1' having a session-id '13d8f72f08d1a4e1c418a7cb8fc31437' requested to read a document having id '10251051' on the ebrary server at the given time stamp and the status code 200 says, the requested resource was found on the server so 29732 bytes were transferred to the user. Fig. 3.1 shows a sample server log.

EZproxy creates a log file entry every time an e-resource is accessed through the EZproxy server. Log data is recorded automatically when a user clicks on a proxied link to one of the library's resources including the time of the request, the specific e-resource requested, and the requesting user's IP address. EZproxy logs store information about a user's entire visit to a webpage, logging every file downloaded and it also logs the search queries made against databases and indexes. Interpretation of this data cannot be standardized because each resource vendor structures their website differently. So URLs

3.2. WEB LOG ANALYZER

pointing to journal articles, e-books, and database queries will differ significantly from vendor to vendor. Because of this reason we cannot use the generic log analyzer in our project.

```
1 10.2.2.0 b8mBa6nu7yWmpG7 - 0675a41e13e70fe61ca75821752a66bc [16/Dec/2013:00:01:41 -0500] "GET
http://www.sciencedirect.com:80/science/article/pii/S0301421511010603?np=y HTTP/1.1" 200 143849
2 10.2.2.1 gNMAD4Mc4iyUCmG - 708f82ff2556f6933a7b90885af3da05 [16/Dec/2013:00:01:42 -0500] "GET
http://journals1.scholarsportal.info:80/tmp/2104330556959676391.pdf HTTP/1.1" 200 1601355
3 10.2.2.0 b8mBa6nu7yWmpG7 - 0675a41e13e70fe61ca75821752a66bc [16/Dec/2013:00:02:59 -0500] "GET
http://site.ebrary.com:80/lib/oculryerson/docDetail.action?docID=10172938 HTTP/1.1" 200 29026
4 10.2.2.4 uhFyuZQXMRKKGD2 - 5123d224ca13f993ec68d7329259d7cf [16/Dec/2013:00:07:37 -0500] "GET
http://journals2.scholarsportal.info:80/tmp/17176093651067359023.pdf HTTP/1.1" 200 46478
5 10.2.2.9 o4DkDAF1R2UmFTk - 96a5ff56c17baf983df201eb377cf303 [16/Dec/2013:00:08:10 -0500] "GET
http://www.swetswise.com:80/swoc-web/linkingDetails.html?openURL=false&issn=0092-2102&eissn=0092-2102
&volume=39&issue=4&page=379 HTTP/1.1" 200 323316
6 10.2.2.9 o4DkDAF1R2UmFTk - 96a5ff56c17baf983df201eb377cf303 [16/Dec/2013:00:09:42 -0500] "GET
http://site.ebrary.com:80/lib/oculryerson/docDetail.action?docID=10240304 HTTP/1.1" 200 29020
7 10.2.2.29 MVsasq4USS5GL3k - 11f06b0ab82e75b1831e7956e0268778 [16/Dec/2013:22:37:34 -0500] "GET
http://vibrationacoustics.asmedigitalcollection.asme.org:80/data/Journals/JVACEK/28877/604_1.pdf
HTTP/1.1" 200 110078
8 10.2.2.98 r27Y5TRLDo3pWL2 - c8c1474452f01833f31ef51116a5f96f [16/Dec/2013:22:37:48 -0500] "GET
http://journals2.scholarsportal.info:80/tmp/3129382860591231552.pdf HTTP/1.1" 200 1238067
9 10.2.2.99 RCwfWGEqtKHMjKc - 0d3e183cfcc021d138090ddc1de5412b [16/Dec/2013:22:39:57 -0500] "GET
http://journals1.scholarsportal.info:80/tmp/10001846560629628863.pdf HTTP/1.1" 200 391105
```

Figure 3.1: A Sample Log File

A single access event generates multiple lines of code in the log file. So the lines of code must be parsed to ensure that it is counted as only one transaction for that e-resource. So for proper parsing we need to apply pre-processing steps on the dataset. Next we are going to describe the different steps associated with pre-processing. In the pre-processing part we have implemented Java multi-threading to speed-up the procedure. For the pre-processing we could use a free/commercial log analyzer but then further customization is required to categorize the URLs. So we cannot use them directly in our work. Similar

type of work like RAPTOR is good for the institutions that have a need for statistical usage information of their authentication systems. So to use RAPTOR directly further customization is required.

3.2.1 Pre-processing of the Log

Data pre-processing is an important step in the data mining process. It transforms raw data into an reliable format. Once pre-processing is done then data can be used for knowledge discovery. Data pre-processing is done through a series of steps such as: Data Cleaning, User identification and User session identification.

When we browse a web page different images and scripts are downloaded along with the HTML file. To download each and every item the server is requested every time and the server works to serve the request. As each request is served an entry is also created in the log file. So for every visit multiple lines of entries are written to the server log. Since the target of web usage mining is to find out the browsing patterns or behavior of different users so we have to remove all additional requests which are not requested by the users. To remove those extra lines we need to clean the log file. The cleaning process of pre-processing includes the removal of unnecessary and irrelevant entries from the log files. In this cleaning process every URL is checked for suffixes like jpeg, jpg, gif, css etc. URLs that contain these suffixes are removed from the log files. Data cleaning is also done based on the status codes which are returned by the server. Status codes are

3.2. WEB LOG ANALYZER

used to describe the status of each request. By examining the status code we can figure out whether the submitted request was successful or it was a failure. The status code of successful request ranges from 200-299 [18]. Entries with status code less than 200 and greater than 299 are not successful requests. So all the entries having failure status codes can be removed from the log files.

In our implementation to clean the logs we have used multi-threading on a multi-core processor. Initial implementation of single threading was really slow to remove the noise from the log files. Multi-threading made the processing faster. To clean the log every thread executes the following algorithm:

Algorithm 1 Cleaning Server Log File

Require: Library's web server log file

Ensure: Noise free data is saved in the SQL Server Database

- 1: Store all the invalid extensions such as: gif,jpeg,jpg,css etc. in a buffer
 - 2: Read a log entry from the log file
 - 3: **if** the URL in the log entry contains any of the extensions stored in step1 OR the status code is greater than 299 or smaller than 200 **then**
 - 4: REMOVE that entry from the log file
 - 5: **else**
 - 6: STORE the entry in the database
 - 7: **end if**
 - 8: REPEAT steps 2 and 3 until the end of file of web log file is reached
 - 9: RETURN
-

Once the log file is cleaned we start to identify the users. In the EZproxy log a user can be identified in two ways: i) By the IP address and ii) By the userid. In our system we have used both of the approaches to identify the users. After user identification next

step is to identify the sessions created by different users. User session identification in a EZproxy server log is relatively easy because the log file stores the session id as well. To calculate the session length we have used the following algorithm:

Algorithm 2 Sessionization

Require: Noise free Log Database

Ensure: A list of distinct sessionids along with session length

- 1: Read a log entry from the database
 - 2: For every sessionid store the start time and end time of that session
 - 3: REPEAT step 1 and 2 until the End of File of Log Database
 - 4: For every sessionid get the difference between start time and end time
 - 5: RETURN
-

3.2.2 URL Categorization

Libraries have subscriptions to a number of research databases. Different research databases have their own URL structure to access resources like journal articles, e-books and also they have different URL structures for database queries. So we have categorized the URLs into two groups: Resource Group and Query Group. Following is an example of a resource group URL:

site.ebrary.com:80/lib/oculryerson/docDetail.action?docID=10017060

And the following is an example of query group URL:

web.a.ebscohost.com:80/ehost/results?sid=67072093-b7d2-4bb0-b27e-dac464b238b4
&bquery=%22family+experience%22

Different vendor has got different patterns for resource group url and query group url. To categorize the URLs all these patterns of resource group and query group urls

3.2. WEB LOG ANALYZER

are stored in database. Once the URL categorization is done next step is to retrieve article information like journal id, page numbers, volume number etc. from the Resource Group URLs. To categorize the URLs and to retrieve information from the URLs we have applied Algorithm-3.

Algorithm 3 URL Categorization and Information Retrieval

Require: Noise free URLs from the Web Log and URL structure of search queries

Ensure: Resource Group URLs and Query Group URLs along with the retrieved information

- 1: Read a URL from the weblog
 - 2: **if** the structure of the URL is similar to any of the patterns of the URLs of search queries **then**
 - 3: ADD this URL to Query Group and retrieve the query text
 - 4: **else**
 - 5: ADD this URL to Resource Group and retrieve information like journal id, page numbers, volume number etc.
 - 6: **end if**
 - 7: REPEAT step 1 and step 2 until the End of File of Web Log of Noise Free URLs
 - 8: RETURN
-

In the next step we will focus on the Visualization part of the Log Analyzer.

3.2.3 Log Analyzer and Visualization

After the pre-processing and URL categorization, our log analyzer needs to consider the specific requirement from librarians about how the log should be analyzed. The main target is to identify how different e-resources are being used, how patrons in individual department or faculty level use library, how different types of patrons use the library resources, as well as the dynamics or trends during different days of the year or different time of the day. Based on these requirements the following analytics tasks are identified:

- Database accessed by different department/faculty/patron based on total number of sessions, duration and number of hits: It will show the number of times (along with duration) each department/faculty/patron has accessed each database in a specific time period. Algorithm-4 shows the steps required to generate department wise database usage reports based on total number of sessions.
- Database accessed by number of unique users: It will show how many unique users are using every database in a specific time period to indicate the popularity of a database.
- Most accessed database: It will show the top databases based on usage duration/total hits/ total number of sessions in a specific time period.
- Timeslots showing busiest time of the server: It will show an overview of server traffic grouped by hours in a given period and it will help the library to figure out the busiest time in the server, so you can keep server free by not doing other tasks during the busiest time.
- Top database users based on total number of sessions, duration and number of hits: It will show top database users in a specific department/patron type (faculty/staff or student)/year of study (graduate or undergraduate, for undergraduate, 1st/2nd/3rd/4th year students) based on usage duration/hits/total number of sessions.

3.2. WEB LOG ANALYZER

- Least accessed databases: It will show the least accessed databases based on usage duration/total hits/ total number of sessions in a specific time period.
- Most/Least accessed articles: It will show the most/least accessed articles in a given period of time.

Figure 3.2: Library Log Analyzer System Design

Fig. 3.2 shows the system design diagram for the Library Log Analyzer. In the figure we see that the library log is passed through a noise reducer where different filters are applied. Filters are different texts such as jpg, css, gif etc. and they are applied to remove irrelevant log entries as explained before. Then the noise free data is processed to complete the analytics tasks.

Figure 3.3: Reports generated by the Data Visualization Tool

After the development of the Log Analyzer we worked to develop a visualization engine for the processed logs. To get a proper visualization of data we have represented information clearly and efficiently through graphics like bar charts and pie charts. Fig. 3.3

3.2. WEB LOG ANALYZER

Algorithm 4 Department wise database usage based on total number of sessions

Require: Department Name and a date range

Ensure: Database usage report

- 1: GET all the users for the given department and saved the users in a buffer
 - 2: READ one user from the buffer
 - 3: COUNT the number of created sessions grouped by different databases by the user from step-2
 - 4: REPEAT step-2 and step-3 until the End of the buffer
 - 5: RETURN
-

shows the names of the reports generated by the visualization tool. Now to select the type of chart we have applied the following principles:

- To visually represent categorical data bar charts are used. Categorical data can also be grouped based on different criteria. For example to represent monthly database usage we have used bar charts.
- To visually represent percentage values pie charts are used. For example to represent percentage usage of a database by different departments we have used pie charts.
- To visually represent a trend in data over intervals of time line charts are used. For example to represent server usage by library users we have used line charts.

To work with the data visualization engine is the last step to develop the log analyzer.

The SQL statements which are generated by the log analyzer are used by the report generator to generate different types of visualizations.

3.3 Clustering of e-resource items

In the second stage of our work, we try to cluster various e-resource items such as books, journals, articles etc. The clustering results can be used for easier browsing of library resources, recommending new resources to users, as well as helping the library to identify some hidden relationship between different resources. In our research we have applied different types of clustering algorithms on the library dataset. There are mainly two types of clustering algorithms: Content Based Clustering and Association Rule Based Clustering. In the following sections the details of the algorithms are described.

3.3.1 Content Based Clustering

3.3.1.1 Introduction

We can cluster all the documents based on their contents (in this work, their titles). To prepare the input dataset for the clustering algorithms, we have developed a dictionary with all the words from titles of all the documents. Words that are not necessary were removed based on a stop word list. A stop word list is defined as a collection of words that are not important such as: I, a, about, an, and, are, etc. Then we constructed the datasets in two ways: i) based on the Inverse document frequency (IDF) values of each word of every title, ii) based on the appearance of each word of every title - here the resultant matrix is a binary matrix. To generate the IDF values we have used equation (3.1):

$$idf_i = \log \frac{N}{df_i} \quad (3.1)$$

where N is total number of documents and df_i is the number of documents that have the i -th term. Then on the resultant matrix we have applied different clustering algorithms.

3.3.1.2 K-means Clustering

K-means clustering is a very popular cluster analysis algorithm in the data mining world. K-means algorithm groups the data in K clusters. K-means clustering is done based on the nearest mean and we have to supply the value of K . To calculate the distance of every point from the cluster we use Euclidean distances.

3.3.1.3 Farthest First Clustering

Farthest First clustering algorithm is similar to k-means algorithm. It also chooses centroids and assigns different items in cluster based on maximum distance whereas in K-means clustering it is done based on the nearest mean. A random seed value is required for the clustering algorithm. Cluster center is chosen depending on the seed value. It also uses Euclidean distances.

3.3.1.4 Filtered Clustering

In filtered clustering any arbitrary clustering algorithm is used and the clusterer passes through an arbitrary filter. The structure of clusterer as well as the structure of the filter

is based on the dataset. When the filter processes the test instances the structure of the filter is not changed. In our case we have selected K-means as the base clusterer and a filter that removes the useless attributes. This unsupervised filter automatically removes attributes that do not vary at all or that vary too much.

3.3.1.5 Density Based Clustering

In density-based clustering algorithms local density of the points are used to determine the clusters, not only the distance between points. Here a circular area of radius *Eps* around point X is defined. The circular area is known as the Eps-neighbourhood of X. For any point B, we say that B is a core point if there are at least *minpts* points in its neighbourhood, here *minpts* is a user-defined parameter that provides minimum threshold for the local density. If a point does not meet the *minpts* threshold but it belongs to the neighbourhood of some other core point then that point is known as a border point. If a point is neither a core nor a border point then it is known as a noise point or an outlier. At first the algorithm computes the neighbourhood for each point in the dataset and checks if it is a core point. Then the algorithm recursively finds all other points connected to it and all such points belong to the same cluster. Some border point may be reachable from core points in more than one cluster. Such points are arbitrarily assigned to one of the clusters.

3.3.1.6 Hierarchical Clustering

Application of hierarchical clustering builds a hierarchy of clusters. It can be applied in two ways: i) Agglomerative and ii) Divisive. In this work, we used bottom-up approach. To cluster a set of N items we can start by assigning each item to its own cluster i.e. initially we have N clusters for N items and size of each cluster is one. Then two closest pair of clusters are found and merged into a single cluster. At the end of the algorithm we have a single cluster of size N . We have used single link clustering, in which the distance between two clusters is considered to be equal to the shortest distance from any member of one cluster to any member of the other cluster.

3.3.1.7 Expectation and Maximization Clustering

The expectation-maximization algorithm is an iterative approach which finds the parameters of the probability distribution that has the maximum likelihood of its attributes. The input to the algorithm is the dataset, the total number of clusters, the maximum number of iterations and the accepted error. In every iteration the expectation step is executed and this step estimates the probability of each point belonging to each cluster. Then the maximization step re-estimates the parameter vector of the probability distribution of each class and this value is used in the next step. After each iteration is performed a convergence test is done. This test verifies if the difference of the attributes vector of an iteration to the previous iteration is smaller than the acceptable error tol-

erance. The algorithm finishes when the distribution parameters converges or it reaches the maximum number of iterations.

3.3.2 Association Rule Based Clustering

Association rule can identify different e-resource items accessed in the same session which can give better insight to library about how their resources are being used. It can also be used for recommendation and to improve the structure of the website. We can reorganize the website so that the most frequently browsed articles are easily accessible to the users and this will increase the usability of the library website. By clustering based on the association rules, some interesting results could be achieved. In the following subsections we have described how association rule based clustering is done.

3.3.2.1 Introduction

An association rule is defined by the expression: $X \implies Y$ where X and Y are itemsets, itemsets are defined as a collection of one or more items. It states that items in X and Y together form a frequent itemset. Association Rules are used to find interesting relationships among the data items of a large database. There are different types of association rule mining algorithms available. To use the algorithms we need to create transactions of itemsets. In our dataset every transaction consists of all the URLs that are accessed by a specific user in a specific user session. When we apply association rule mining algorithms on these transactions we can find which e-resource items are frequently

accessed together by the users of the library website. The result of this analysis is really helpful to find hidden relationship between books, articles etc. Then we have applied clustering on these generated association rules to group the similar articles.

3.3.2.2 Generating Association Rule

Agrawal in [2] came up with the idea of association rule mining. They apply the algorithm on a set of n distinct items say $X = X_1, X_2, \dots, X_n$. They define T as a transaction which contains items from X that is $T \subseteq X$ and all the transactions are stored in a database. According to their definition association rule is expressed by the following:

$$(X_1, X_2) \implies X_3 \quad (3.2)$$

This rule means X_1 and X_2 imply X_3 i.e. if a X_1, X_2 and X_3 are three articles then any user who reads articles X_1 and X_2 there is a good chance that he is going to read article X_3 . To measure the chance of appearance of article X_3 association rules provide two basic important attributes: support and confidence. Support and confidence are two user defined parameters. Minimum support and minimum confidence are set by the user to generate the most frequent itemsets. The percentage of records that contain $(X_1, X_2) \cup X_3$ to the number of total records in the database is known as the support for the association rule. If support of an association rule is 5% then it means 5% of the total records in the database contain this rule. So support can be written as follows:

$$Support((X_1, X_2) \implies X_3) = \frac{\text{Count of records having } X_1 X_2 X_3}{\text{Total number of records in the database}} \quad (3.3)$$

The association rules that have lower support are not frequent in the database so rules

having lower support will be ignored by the users. Before running the algorithm user can set the support level and then he can generate all the rules whose support levels are more than the minimum support level.

Another important parameter for association rule mining is confidence. The percentage of the number of records that contain $(X_1, X_2) \cup X_3$ to the total number of records that contain (X_1, X_2) is considered as confidence. The strength of any association rule is measured by confidence. This value can be set by the users and all the association rules that have confidence greater than or equal to the minimum confidence can be generated. If confidence for the association rule $\{(X_1, X_2) \implies X_3\}$ is 90% then it means 90 percent of the records that contain (X_1, X_2) also contain X_3 . Confidence for this association rule can be written as:

$$Confidence\{(X_1, X_2) \implies X_3\} = \frac{Support(X_1X_2X_3)}{Support(X_1X_2)} \quad (3.4)$$

In our research to find out surprise relationship between the documents we have set support to a very low value: 1% and to get the strong rules we have set confidence to 80%. As the minimum support level is low, the association rule mining algorithm will generate almost all the rules greater than or equal to minimum confidence level. Now that association rules are generated, our next task is to cluster these rules.

3.3.2.3 Hypergraph Generation

Han et al. in [20] used a graph partitioning algorithm to cluster market basket data based on association rule mining. On their research they have shown that clustering of

association rules based on hypergraphs show better performance than other approaches. Before we can apply a hypergraph partitioning algorithm we have to generate a hypergraph. A weighted hypergraph has been constructed from the association rules. A hypergraph [6] consists of a set of vertices and a set of hyperedges. The difference between a graph and a hypergraph is that each hyperedge can connect more than two vertices whereas in a graph one edge can connect only two vertices. The vertex set of the hypergraph that we have generated corresponds to the distinct documents of the library and hyperedges correspond to the association rules that is one association rule is termed as one hyperedge. In our model we have used confidence as the weight of each hyperedge. To generate the hypergraph the weight and hyperedge were saved in a hypergraph file. As the hypergraph is generated now is the task to partition the hypergraph.

3.3.2.4 Partitioning the Hypergraph

To partition the hypergraph we have used HMETIS. HMETIS is a standalone program [26] which is based on a multilevel fast partitioning algorithm [25] that can produce high quality balanced k way partitions where k is specified by the user. A multilevel graph partitioning algorithm has got different phases including: i) Coarsening phase, ii) Initial partitioning phase, and iii) Uncoarsening and refinement phase.

During the coarsening phase smaller hypergraphs are constructed, because bisection of a smaller hypergraph provides better result than the result obtained from bisecting a larger hypergraph. So after several levels of coarsening, the size of large hyperedges

are reduced to hyperedges that connect just a few vertices. If G_0 is a hypergraph then coarsening of G_0 will generate a sequence of graphs $G_1, G_2, \dots, G_N$, where G_N has the fewest number of vertices.

During the initial partitioning phase the smallest hypergraph is selected and the bisection of the selected hypergraph is computed. To partition the hypergraph a vertex is randomly selected and using a breadth-first movement the hypergraph grows around the region until half of the vertices are in this region. Then vertices those are found in the grown region are assigned to the first part, and the rest of the vertices are assigned to the second part.

During the uncoarsening phase partitioning of the coarser hypergraph is successively projected to the next-level finer hypergraph. Then to improve the quality of the partitioning the refinement phase is used. In the refinement phase vertices are moved repeatedly between partitions in order to improve the solution quality. Hence the refinement phase ends the partitioning process.

So far in this chapter we have described how all the clustering algorithms are applied on the library dataset. In the next chapter we are going to analyze the results that are generated in this chapter.

Chapter 4

Experiments

In this chapter, we explain about prototype implementation, experiment design and how we prepare the input dataset, and then we compare the results of different clustering algorithms. To complete the experiment we have used: i) WEKA to apply clustering algorithms like K-means Clustering algorithms, Filtered Clustering algorithm etc. ii) HMETIS graph application to cluster the association rules, iii) programming language Java to process the input dataset and to compare the results of different clustering approaches, iv) SQL Server 2012 as the database server, and v) iReport [12] from Jasper-soft is used to visualize the results from the Log Analyzer. The experiment was run on a computer with Intel core i7 processor 2630QM, 2.0 GHz clock speed, 8GB RAM and Windows-7 as the operating system.

4.1 Prototype Implementation

The implemented prototype has three modules: i) Data pre-processing module, ii) Data visualization module, and iii) Input data preparation module for different types of clustering algorithms. Fig. 4.1 shows the Graphical User Interface that is used to pre-process the ezproxy logs. After the pre-processing we saved all the noise free data into another file and later these noise free data is saved into the database for further processing.

As the data is saved into the database, we used iReport to visualize the required reports. Our program uses the iReport API and the program forwards different SQL statements related to different reports to the iReport engine. Then iReport generates various reports based on the submitted queries. Fig. 4.2 is showing one of the Graphical User Interfaces that is used to generate different reports on Database access by the departments based on duration, total hits and total sessions. The visualizer also generates reports on different database access by unique users, database access by all users, it also generates report that shows the busiest time of the server etc. A few example reports are shown in Fig. 4.3, 4.4, 4.5 and 4.6.

In the third module we have included WEKA and HMETIS APIs in our Java code. We wrote different Java functions and TSQL to generate the input dataset for WEKA and HMETIS. We also wrote Java functions to compare and evaluate different clusters created by the various clustering algorithms. In the following section we explain about

4.1. PROTOTYPE IMPLEMENTATION

Figure 4.1: Graphical User Interface for Log File Preprocessing

Figure 4.2: Graphical User Interface for Log Visualization

4.1. PROTOTYPE IMPLEMENTATION

Figure 4.3: Database Accessed by Unique Users

Figure 4.4: Download Chart for a Single Database

Figure 4.5: Server usage by library users based on Hits

Top 10 Departments Based On Total Duration
Database Name: www.ebrary.com

Figure 4.6: Top 10 Departments based on a single database usage

the experiment design.

4.2 Experiment on Clustering Algorithms

In our experiment we have applied different clustering algorithms on the library data set to cluster the data. The target of our experiment is to figure out the best clustering algorithm by analyzing the generated results of the clustering algorithms. In our approach we will compare the quality of the generated clusters and also the time efficiency of the clustering algorithms.

Figure 4.7: Count of items for different clusters in Manual Clustering

Figure 4.8: Count of items for different clusters in HMETIS Clustering

Figure 4.9: Count of items for different clusters in K-means Clustering

Figure 4.10: Count of items for different clusters in FarthestFirst Clustering

Figure 4.11: Count of items for different clusters in Filtered Clustering

Figure 4.12: Count of items for different clusters in MakeDensityBased Clustering

Figure 4.13: Count of items for different clusters in EM Clustering

Figure 4.14: Count of items for different clusters in Hierarchical Clustering

First we prepared dataset for the HMETIS application. As HMETIS works on association rules so at first we generated association rules using apriori algorithm, which is used for frequent itemset generation and apriori algorithm is a built-in feature of WEKA. WEKA is an application where we have a collection of machine learning algorithms for data mining tasks. We have worked with all the documents from www.ebrary.com. To generate association rules we had to prepare the dataset for WEKA. To identify the frequent users who use the library most we selected 2000 most browsed documents from the ebrary database and then selected all the sessions that have all these top browsed documents. Then we applied WEKA on this dataset to generate the frequent itemsets i.e. the association rules. Next we used Java to format WEKA output and after the

formatting Java provides the input file for the HMETIS application. From WEKA we have generated 316 association rules so for HMETIS application we have 316 hyperedges and 254 vertices i.e. 254 ebrary titles.

Again to test the accuracy of the clustering results we had to have the knowledge of the correct clusters. To accomplish the task we group the top browsed 254 ebrary titles into 17 clusters by manual inspection. Fig. 4.7 shows the count of items for different clusters in Manual Clustering. As the input for HMETIS is ready, we have applied the HMETIS algorithm to partition the association rules. To invoke HMETIS we have to provide 9 or 10 command line arguments, such as the following:

hmetis HGraphFile Nparts UBfactor Nruns CType RType Vcycle Reconst dbgvl

OR

hmetis HGraphFile FixFile Nparts UBfactor Nruns CType RType Vcycle Reconst dbgvl

The meaning of the various parameters is as follows:

- **HGraphFile:** The name of the file that stores the hypergraph
- **FixFile:** To store information about the pre-assignment of vertices
- **Nparts:** The number of desired partitions
- **UBfactor:** To specify the allowed imbalance
- **Nruns:** The number of the different bisections that are performed by hmetis
- **CType:** The type of vertex grouping scheme

- **RType:** The type of refinement policy to use during the uncoarsening phase.
- **Vcycle:** To select the type of V-cycle refinement
- **Reconst:** To select the scheme to be used in dealing with hyperedges
- **dbgvlvl:** To print debugging information.

To generate the clusters we invoked HMETIS as the follows:

```
HMETIS hmetis.hgr 17 5 20 4 1 3 1 0
```

Once the run is successful HMETIS generates a file with all the clustering items. In here 17 clusters have been created. We invoked the HMETIS program to generate 17 clusters because the manual clustering has got 17 clusters. HMETIS application saves the clustering information in an output file. Next we have used a Java function to parse that file to figure out the clustering information. Fig. 4.8 shows the count of items in different clusters in case of HMETIS clustering. From the figure it is viewed that the output of HMETIS clustering is well balanced.

Then we prepared the dataset for different content based clustering algorithms. To prepare the dataset we selected the same 254 articles which were used to generate the input for HMETIS algorithm. Using Java the titles were splitted and then different words were saved in database to form a dictionary. In the dictionary we have 957 words. We used Java to generate a 254 x 957 matrix of term weights (IDF values) for all the titles. Then we applied different content based clustering algorithms on this matrix to generate

17 clusters. Fig. 4.9 shows the count of different items for every cluster in case of K-means clustering. Fig. 4.10 shows the count of different items for every cluster in case of Farthest first clustering. Fig. 4.11 shows the count of different items for every cluster in case of Filtered clustering. Fig. 4.12 shows the count of different items for every cluster in case of Density based clustering. Fig. 4.13 shows the count of different items for every cluster in case of EM clustering. Fig. 4.14 shows the count of different items for every cluster in case of Hierarchical clustering. From all these graphs we see that the clustering of the documents are not balanced as most of the documents are kept in one cluster.

In the following section we will analyze the results of different clustering algorithms.

4.3 Results Analysis for Clustering Algorithms

In this section we analyze the results generated by the different clustering algorithms. To calculate the matching percentage with the manual clusters for the clusters generated by every clustering algorithm we have used the following equation:

$$TotalMatchQuantity = \frac{1}{N} \sum_{i=1}^k max_j |W_k \cap C_j| \quad (4.1)$$

In equation (4.1) W_k is a set of items in the generated k-th cluster and C_j is a set of items in the manually generated j-th cluster. Fig. 4.15 shows the percentage match amount for every clustering algorithm. From the figure we see that association rule based clustering i.e. HMETIS has the highest matching score and all other Content Based Clustering algorithms have not generated better clusters than the association rule based HMETIS

Figure 4.15: Evaluating Clustering algorithms based on percentage match

clustering. Out of all the content based clustering algorithms K-mean clustering algorithm came up with the best score. So we can say Association rule based clustering generates better clusters comparing to other content based clustering algorithms.

Next we compare the processing time required by all the algorithms. Fig. 4.16 shows that running time for the expectation maximization algorithm is the highest. Again we see association rule based clustering algorithm i.e. HMETIS takes the least amount of time to generate all the clusters but it also needs time to generate the association rules as well which is 800 ms on an average. So altogether the total required time is 1250 ms. So we can say association rule based clustering algorithm resides in the moderate range in case of time efficiency. Using association rule based clustering algorithm, we have also found the

Figure 4.16: Running time of different clustering algorithms

hidden relationships between different documents as shown in the following examples. These relationships are not found by any of the content-based clustering algorithms because content-wise they are not similar. However they are frequently accessed by the users together.

- “The Hidden Factor in Climate Policy: Implicit Carbon Taxes” → “Brief History of Neoliberalism”
- “Advancing a Policy Dialogue: An Overview of Policy Goals, Objectives, and Instruments for the Agri-Food Sector” → “The Hidden Factor in Climate Policy: Implicit Carbon Taxes”
- “Chaos and Complexity Theory for Management : Nonlinear Dynamics” → “Big Data and Business Analytics”

4.3. RESULTS ANALYSIS FOR CLUSTERING ALGORITHMS

- “Semiotic Landscapes : Language, Image, Space” -> “Video games and Art”
- “Under the Drones : Modern Lives in the Afghanistan-Pakistan Borderlands” -> “Privacy and Drones: Unmanned Aerial Vehicles”
- “Colour By Numbers: Minority Earnings in Canada 1996-2006” -> “Why Do Skilled Immigrants Struggle in the Labor Market? A Field Experiment with Six Thousand Resumes”

In this chapter we have analyzed the results that are generated by the clustering algorithms. We have found that association rule based clustering algorithm generates the most accurate cluster with moderate time efficiency. Using this approach, documents which are not similar to each other based on their contents may be put in one cluster, which in a way could provide some interesting information to librarians. So we can use association rule based clustering in our future work. In the next chapter we focus on the future work and conclusion.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis work we have applied library analytics on the library web log data. Our research target was to perform general analytic tasks that are required by the librarians and to represent the result graphically, and along with the general analytical tasks we worked with the problem of clustering of different books, journals, articles etc. in the context of library analytics. By investigating different clustering approaches we wanted to find out an approach which can find similarity among different e-resource articles based on their contents and also based on users' browsing patterns. In our research we have accomplished the following:

1. We have designed and implemented a library log analyzing system which is able to

analyze the library log data and it also generates different types of reports for the library analytics system.

2. We have tested and compared different document clustering algorithms based on the title of the documents and users' browsing behavior.
3. We have found that association rule based clustering is able to find out hidden and most accurate relationships with comparable efficiency.

5.2 Future Work

Library analytics is expanding and growing to include different features which are going to help the users of the library. There are a few directions we would like to work on in the future:

- Firstly we will work to make a plugin of our analytics system so that it can be integrated with any other system easily.
- Next we will connect our analytics system to the student record database and it will help us to figure out information such as how students' performance is related to the library usage.
- And at last we will design a recommender system based on the clustering result.

Bibliography

- [1] C. C. Aggarwal and C. Zhai. *A survey of text clustering algorithms*. Mining Text Data, Springer US, pp.77-128, 2012.
- [2] R. Agrawal, T. Imieliński, and A. Swami. *Mining association rules between sets of items in large databases*. ACM SIGMOD Record, Vol.22, Issue.2, pp.207-216, 1993.
- [3] N. Alldrin, A. Smith, and D. Turnbull. *Clustering with EM and K-means*. University of San Diego, California, Tech Report, pp.261-95., 2003.
- [4] J. Arendt and C. Wagner. *Beyond Description: Converting Web Site Usage Statistics into Concrete Site Improvement Ideas*. Journal of Web Librarianship, Vol.4, No.1, pp.3754, 2000.
- [5] F. Beil, M. Ester, and X. Xu. *Frequent term-based text clustering*. In the Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp.436-442, 2002.
- [6] C. Berge. *Graphs and hypergraphs*. ELSEVIER, Vol.7, pp.389-400, 1973.
- [7] D. Blečić. *Usage data for collection, development and management: Changes, Issues, Local Utilization, and Shared Possibilities*. Available from oclc.org/content/dam/oclc/events/2013/ALA_Annual.2013/LibraryAnalytics_Blečić_29Jun_1pm.pdf (Accessed, 2nd February 2015), 2013.
- [8] L. Bridges. *Who is not using the library? A comparison of undergraduate academic disciplines and library use*. The Johns Hopkins University Press, Portal: Libraries and the Academy, Vol.8, No.2, pp.187-196, 2008.
- [9] R. Chen, R. Anne, and B. Benjamin. *How people read books online: mining and visualizing web logs for use information*. Research and Advanced Technology for

- Digital Libraries. Springer Berlin Heidelberg, pp.364-369, 2009.
- [10] E. Collins and G. Stone. *Understanding Patterns of Library Use Among Undergraduate Students from Different Disciplines*. Evidence Based Library and Information Practice, Vol.9, No.3, pp.51-67, 2014.
 - [11] J. Collins, E. and Palmer and G. Stone. *The LAMPscape: Library Analytics and Metrics Project (LAMP)*. Sconul and Jisc workshop on library analytics and metrics, Available from <http://eprints.hud.ac.uk/20229/> (Accessed, 1st January 2015), 2014.
 - [12] Jaspersoft Community. *iReport*. Available from <https://community.jaspersoft.com/project/ireport-designer> (Accessed 25th February 2015), 2015.
 - [13] A. Cooper. *What is analytics? Definition and essential characteristics*. CETIS Analytics Series; Vol.9, No.5, pp.1-10, 2012.
 - [14] B. Cox and M. Jantti. *Discovering the Impact of Library Use and Student Performance*. Educase review, Available from <http://www.educause.edu/ero/article/discovering-impact-library-use-and-student-performance> (Accessed, 2nd February 2015), 2012.
 - [15] W. Dakka and S.P. Cucerzan. *Document clustering based on entity association rules*. Google Patents, Available from <http://www.google.com/patents/US7617182> (Accessed, 2nd February 2015), 2009.
 - [16] Penn Library DataFarm. *Metridoc Project*. Available from <https://metridoc.library.upenn.edu/> (Accessed 18th February 2015), University of Pennsylvania, 2011.
 - [17] C. B. Do and S. Batzoglu. *What is the expectation maximization algorithm?* Nature biotechnology, Vol.26, No.8, pp.897-899, 2008.
 - [18] Fielding et al. *10. Status Code Definitions*. Available from <https://tools.ietf.org/html/rfc2616/#section-10> (Accessed 18th February 2015), 1999.
 - [19] B. D. Gerardo, Y. Byun, and B. Tanguilig III. *Hierarchical Clustering and Association Rule Discovery Process for Efficient Decision Support System*. Communication and Networking, Communications in Computer and Information Science Vol.266, pp.239-247, 2011.

- [20] E. Han, G. Karypis, and B. Mobasher. *Clustering Based On Association Rule Hypergraphs*. Workshop on Research Issues on Data Mining and Knowledge Discovery, pp.9-13, 1997.
- [21] T. Hastie, R. Tibshirani, and J. Friedman. *14.3.12 Hierarchical clustering*. The Elements of Statistical Learning (PDF) (2nd ed.). New York: Springer, pp.520528, ISBN 0-387-84857-6, 2009.
- [22] EDUCAUSE Learning Initiative. *7 Things you should know about ANALYTICS*. Available from <http://net.educause.edu/ir/library/pdf/ELI7059.pdf> (Accessed, 24th January 2015), April 2010.
- [23] K. Jager. *Successful students: does the library make a difference?* Performance Measurement and Metrics, MCB UP Ltd, Vol.3, No.3, pp.140-144, 2002.
- [24] D. M. Kanungo, T. and Mount, C. D. and Silverman R. Netanyahu, N. S. and Piatko, and A. Y. Wu. *An efficient k-means clustering algorithm: Analysis and implementation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.24, No.7, pp.881-892., 2002.
- [25] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. *Multilevel hypergraph partitioning: applications in VLSI domain*. Very Large Scale Integration (VLSI) Systems, IEEE Transactions, Vol.7, Issue.1, pp.69-79, 1997.
- [26] G. Karypis and V. Kumar. *A hypergraph partitioning package*. Army HPC Research Center, University of Minnesota, Department of Computer Science & Engineering, 1998.
- [27] G. Karypis, V. Kumar, R. Smith, and G. Mason. *Retrieval, Analysis, and Presentation Toolkit for usage of Online Resources (RAPTOR)*. Available from <http://iam.cf.ac.uk/RAPTOR> (Accessed 25th February 2015), Cardiff University, 2011.
- [28] D. Kiron, R. Shockley, N. Kruschwitz, G. Finch, and M. Haydock. *Analytics: The widening divide*. MIT Sloan Management Review, Vol.53, No.3, pp.1-22, 2011.
- [29] W. A. Kusters, E. Marchiori, and A. J. Oerlemans. *Mining Clusters with Association Rules*. Advances in Intelligent Data Analysis, Vol.1642, pp.39-50, 1999.

- [30] H. Kriegel, J. Krger, P. and Sander, and A Zimek. *Density-based clustering*. Data Mining and Knowledge Discovery, John Wiley and Sons, Vol.1, Issue.3, pp.231-240, 2011.
- [31] S. G. Langhnoja, P. B. Mehul, and B. M. Darshak. *Web Usage Mining to Discover Visitor Group with Common Behavior Using DBSCAN Clustering Algorithm*. International Journal of Engineering and Innovative Technology (IJEIT), Vol.2, pp.169-173, 2013.
- [32] S. G. Langhnoja, P. B. Mehul, and B. M. Darshak. *Web Usage Mining Using Association Rule Mining on Clustered Data for Pattern Discovery*. International Journal of Data Mining Techniques and Applications, Vol.2, Issue.1, ISSN: 2278-2419, pp.141-150, 2013.
- [33] G. Liu, S. Huang, C. Lu, and Y. Du. *An improved K-Means Algorithm Based on Association Rules*. IJCTE ,ISSN: 1793-8201, Vol.6, No.2, pp.146-149, 2014.
- [34] H. Malik and Kender R. *Clustering Web Images using Association Rules, Interestingness Measures, and Hypergraph Partitions*. Proceedings of the 6th International Conference on Web engineering, ACM, pp.48-55, 2006.
- [35] K. Marek. *Using Web Analytics in the Library*. Library technology Reports Expert Guides to Library Systems and Services, Vol.47, No.5, ISSN 0024-2586, pp.1-50, 2011.
- [36] N. Negm, M. Amin, P. Elkafrawy, and A. Badeeh M. Salem. *Investigate the Performance of Document Clustering Approach Based on Association Rules Mining*. IJACSA, Vol.4, No.8, pp.142151, 2013.
- [37] M. Plasse, N. Niang, A. Saporta, G. and Villeminot, and L. Leblond. *Combined use of association rules mining and clustering methods to find relevant links between binary rare attributes in a large data set*. Computational Statistics & Data Analysis, Vol.52, No.1, pp.596-613, 2007.
- [38] M. Poongothai, K. and Parimala and S. Sathiyabama. *Efficient Web Usage Mining with Clustering*. International Journal of Computer Science, Vol.8, Issue.6, No.3, pp.203-209, 2011.
- [39] N. Sahoo, J. Callan, R. Krishnan, G. Duncan, and R. Padman. *Incremental Hierarchical Clustering of Text Documents*. Proceedings of the 15th ACM Interna-

- tional Conference on Information and Knowledge Management,CIKM, pp.357-366, 2006.
- [40] K Santhisree, A Damodaram, S Appaji, and D NagarjunaDevi. *Web usage data clustering using DBSCAN algorithm and set similarities*. International Conference on Data Storage and Data Engineering,IEEE, pp.220-224, 2010.
- [41] B. Showers, J. Palmer, and G Stone. *JiscLAMP: shining a light on our analytics and usage data*. UKSG 37th Annual Conference and Exhibition, pp.14-16, 2014.
- [42] B. Showers and G. Stone. *Developing a Shared Analytics Services for Academic Libraries*. 10th Northumbria International Conference on Performance Measurement in Libraries and Information Services, pp.22-25, 2013.
- [43] V.K. Singh, N. Tiwari, and S. Garg. *Document Clustering using K-means, Heuristic K-means and Fuzzy C-means*. International Conference on Computational Intelligence and Communication Networks (CICN),pp.297-301, 2011.
- [44] J. Srivastava, R. Cooley, M. Deshpande, and P. Tan. *Web usage mining: Discovery and applications of usage patterns from web data*. ACM SIGKDD Explorations Newsletter, Vol.1, No.2, pp.12-23, 2000.
- [45] M. Steinbach, G. Karypis, and V. Kumar. *A comparison of document clustering techniques*. In KDD workshop on text mining, Vol.400, No.1, pp.525-526, 2000.
- [46] G. Stone and E. Collins. *Library usage and demographic characteristics of undergraduate students in a UK university*. Performance Measurement and Metrics, Emerald Group Publishing Limited, Vol.14, No.1, pp.25-35, 2013.
- [47] OCLC Support. *EZproxy Configuration: LogFormat*. Available from <https://www.oclc.org/support/services/ezproxy/documentation/cfg/logformat.en.html> (Accessed 10th February 2015), 2015.
- [48] J. Szymkowiak, A. and Larsen and L. K. Hansen. *Hierarchical clustering for datamining*. In Proceedings of KES-2001 Fifth International Conference on Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies, pp.261-265, 2001.
- [49] P. Tan, M. Steinbach, and V. Kumar. *8.4 DBSCAN*. Introduction to Data Mining, Addison-Wesley, pp.526-532, 2006.

- [50] LLC Taylor & Francis Group. *Web Analytics: A Picture of the Academic Library Web Site User*. Journal of Web Librarianship, Vol.3, No.1, pp.3-14, 2009.
- [51] D. Waisberg and A. Kaushik. *Web Analytics 2.0: Empowering Customer Centricity*. The Original Search Engine Marketing Journal, Vol.2, No.1, pp.5-11, 2009.
- [52] M. Wilson. *Understanding the needs of tomorrow's library user: Rethinking library services for the new age*. Australasian Public Libraries and Information Services Vol.13, No.2, pp.81, 2000.
- [53] S. Wong and T. Webb. *Uncovering meaningful correlation between student academic performance and library material usage*. Association of College & Research Libraries, crl-129, 2010.
- [54] B. Xu. *Clustering educational digital library usage data: Comparisons of latent class analysis and K-Means algorithms*. Journal of Educational Data Mining, Vol.5, No.2, pp.38-68, 2011.
- [55] J. Xu and H. Liu. *Web user clustering analysis based on KMeans algorithm*. International Conference on Information Networking and Automation (ICINA), Vol.2, pp.2-6, 2010.
- [56] R. Xu and D. Wunsch. *1.Cluster Analysis*. Clustering, Wiley-IEEE Press, pp.1-11, 2008.
- [57] R. Xu and D. Wunsch. *3.Hierarchical Clustering*. Clustering, Wiley-IEEE Press, pp.31-35, 2008.
- [58] R. Xu and D. Wunsch. *4.Partitional Clustering*. Clustering, Wiley-IEEE Press, pp.63-70, 2008.
- [59] Y. Zhao and G. Karypis. *Evaluation of Hierarchical Clustering Algorithms for Document Datasets*. Proceedings of the Eleventh International Conference on Information and Knowledge Management, CIKM, pp.515-524, 2002.
- [60] Y. Zhao, G. Karypis, and U. Fayyad. *Hierarchical clustering algorithms for document datasets*. Data mining and knowledge discovery, Vol.10, No.2, pp.141-168, 2005.