PATTERN DISCOVERY USING ASSOCATION RULE MINING ON CLUSTERED DATA

HTUN ZAW OO

M.C.Sc **AUGUST 2018**

PATTERN DISCOVERY USING ASSOCIATION RULE MINING ON CLUSTERED DATA

BY

HTUN ZAW OO

B.C.Sc. (Hons:)

A Dissertation Submitted in Partial Fulfilment of the Requirements for the Degree of

Master of Computer Science

(M.C.Sc.)

University of Computer Studies, Yangon

AUGUST 2018

ACKNOWLEDGEMENTS

Foremost, I would like to express my respectful gratitude to **Dr. Mie Mie Thet Thwin**, Rector, University of Computer Studies, Yangon, for giving me a chance to compile this thesis and for her kind general guidance.

My special thanks goes to my supervisor **Dr. Nang Saing Moon Kham**, Professor and Head of Faculty of Information Science, University of Computer Studies, Yangon, for her continuous support of my study and for her patience, motivation, enthusiasm, and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better supervisor and mentor for my study.

I also would like to thank **Dr. Thi Thi Soe Nyunt**, Professor and Head of Faculty of Computer Science, University of Computer Studies, Yangon and **Dr. Khin Nwe Ni Tun**, Professor, University of Computer Studies, Taung Gyi, for their kind advice and arrangement to complete this thesis.

I am grateful to **Daw Khine Yin Mon**, Lecturer, Department of Language, University of Computer Studies, Yangon, for editing my thesis from language point of view.

Last but not the least, I would like to thank my friend, Tin Maung for his encouragement and support from start to end of this thesis and thanks to all those who provided support directly or indirectly to accomplish this thesis.

ABSTRACT

Many organizations use World Wide Web for multipurpose platform during these days. It is very important to understand how a web site is being used by users. Web usage mining also known as web log mining, aims to discover interesting and frequent user access patterns from web browsing data that are stored in web server logs, proxy server logs or browser logs. Web usage mining involves the automatic discovery of patterns from one or more Web servers using web log data. Usage Mining tools discover and predict user behavior, in order to help designer, improve the web site, attract visitors, or give regular users a personalized and adaptive service. In this thesis, the aim is to find frequent user access pattern from web log entries. Combined effort of clustering and association rule mining is used to apply for pattern discovery. The 30 web log files are used from United Nations High Commissioner for Refugees. Density-based clustering spatial clustering application with noise (DBSCAN) has been used to group the users based on their access patterns and Apriori algorithm is applied to generate frequent user access patterns. As DBSCAN groups the user based on their access patterns, those users who don't share the similar access patterns are removed. Hence clustering reduces the data size and Apriori generates concise and relevant rules. The result from this system is highly depends on the parameters provided by users. This system is implemented using python programming language and SQLite is used a storage layer.

CONTENTS

LIST OF FIGURES

Non-clustered

LIST OF TABLES

LIST OF EQUATIONS

CHAPTER 1
INTRODUCTION

1.1 Evolution of Web Technology

Web technology is not evolving in comfortable and incremental steps, but it is turbulent, erratic, and often rather uncomfortable. The Internet, arguably the most important part of the new technological environment, has expanded with an unexpected speed. In recent years, the advance in computer and web technologies and the decrease in their cost have expanded the means available to collect and store data. As an intermediate consequence, the amount of information (meaningful data) stored has been increasing at a very fast pace. Traditional information analysis techniques are useful to create informative reports from data and to confirm predefined hypothesis about the data. However, huge volumes of data being collected create new challenges for such techniques as organizations look for ways to make use of the stored information to gain an edge over competitors. It is reasonable to believe that data collected over an extended period contains hidden knowledge about the business or patterns characterizing user profile and behavior. With the rapid growth of the World Wide Web, the study of knowledge discovery in web, modeling and predicting the user's access on a website has become very important. [12]

From the administration, business and application point of view, knowledge obtained from the web usage patterns could be directly applied to efficiently manage activities related to marketing strategies, web server performance, website maintenance, web page personalization and etc. Web is becoming the necessity of the businesses and organizations because of its demand from the clients. With the large number of companies using the Internet to distribute and collect information, knowledge discovery on the web has become an important research area. With the explosive growth of information sources available on the World Wide Web, it has become necessary for organizations to discover the usage patterns and analyze the discovered patterns to gain an edge over competitors.

Web mining can be divided into three areas, namely web content mining, web structure mining and web usage mining. Web content mining focuses on discovery of information stored on the Internet. Web structure mining focuses on improvement in

structural design of a website. Web usage mining, the main topic of this thesis, focuses on knowledge discovery from the usage of the web server logs.

1.2 Objectives of the Thesis

This section mentions how the proposed approach can be applied in the application domain. The advantages and objectives of the thesis are presented as below.

1. To study density based clustering techniques and its strengths.
2. To study how clustering could contribute as part of pre-processing step for association rule mining.
3. To discover frequent access patterns based on clustered data for Operational Data Portal of United Nations High Commissioner for Refugees (UNHCR).
4. To design of the new website development of organizations using frequent user access patterns discovered.
5. To generate more relevant and meaningful association rules to users.

1.3 Problem Statements

United Nation High Commissioner for Refugees (UNHCR) is one of the United Nations agencies that is providing international protection to persons of concern. Persons of concern are Refugees, Asylum-seekers, Internally Displaced Persons (IDPs), Returnees (refugees and IDPs), Stateless persons and others of concern to UNHCR. Data is the core asset to every organization in this age and doing analysis and discovering hidden patterns in the huge amount of data is very interesting and beneficial to the different stage of staff members in the organization. UNHCR has been using Operation Data Portal as information and data sharing platform to coordinate with other United Nations agencies and International Non-governmental Organizations (INGOs) since 2012. The portal is being upgraded to another a new site and it will be very interesting to know how the current website is being used by the users. Discovering user frequent access patterns help in designing the layout of the new website, personalized website based on users' interest. Therefore, web usage mining is applied to the web log data of Operational Data Portal to uncover user frequent access patterns. As for empirical study, UNHCR kindly contributed 30 web log files for this thesis. The

size of the files varies from 10 MB to 22 MB. After consolidating all 30 files, the total file size is more than 8 GB.

1.4 Related Works

With growing internet industries, web usage mining is one of popular areas where many research works have been done. Based on users' interest, web usage mining can be used for many different purposes. Discovering which web pages are mostly by users, detecting those users who have completely different usage patterns from other users, finding sequence of web pages which are accessed together are a few key areas of web usage mining. As this thesis focuses on finding frequent user access patterns from web log data, some related works on pattern discovery from web usage data are as below.

The author Khandakar Entenam Unayes Ahmed et al. have implemented a system to discover different patterns using web usage mining. In this system, users are identified converting IP address to domain name by using DNS lookup, cookies and cache busting. The system presents many different analysis from web log data. Using client IP address, countries which mostly access the website can be revealed. Path analysis which is sequence of pages the visitors like most or how long path they like to visit in a website, the number of hits and visitors count of the website and number of web log entries per day. [9]

Pattern discovery using association rules is developed by the author Kiruthika M, Rahul Jadhav et al. This paper uses the clustering approach to select the data based on client IP address. If count of client IP address is lower than certain threshold, then those web log entries are discarded but if the count is higher than the threshold then the web log entries are selected for session identification step. A session is identified if the time taken between pages is at least 5 minutes. Depending on the pages requested the entries of each session are classified among 5 different predefined classes. The final result is to generate association rules between client IP address and predefined class of the page which are accessed by that IP address. [2]

The author R Suguna and D Sharmila have presented association rule mining for web recommendation. This paper uses bird flocking algorithm for clustering the web logs. The bird flocking algorithm effectively preprocesses the web logs which fit

for the biological based. The aim of this paper is to generate association rules for web personalization. [5]

The author Aarti M. Parekh, Anjali S. Patel et al. have introduced a modified k-means algorithm for web log clustering. The original k-means algorithm consists of limitations of defining k clusters and initial centroid. The modified k-means algorithm determines the initial centroid. [1]

1.5 Organization of the Thesis

The structure of the book is described briefly for each chapter in this section. There are five major chapters and a brief introduction to each chapter is given as below.

Chapter 1 introduces the evolution of web technology over time and how it impacts in different industries. It also mentions how the traditional data analysis approaches are not able to fit to handle such a huge amount of data. This chapter also outlines the objectives of this thesis, what problems and issues are facing in UNHCR's web development and web usage mining comes to play as a critical role to overcome the challenges.

Chapter 2 describes web mining along with three types of categories such as web content mining, web structure mining and web usage mining. It explains web usage mining in detail including the process of pre-processing, pattern discovery and pattern analysis. Several major clustering approaches are presented, including partitioning methods, hierarchical methods, density-based methods, grid-based methods and model-based methods. Association rule mining section discusses finding patterns, associations and correlations among the data in the database. It also explains two important concepts, minimum support and minimum confidence and it introduces Apriori algorithm which is one of a well-known algorithms for finding frequent itemset and association rules based on Apriori theory.

Chapter 3 provides the architecture of the system. It outlines the process flow including data collection, data cleaning, and user identification, clustering and generating frequent user access patterns.

Chapter 4 presents how the system is implemented in detail. It demonstrates the data collection and cleaning, the data volume reduction using stratified random sampling technique, clustering the users using DBSCAN algorithm and generates frequent user access patterns with Apriori algorithm of association rule mining. It also

evaluates the performance factors of the system. It measures how clustering can effect on data reduction and the number of relevant and meaningful rules generated between clustered and non-clustered data.

Chapter 5 concludes providing the advantages and the efficiency of proposed approach over simple traditional approach. It mentions how the proposed approach can be improved by automatic selection of parameters for clustering. It also mentions that the system could be improved by using multi-threaded programming model when applying Apriori algorithm.

CHAPTER 2
THEORETICAL BACKGROUND

2.1 Web Mining

Web mining aims to discover useful information or knowledge from the web hyperlink structure, page content, and usage data. Although web mining uses many data mining techniques, it is not purely an application of traditional data mining techniques due to the heterogeneity and semi-structured or unstructured nature of the web data. Based on the primary kinds of data used in the mining process, web mining tasks can be categorized into three types: web structure mining, web content mining and web usage mining. [4]

2.1.1 Web Usage Mining

Web usage mining refers to the automatic discovery and analysis of patterns in clickstreams, user transactions and other associated data collected or generated as a result of user interactions with web resources on one or more websites. The goal is to capture, model, and analyze the behavioral patterns and profiles of users interacting with a website. The discovered patterns are usually represented as collections of pages, objects, or resources that are frequently accessed or used by groups of users with common needs or interests. Web usage mining process can be divided into three inter-dependent stages: data collection and pre-processing, pattern discovery, and pattern analysis. In the pre-processing stage, the clickstream data is cleaned and partitioned into a set of user transactions representing the activities of each user during different visits to the site. In the pattern discovery stage, statistical, database, and machine learning operations are performed to obtain hidden patterns reflecting the typical behavior of users, as well as summary statistics on web resources, sessions, and users. In the final stage of the process, the discovered patterns and statistics are further processed, filtered, possibly resulting in aggregate user models that can be used as input to applications such as recommendation engines, visualization tools, and web analytics and report generation tools. [4]

2.1.2 Web Structure Mining

Web structure mining discovers useful information based on hyperlinks structure of the web site. It explores topology of hyperlinks of the web site. The structure of the World Wide Web can be viewed as graph where web pages are nodes, and hyperlinks are edges connecting related pages. Then, web structure mining is the process of discovering structure information from the web. Web structure mining can be divided to two main parts based on the kind of structure information used: Hyperlinks and Document structure. A Hyperlink is a structural unit which connects a location in web page to either different place on the same web page or to different web page. The Intra -Document Hyperlink is the hyperlink pointing to place within the same web page. On other hand, Inter-Document Hyperlink connects two different web pages. Document structure is a web page organized in a tree-structured format, based on the various HTML and XML tags within the page. The main effort is focused on automatically extracting document object model (DOM) structures out of documents. [6]

2.1.3 Web Content Mining

Web content mining is the process where useful information is extracted from the contents of web documents. Content data correspond to the collection of facts a web page was designed to pass on to the users. Data on the web page can be in the form of text, video, pictures and audio. The multimedia mining is working with all forms of data such as video, pictures, audio and text. The web content data are in the form of unstructured data such as free texts, semi-structured data such as HTML and XML documents, and a more structured data such as data in the tables or database generated HTML pages. Much of the data on the Web are in unstructured form. Web content mining can be divided in two sections based on the point of the view: Information Retrieval and Database views.

Information Retrieval view deals with unstructured and semi-structured documents. The unstructured documents are free texts such as news stories. There are mainly three main types of the unstructured documents pre-processing: the bag of words or vector representation, Latent Semantic Indexing (LSI) and using information about word position in the document. The bag of words or vector representation uses

single words find in training corpus as feature which can be Boolean (occurs or not) or frequency based (number of occurrences in document). LSI is an indexing and retrieval method that uses a mathematical technique called singular value decomposition to identify patterns in the relationships between the terms and concepts contained in an unstructured collection of text. LSI is based on the principle that words that are used in the same contexts tend to have similar meanings. A key feature of LSI is its ability to extract the conceptual content of a body of text by establishing associations between those terms that occur in similar contexts. The information about word position in the document is using n-grams representation (word sequences of length up to n). The semi-structured documents have as addition structure (HTML and hyperlink) when comparing to unstructured documents.

Database view on web content mining is focused on techniques for organizing the semi-structured data on the web into more structured collections of resources and using standard database querying mechanisms and data mining techniques to analyze it. There are two different approaches: Multilevel Databases and Web Query Systems. Multilevel Databases approach uses idea that the lowest level of the database contains semi-structured information such as hypertext documents stored in various web repositories. The Meta data or generalizations are extracted from lower levels to the higher level(s) and organized in structured collections, i.e., relational or object-oriented databases. Web Query systems approach is using fact that many web-based query systems and languages use standard database query languages such as SQL, structural information about web documents, and even natural language processing for the queries that are used in World Wide Web searches. [6]

2.2 Web Usage Mining

Web Usage Mining is the application of data mining techniques to discover interesting usage patterns from web data, in order to understand and better serve the needs of web-based applications. Web servers collect large amounts of data from the web sites' usage. These data are stored in web access log files. Together with the web access log files, other data can be used in Web Usage Mining like the web structure information, user profiles, refers logs (contain information about the referring pages for each page reference), etc. This data analysis can be used by organization or e-commerce for cross-marketing strategies across the products, effectiveness of promotions and

other things. The overall web usage mining process can be divided into three inter-dependent stages: pre-processing, pattern discovery, and pattern analysis. The overall process is depicted in the Figure 2.1.

Pre-processing is the gathering of data and their transformation to format to which mining algorithms can be applied. It is the most important stage of usage mining because data are usually collected from multiple resources and across different channels. Pre-processing of collected data is challenging because of time consuming and intensive use of computation power. Usage data preparation presents a number of unique challenges which are leading to a variety of algorithms and heuristic techniques for pre-processing tasks such as data cleaning, user and session identification, pageview identification. Data cleaning involves removal of uninteresting data and references of crawler navigations. User identification deals with identification of individual users with help of client-side cookies, a combination of IP addresses or other information such as user agents and referrers. Session identification is the process of separating the user activity record of each user into sessions, each representing a single visit to the site. The pageview is a collection of web objects or resources representing a specific user event such as clicking on a link or viewing a product page. Identification of pageviews is dependent on the intra-page structure of the site, page contents and the underlying site domain knowledge.

Pattern Discovery mines knowledge from the datasets which are result of pre-processed raw logs. The data mining techniques to accomplish this are mainly association rule mining, sequential pattern mining and clustering. The association rule mining is based on identification of strong rules discovered in databases using different measures of interestingness. Sequential pattern mining is similar to association rule mining with addition of time element (order of events, i.e. clicks). Clustering is the division of data into groups of similar objects.

Pattern Analysis is used to understand, visualize and interpret the patterns which are results of patterns discover. Web usage mining can be also divided into three main categories based on origin of the data: Web Server data, Application Server Data and Application Level data. Web Server data are data collected from Web Server logs such as IP addresses, page reference and access time. Application Server Data are data coming from commercial application servers and are used to track various kinds of

business events. Application Level data are data which are resulting from events specially define in an application. [6]

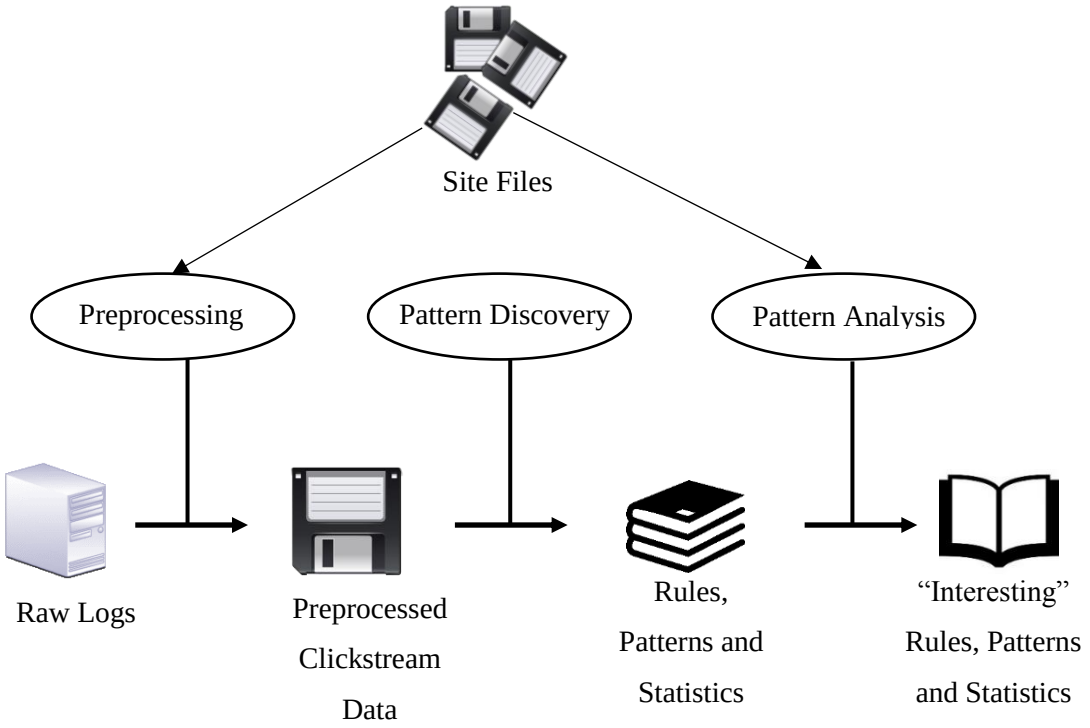


Figure 2.1 Web Usage Mining

2.2.1 Data Modelling for Web Usage Mining

Usage data pre-processing results in a set of n pageviews, P = {p_1, p_2, ⋯, p_n}, and a set of m user transactions, T = {t_1,t_2,⋯,t_m}, where each t_i in T is a subset of P. Pageviews are semantically meaningful entities to which mining tasks are applied (such as pages or products). Conceptually, each transaction t can be viewed as an l-length sequence of ordered pairs as shown in equation 2.1.

$$t = (p_1^t, w(p_1^t)), (p_2^t, w(p_2^t)), \dots, (p_l^t, w(p_l^t)) \quad (2.1)$$

Where each $p^t_i = p_j$ for some j in {1, 2, ⋯, n}, and $w(p^t_i)$ is the weight associated with pageview p^t_i in transaction t, representing its significance. The weights can be determined in a number of ways, in part based on the type of analysis or the intended personalization framework. In most web usage mining tasks the weights are either binary, representing the existence or non-existence of a pageview in the transaction; or

they can be a function of the duration of the pageview in the user's session. In the case of time durations, it should be noted that usually the time spent by a user on the last pageview in the session is not available. One commonly used option is to set the weight for the last pageview to be the mean time duration for the page taken across all sessions in which the pageview does not occur as the last one. In practice, it is common to use a normalized value of page duration instead of raw time duration in order to account for user variances. In some applications, the log of pageview duration is used as the weight to reduce the noise in the data. [4]

2.2.2 Data Collection and Pre-processing

An important task in any data mining application is the creation of a suitable target data set to which data mining and statistical algorithms can be applied. This is particularly important in web usage mining due to the characteristics of clickstream data and its relationship to other related data collected from multiple sources and across multiple channels. The data preparation process is often the most time consuming and computationally intensive step in the web usage mining process, and often requires the use of special algorithms and heuristics not commonly employed in other domains [3]. This process is critical to the successful extraction of useful patterns from the data. The process may involve pre-processing the original data, integrating data from multiple sources, and transforming the integrated data into a form suitable for input into specific data mining operations. Collectively, this process is referred to as data preparation. The successful application of data mining techniques to web usage data is highly dependent on the correct application of the pre-processing tasks.

2.3 Clustering

Clustering often called unsupervised learning is the process of organizing data instances into groups whose members are similar in some way. A cluster is therefore a collection of data instances which are "similar" to each other and are "dissimilar" to data instances in other clusters. In the clustering, a data instance is also called an object as the instance may represent an object in the real world. It is also called a data point as it can be seen as a point in a dimensional space. Clustering is also called data

segmentation in some applications because clustering partitions large data sets into groups according to their similarity.

2.3.1 Different Major Clustering Methods

Many different clustering algorithms exist in the literature. Generally, they fall under one of the below major clustering methods. [10]

(i) Hierarchical Methods

These methods construct the clusters by recursively partitioning the instances in either a top-down or bottom-up fashion. These methods can be subdivided as following:

(a) Agglomerative hierarchical clustering: Each object initially represents a cluster of its own. Then clusters are successively merged until the desired cluster structure is obtained.

(b) Divisive hierarchical clustering**:** All objects initially belong to one cluster. Then the cluster is divided into sub-clusters, which are successively divided into their own sub-clusters. This process continues until the desired cluster structure is obtained.

(ii) Partitioning Methods

Partitioning methods relocate instances by moving them from one cluster to another, starting from an initial partitioning. Such methods typically require that the number of clusters will be pre-set by the user. To achieve global optimality in partitioned-based clustering, an exhaustive enumeration process of all possible partitions is required. Because this is not feasible, certain greedy heuristics are used in the form of iterative optimization. Namely, a relocation method iteratively relocates points between the k clusters. The following subsections present various types of partitioning methods.

(a) Error minimization algorithms: These algorithms, which tend to work well with isolated and compact clusters, are the most intuitive and frequently used methods. The basic idea is to find a clustering structure that minimizes a certain

error criterion which measures the "distance" of each instance to its representative value. The most well-known criterion is the Sum of Squared Error (SSE), which measures the total squared Euclidian distance of instances to their representative values.

(b) Graph-theoretic clustering: Graph theoretic methods are methods that produce clusters via graphs. The edges of the graph connect the instances represented as nodes. A well-known graph-theoretic algorithm is based on the Minimal Spanning Tree.

(iii) Density-based Methods

Density-based methods assume that the points that belong to each cluster are drawn from a specific probability. The overall distribution of the data is assumed to be a mixture of several distributions. The aim of these methods is to identify the clusters and their distribution parameters.

(a) DBSCAN algorithm (density-based spatial clustering of applications with noise) discovers clusters of arbitrary shapes and is efficient for large spatial databases. The algorithm searches for clusters by searching the neighborhood of each object in the database and checks if it contains more than the minimum number of objects.

(iv) Model-based Methods

These methods attempt to optimize the fit between the given data and some mathematical models. Unlike conventional clustering, which identifies groups of objects, model-based clustering methods also find characteristic descriptions for each group, where each group represents a concept or class. The most frequently used induction methods are decision trees and neural networks.

(a) Decision Trees: In decision trees, the data is represented by a hierarchical tree, where each leaf refers to a concept and contains a probabilistic description of that concept. The most well-known algorithms are COBWEB and CLASSIT.

(b) Neural networks. This type of algorithm represents each cluster by a neuron or "prototype". The input data is also represented by neurons, which are connected to the prototype neurons. Each such connection has a weight, which is learned

adaptively during learning. A very popular neural algorithm for clustering is the self-organizing map (SOM).

(v) Grid-based Methods

These methods partition the space into a finite number of cells that form a grid structure on which all of the operations for clustering are performed. The main advantage of the approach is its fast processing time.

2.4 Distance Functions

Distance or similarity functions play a central role in all clustering algorithms. Numerous distance functions have been reported in the literature and used in applications. Different distance functions are also used for different types of attributes (also called variables). [4]

2.4.1 Numeric Attributes

The most commonly used distance functions for numeric attributes are the Euclidean distance and Manhattan (city block) distance. Both distance measures are special cases of a more general distance function called the Minkowski distance.

2.4.2 Binary and Nominal Attributes

For binary and nominal attributes (also called unordered categorical attributes), different functions should be used. A binary attribute has two states or values, usually represented by 1 and 0. The two states have no numerical ordering. For example, Gender has two values, male and female, which have no ordering relations but are just different. Existing distance functions for binary attributes are based on the proportion of value matches in two data points. A match means that, for a particular attribute, both data points have the same value. Given the i^{th} and j^{th} data points, x_i and x_j, the confusion matrix shown in Figure 2.2 can be constructed. [4]

		Data point X_j		
		1	0	
Data point X_i	1	a	b	a+b
	0	c	d	c+d
		a+c	b+d	a+b+c+d

Where,

a: the number of attributes with the value of 1 for both data points.

b: the number of attributes for which x_{ij}=1 and x_{ij}=0, where x_{ij}(x_{ij}) is the value of the j^{th} attribute of the data point x_i(x_j).

c: the number of attributes for which x_{ij}=0 and x_{ij}=1.

d: the number of attributes with the value of 0 for both data points.

Figure 2.2 Confusion Matrix for Data Points X_i, X_j

To give the distance functions, binary attributes can further into symmetric and asymmetric attributes. Symmetric attribute is a binary attribute if both of its states (0 and 1) have equal importance and carry the same weight. Asymmetric attribute is also a binary attribute if one of the states is more important or valuable than the other. The most commonly used distance measure for asymmetric attributes is the Jaccard distance in equation 2.2. [4]

$$dist(x_i, x_j) = \frac{b+c}{a+b+c} \tag{2.2}$$

Alternatively, Jaccard similarity can be computed as in equation 2.3.

$$sim(x_i, x_j) = \frac{a}{a+b+c} \tag{2.3}$$

For general nominal attributes with more than two states or values, the commonly used distance measure is also based on the simple matching distance.

2.4.3 Text Documents

Although a text document consists of a sequence of sentences and each sentence consists of a sequence of words, a document is usually considered as a “bag” of words in document clustering. The sequence and the position information of words are ignored. Thus, a document can be represented as a vector just like a normal data point. However, similarity is used to compare two documents rather than distance. The most commonly used similarity function is the cosine similarity.

2.5 Density-based Spatial Clustering Applications with Noise (DBSCAN)

Density-based spatial clustering applications with noise is a density-based clustering algorithm. The algorithm grows regions with sufficiently high density into clusters and discovers clusters of arbitrary shape in spatial databases with noise. It defines a cluster as a maximal set of density-connected points.

The basic ideas of density-based clustering involve a number of new definitions as below. [7]

(i) The neighborhood within a radius ε of a given object is called the ε-neighborhood of the object.

(ii) If the ε-neighborhood of an object contains at least a minimum number, MinPts, of objects, then the object is called a core object.

(iii) If the ε-neighborhood of an object does not contain a minimum number, MinPts, of objects but the object is neighborhood of a core object then the object is called a border object.

(iv) If the ε-neighborhood of an object does not contain a minimum number, MinPts, of objects, or the object is not neighborhood of a core object then the object is considered as an outlier (noise).

(v) Given a set of objects, D, an object p is directly density-reachable from object q if p is within the ε-neighborhood of q, and q is a core object.

(vi) An object p is density-reachable from object q with respect to ε and MinPts in a set of objects, D, if there is a chain of objects $p_1,\ldots, p_n$, where $p_1 = q$ and $p_n = p$ such that p_i+1 is directly density-reachable from p_i with respect to ε and MinPts, for $1 \leq i \leq n$, $p_i \in D$.

(vii) An object p is density-connected to object q with respect to ε and MinPts in a set of objects, D, if there is an object $o \in D$ such that both p and q are density-reachable from o with respect to ε and MinPts.

Density reachability is the transitive closure of direct density reachability, and this relationship is asymmetric. Only core objects are mutually density reachable. Density connectivity, however, is a symmetric relation. The Figure 2.3 shows the core

point (A), border point (B) and noise point (C) of DBSCAN algorithm using a given radius (Eps) and MinPts =5.

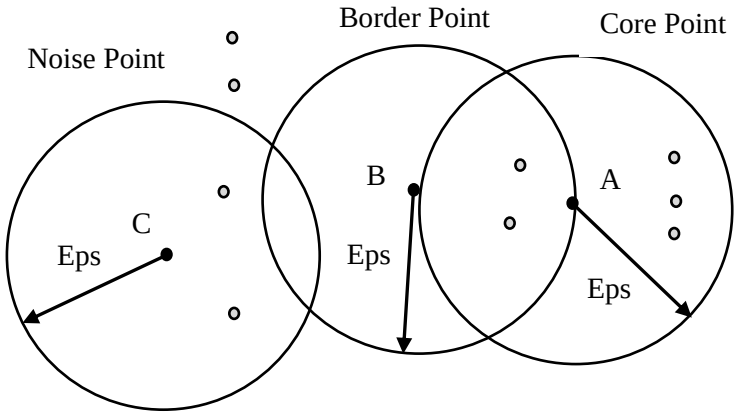


Figure 2.3 DBSCAN with Core (A), Border (B) and Noise (C) Points

2.6 Association Rule Mining

Association rule mining is the process of finding patterns, associations and correlations among sets of items in a database. The generated association rules have an antecedent and a consequent. An association rule is a pattern of the form X & Y $\Rightarrow$ Z [support, confidence], where X, Y, and Z are items in the dataset. The left-hand side of the rule X & Y is called the antecedent of the rule and the right hand side Z is called the consequent of the rule. This means that given X and Y there is some association with Z. Within the dataset, confidence and support are two measures to determine the certainty or usefulness for each rule. Support is the probability that a set of items in the dataset contains both the antecedent and consequent of the rule or P ($X \cup Y \cup Z$). Confidence is the probability that a set of items containing the antecedent also contains the consequent or P ($Z| X \cup Y$). Typically, an association rule is called strong if it satisfies both a minimum support threshold and a minimum confidence threshold that is determined by the user [4]. The below are some well-known association rule mining algorithms.

2.6.1 Apriori Algorithm

The Apriori algorithm is a well-known algorithm for finding frequent itemsets from a set of data by using candidate generation. Apriori uses an iterative approach

known as a level-wise search because the *k*-itemsets is used to determine the (*k* + 1)-itemsets. The search begins for the set of frequent 1-itemsets denoted L_1. L_1 is then used to find the set of frequent 2-iemsets, L_2. L_2 is then used to find L_3 and so on. This continues until no more frequent *k*-itemsets can be found.

To improve efficiency of a level-wise generation, the Apriori algorithm uses the Apriori property. The Apriori property states that all nonempty subsets of a frequent itemset are also a frequent itemsets. So, if {A, B} is a frequent itemset then subsets {A} and {B} are also frequent itemsets. The level-wise search uses this Apriori property when stepping from level to the next. If an itemset *I* does not satisfy the minimal support, then *I* will not be considered a frequent itemset. If item *A* is added to the itemset *I* then the new itemset $I \cup A$ cannot occur more frequently than the original itemset *I*. If an itemset fails to be considered a frequent itemset then all supersets of that itemset will also fail that same test. The Apriori algorithm uses this property to decrease the number of itemsets in the candidate list therefore optimizing search time. As the Apriori algorithm steps from finding L_{k-1} to finding L_k it uses a two-step process consisting of the Join Step and the Prune Step.

The first step is the Join Step and it is responsible for generating a set of candidates *k*-itemsets denoted C_k from L_{k-1}. It does this by joining L_{k-1} with itself. Apriori assumes that items within a transaction or itemset are sorted in lexicographic order. The joining L_{k-1} to L_{k-1} is only performed between itemsets that have the first (k-2) items in common with each other. Suppose itemsets I_1 and I_2 are members of L_{k-1}. They will be joined with each other if ($I_1[1] = I_2[1]$ and $I_1[2] = I_2[2]$ and … and $I_1[k-2] = I_2[k-2]$ and $I_1[k-1] < I_2[k-1]$). Where $I_1[1]$ is the first item in itemset I_1 and $I_1[k-1]$ is the last item in I_1 and so on for I_2. It checks to make sure all *k*-2 items are equal and then lastly makes sure the last item in the itemset is unequal in order to eliminate duplicate candidate *k*-itemsets. The new candidate *k*-itemset generated from joining I_1 with I_2 would be $I_1[1]\ I_1[2]\ldots\ I_1[k-1]I_2[k-1]$.

The second step is the Prune Step and converts C_k to L_k. The candidate list C_k contains all of the frequent *k*-itemsets but it also contains k-itemsets that do not satisfy the minimum support count. The scan of the database determines the occurrence frequency of every candidate k-itemset to determine if it satisfies the minimum support. [8]

2.6.2 Eclat Algorithm

Eclat algorithm, proposed by ZAKI in 2000, is based on the breadth-first search strategy, which adopts the technologies of vertical data format, lattice theory, equivalence classes, intersection and so on. The main steps of Eclat are listed as follows: scan the database to get all frequent 1-itemsets, generate candidate 2-itemsets from frequent 1-itemsets, then get all frequent 2-itemsets by clipping non-frequent candidate itemsets; generate candidate 3-itemsets from frequent 2-itemsets and then get all frequent 3-itemsets by clipping non-frequent candidate itemsets; repeat the above steps, until no candidate itemset can be generated. Same as Apriori, Eclat algorithm also adopts the join operation to generate candidate (K+1)-itemset by taking the union of two k-itemset. The condition of two kitemset can be joined is that the front k-1 items of the two k-itemset must be the same. For example, there are two 3-itemset: l_{31}={I1,I2,I3} and l_{32}={I1,I2,I4}, the first and second items of l_{31} and l_{32} are the same, so l_{31} and l_{32} can be joined to generate a 4-itemset: l_4= l_{31} Join l_{32}={I1,I2,I3,I4}. By using the concept of equivalence classes, Eclat divides the search space into multiple non-overlapping sub spaces. The itemsets which have same prefix can be classified into a same class, and the generation of candidate itemsets can be only operated in a same class. The technology of equivalence classes can obviously improve the efficiency of generating candidate itemset and can reduce the occupation of memory. [11]

2.6.3 FP-growth Algorithm

Frequent pattern growth, or simply FP-growth, which adopts a divide-and-conquer strategy as follows. First, it compresses the database representing frequent items into a frequent pattern tree, or FP-tree, which retains the itemset association information. It then divides the compressed database into a set of conditional databases (a special kind of projected database), each associated with one frequent item or “pattern fragment,” and mines each database separately. For each “pattern fragment,” only its associated data sets need to be examined. Therefore, this approach may substantially reduce the size of the data sets to be searched, along with the “growth” of patterns being examined. [4]

CHAPTER 3
ARCHITECTURE OF THE SYSTEM

3.1 Overview of the Proposed System

This system intends to extract relevant and meaningful frequent user access patterns with the combination of clustering and association rule mining from Operational Data Portal of UNHCR. DBSCAN has been selected from the different clustering techniques and Apriori algorithm is used to discover frequent user access patterns. The architecture of the system can be simplified shown in the Figure 3.1. The system has four main steps, collection of web log data, data pre-processing, clustering users who share the similar access patterns and generate the frequent user access patterns from each user cluster.

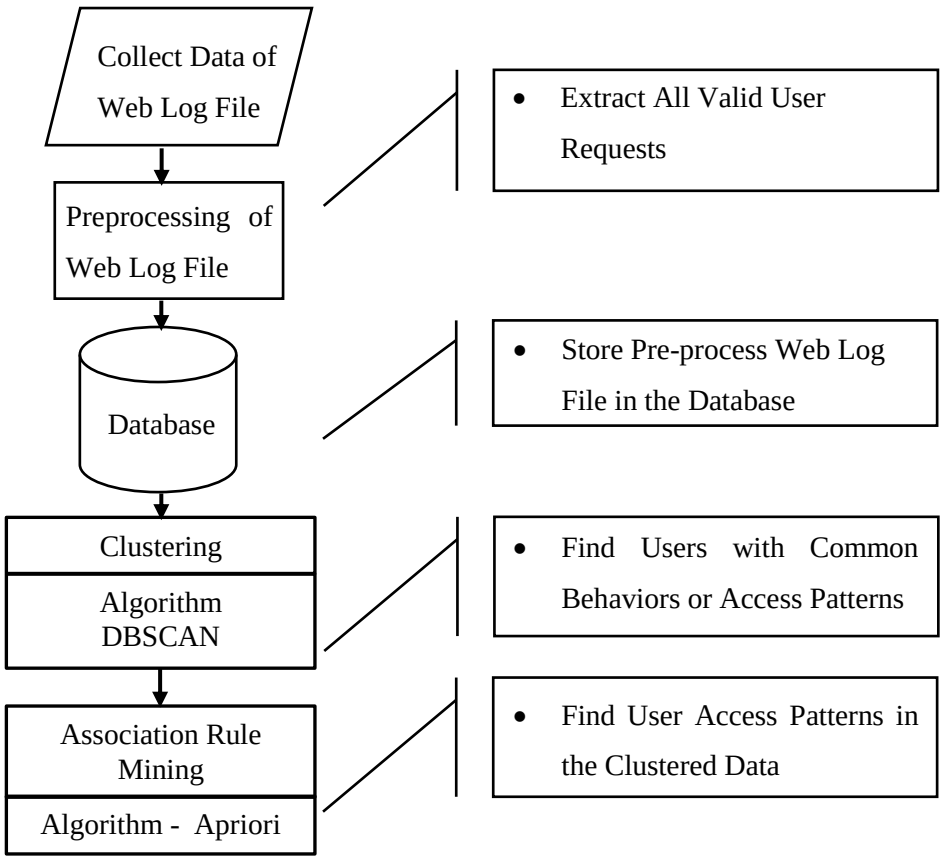


Figure 3.1 Flow of the System

3.2 Collection of Raw Web Log Files

As indicated in the architecture of the system, the first step is to collect the raw web log data. United Nations High Commissioner for Refugees (UNHCR) has kindly contributed 30 web log files from operational data portal. The operational data portal website is served by an Nginx web server and the standard web log format saves the following information: the IP of the computer that requested information, date/time at which the transaction was completed, time taken for transaction completion, bytes transferred, the web resources accessed, HTTP status code, the referrer and web browser information. The Figure 3.2 is the sample web log information from data.unhcr.org. As the website is public to all users, user identity and authenticated user name fields are not necessary to access the data. Thus the fields are empty.

1. 51.255.65.65 - - [07/Apr/2016:06:34:26 +0200] "GET /wiki/index.php?title=Special:RecentChangesLinked&feed=atom&days=30&from=20151018174520&hidebots=0&hideanons=1&hidemyself=1&target=File%3AModule.png HTTP/1.1" 301 686 "-" "Mozilla/5.0 (compatible; AhrefsBot/5.1; +http://ahrefs.com/robot/)"
2. 52.4.48.181 - - [07/Apr/2016:06:34:27 +0200] "GET /robots.txt HTTP/1.1" 404 2434 "-" "Mozilla/5.0 (compatible; alexa site audit/1.0; +http://www.alexa.com/help/webmasters;)"
3. 151.80.31.168 - - [07/Apr/2016:06:34:27 +0200] "GET /wiki/index.php?title=Special:RecentChangesLinked&feed=atom&days=30&from=20151009145721&hidebots=0&hideliu=1&hidemyself=1&target=File%3AHighlight.png HTTP/1.1" 301 690 "-" "Mozilla/5.0 (compatible; AhrefsBot/5.1; +http://ahrefs.com/robot/)"

Figure 3.2 Examples of Web Server Logs from data.unhcr.org

3.3 Pre-processing of Raw Web Log Data

Once raw web log data has been collected, the data needs to be cleaned. Web server records all transaction between web server and client regardless whether the

request explicitly requested or not. The system can easily detect those requests which are not requested by users. The uniform resource locators (URLs) ending with css, js, jpeg are not explicitly requested by users and these records should be removed. And also, those log entries with are not successfully delivered to users should be removed (for example, those records with HTTP status code other than 200). If the log entries are valid, then URL is cleaned removing unnecessary query parameters. Table 3.2 shows a web log entry ending with PNG extension, a web log entry with 301 HTTP status code and a valid URL. For a valid URL, a clean URL is also presented.

Table 3.1 Valid and Invalid Web Log Entries

URL	HTTP Status Code	Valid	Clean URL
GET /syrianrefugees/images/syria/factfigures_panel.png HTTP/1.1	200	NO	
GET /mediterranean/regional.html HTTP/1.1	301	NO	
GET /mediterranean/regional.php HTTP/1.1 200	200	YES	/mediterranean/regional.php

3.4 Identifying Distinct Users

Once the data is cleaned, it is ready to do user identification. It simply means that the system estimates which records are accessed by which users based on the available variables in the web log. There are nine variables in the web log data, 1) IP address, 2) user identity, 3) authenticated user name, 4) requested date and time 5) requested URL, 6) HTTP status code, 7) response size, 8) referrer of URL and 9) Useragent. The system uses two variables IP address and useragent to estimate user. If the records have the same IP address and useragent, it is estimated that it is the same user but if IP address and useragent are not the same, then it is different user. Table 3.3 shows 3 web log entries and they have different IP address. Thus, it is assumed that they are three different users and assigned unique identifier (1, 2, and 3).

After identifying users for all web log entries, the data will be stored in SQLite database for subsequent processes. Out of nine variables in web log data, user identify, and authenticated user name fields are empty, and they are not saved into the database. Table 3.4 displays table schema in which web log data is saved.

Table 3.2 Clean Log Data after User Identification

User ID	Client IP Address	Date	Requested URL	HTTP Status Code	Response Size	Referrer of URL	User Agent
1	66.249.78.176	4/9/2016 6:40:17	/horn-of-africa/cal_region.php	200	7299	-	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)
2	151.80.226.31	4/9/2016 6:40:19	/syrianrefugees/download.php	200	1505392	-	Mozilla/5.0 (Windows NT 6.0; WOW64; rv:24.0) Gecko/20100101 Firefox/24.0
3	54.161.157.120	4/9/2016 6:40:21	/syrianrefugees/rss/country.php	200	58642	-	FactrBot v1.0

Table 3.3 Table Schema for Storing Web Log Data

Sr.	Column Name	Data Type	Primary Key	Allow Null	Description
	ID	INTEGER	YES	NO	Auto number which can uniquely identify each row
2	Remote	TEXT	NO	NO	Client IP address
3	user	INTEGER	NO	NO	User ID which uniquely identify each user
4	date	DATETIME	NO	NO	Requested date
5	request	TEXT	NO	NO	URL requested by users
6	request_i	INTEGER	NO	NO	URL Numerical ID
7	status	INTEGER	NO	YES	HTTP status code
8	size	INTEGER	NO	YES	Response size
9	referrer	TEXT	NO	YES	URL's referrer
10	useragent	TEXT	NO	NO	Web browser and operation system information

3.5 Clustering Based on Users

After user identification, only those users who share similar access patterns feed into clustering process. The users who do not have similar access patterns with other users, they will be labelled as outlier and will no longer be part of clustering. To run DBSCAN, two parameters are needed, epsilon (the radius of cluster) and minimum of points to form a cluster. Selection of these parameters are solely depending on the underlying web log data. So, having knowledge of website structure greatly help in choosing the parameters for DBSCAN. Once the parameters are identified, DBSCAN process starts selecting one random user. Using epsilon and minimum number of points, DBSCAN scan whether the area is dense enough or not. If there are users including itself more than or equal to minimum number of points, then a cluster is formed. If the number of users is less then minimum number of points, then that user will be initially labelled as noise (outlier) but there is still chance to become a border point if it falls within the range of another user. In this way, DBSCAN process all users until there is no more users visited. At the end of DBSCAN, each user belongs to one cluster ID. The users with cluster ID (0) mean that they do not belong to any cluster and they are merely noise. Those users with cluster ID greater than 0 mean they are owned by proper clusters. Thus, the users in each cluster shares similar access patterns. Table 3.5 shows sample two sample clusters with ten users after DBSCAN process.

Table 3.4 Two Sample Clusters with Ten Users after DBSCAN

User ID	Cluster ID	Note
1	0	Noise
2	1	Cluster 1
3	1	Cluster 1
4	1	Cluster 1
5	1	Cluster 1
6	1	Cluster 1
7	2	Cluster 2
8	2	Cluster 2
9	2	Cluster 2
10	2	Cluster 2

3.6 Rule Generation

Once the users are clustered based on their similarity, it is ready to generate association rules. To be able to generate association rules, users must provide two parameters, minimum support and minimum confidence. For the purpose of generating user frequent access patterns, Apriori algorithm is used. The algorithm is applied to each and everyone of cluster sequentially. Before running Apriori, web data must be modelled in a proper format. When modelling the data, users can be considered as transactions and those web log entries accessed by that user become itemset. In the first step, Apriori generates all frequent URLs which satisfy minimum support threshold. Frequent user access patterns are generated from frequent URLs that are generated in the first step of Apriori algorithm. Sample frequent user access patterns are displayed in Figure 3.2 with 80% minimum support and minimum confidence.

```
(' /syrianrefugees/download.php', '--->', ' /syrianrefugees/settlement.php', 'support:', '1.00', 'conf:', '1.00')
(' /syrianrefugees/settlement.php', '--->', ' /syrianrefugees/download.php', 'support:', '1.00', 'conf:', '1.00')
(' /syrianrefugees/settlement.php', '--->', ' /syrianrefugees/partner.php', 'support:', '0.80', 'conf:', '0.80')
(' /syrianrefugees/partner.php', '--->', ' /syrianrefugees/settlement.php', 'support:', '0.80', 'conf:', '1.00')
(' /syrianrefugees/download.php', '--->', ' /syrianrefugees/partner.php', 'support:', '0.80', 'conf:', '0.80')
(' /syrianrefugees/partner.php', '--->', ' /syrianrefugees/download.php', 'support:', '0.80', 'conf:', '1.00')
(' /syrianrefugees/download.php', '--->', ' /syrianrefugees/partner.php, /syrianrefugees/settlement.php', 'support:', '0.80', 'conf:', '0.80')
```

Figure 3.3 Sample Frequent User Access Patterns from data.unhcr.org

CHAPTER 4
IMPLEMENTATION OF THE SYSTEM

4.1 Data Collection for Web Logs

Collection of web logs data is the first step of pattern discovery process. In this thesis, the web logs data from data.unhcr.org that is hosted by United Nations High Commissioner for Refugees (UNHCR) are used for 34 days from 8 March 2016 to 10 April 2016.

To give some backgrounds on the website that is being used for this thesis, Operational Data Portal is the information sharing and inter-agency platform which is used for coordination and information sharing among United Nations agencies, International Non-governmental organizations and Governments. The website is designed for each humanitarian crisis that may include one or more countries. For example Syrian crisis mainly effects Egypt, Iraq, Turkey, Lebanon and Jordan. So, Syria situation web page covers Turkey, Lebanon, Jordan, Iraq and Egypt.

After importing the web logs, the raw data indicates that there are more than 30.8 million log entries for 34 days. This is huge amount of data and it shows that there are many users who are using this site. The number of raw log entries per day is shown in the Table 4.1.

Table 4.1 Number of Log Entries for 34 days

Sr.	Date	# of Raw Log Entries
1	08-Mar-2016	1,071,150
2	09-Mar-2016	1,213,421
3	10-Mar-2016	1,236,052
4	11-Mar-2016	161,704
5	12-Mar-2016	559,064
6	13-Mar-2016	820,902
7	14-Mar-2016	1,287,759
8	15-Mar-2016	1,280,343
9	16-Mar-2016	1,336,700
10	17-Mar-2016	1,047,589
11	18-Mar-2016	964,972

Sr.	Date	# of Raw Log Entries
12	19-Mar-2016	691,041
13	20-Mar-2016	890,978
14	21-Mar-2016	1,229,216
15	22-Mar-2016	1,198,880
16	23-Mar-2016	1,163,181
17	24-Mar-2016	981,603
18	25-Mar-2016	727,050
19	26-Mar-2016	562,884
20	27-Mar-2016	746,252
21	28-Mar-2016	986,043
22	29-Mar-2016	1,296,967
23	30-Mar-2016	1,247,738
24	31-Mar-2016	1,234,904
25	01-Apr-2016	207,006
26	02-Apr-2016	591,081
27	03-Apr-2016	126,444
28	04-Apr-2016	964,521
29	05-Apr-2016	1,123,754
30	06-Apr-2016	1,169,561
31	07-Apr-2016	1,083,526
32	08-Apr-2016	868,953
33	09-Apr-2016	603,833
34	10-Apr-2016	123,568
Total		30,798,640

4.2 Data Pre-processing

The purpose of data pre-processing is to transform raw log data into clean data. Web server records all transactions which includes the records that users don't request explicitly. For instance, clicking a hyperlink on a website may involve transferring many HTTP messages between web server and the client. So, not all transactions in the web server are interesting for pattern discovery process but only those records which are explicitly requested by users and successfully received by the client. As data.unhcr.org is public information sharing platform, all web request are anonymous

without any credentials information. So, the web log entries need to be identified that which users are accessing them using some available variables. The next two sections explain how the data is cleaned and segmented.

4.2.1 Data Cleaning

The objective of data cleaning process is to remove unnecessary entries from the raw log records. If a URL is ending with gif, jpeg, css and javascript file, these records are only a part of web page and should be removed. And again if the transaction is not successful, this implies that web server doesn't deliver the web page to the client successfully. These unsuccessful entries should be removed as well. The algorithm for data cleaning process is shown in the Figure 4.1.

Algorithm Name: Data Cleansing of Web Log File

Input: Web Server Log file
Output: Clean Web Log Data
Step1: Read Log Record from Web Log File
Step2: IF (Log Record .url contains (gif, jpeg, jpg, css))
OR (Different error like HTTP 404 or more) found THEN
Remove from web log file.
ELSE
Clean URL.
END IF
Step3: Repeat the above two steps until
EOF (Web Log File)

Step4: Stop the process.

Figure 4.1 Algorithm for Data Cleaning

After removing unnecessary log entries, the remaining log entries should be cleaned by removing query parameters for further process. Table 4.2 shows the original versus clean web log.

Table 4.2 Original vs. Clean Web Log

Original Web Log	Clean Web Log
157.55.39.244 - - [09/Apr/2016:06:39:52 +0200] "GET /burundi/partner.php ?OrgId=118 HTTP/1.1" 200 6890 "-" "Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)“	157.55.39.244 - - [09/Apr/2016:06:39:52 +0200] "GET /burundi/partner.php" 200 6890 "-" "Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)"

After cleaning web log data, the number transaction is dramatically decreased from 30.8 million to 3.4 million. It was noted that the website usage is going down during the weekend, but it is high in weekdays. The web usage trends for 34 days after data cleaning is shown in the Figure 4.2.

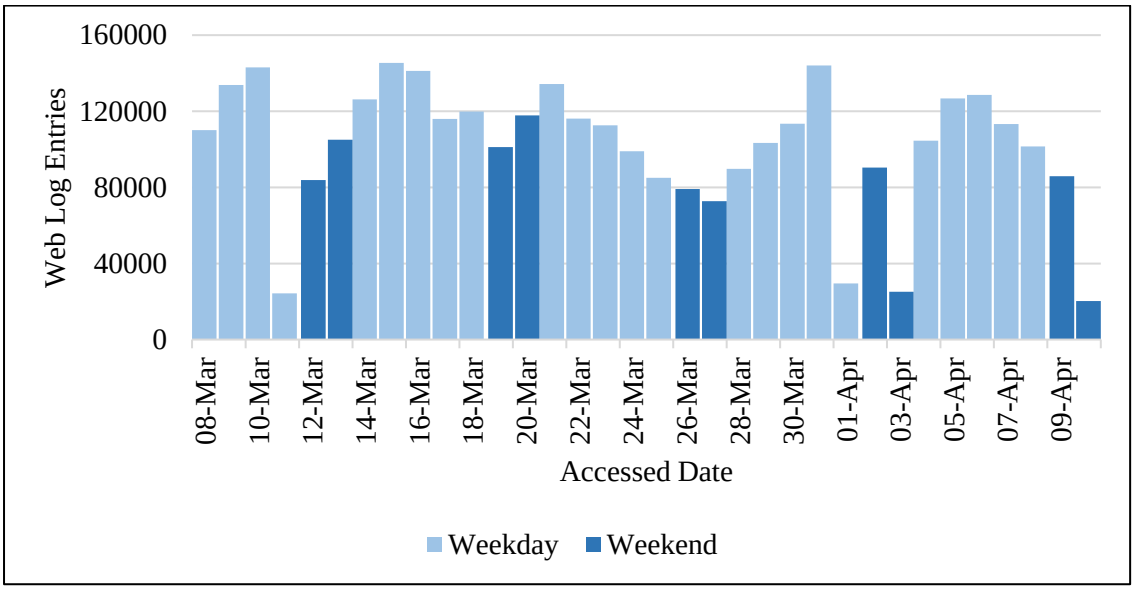


Figure 4.2 Web Usage Trends of data.unhcr.org from 8-Mar-16 To 10-Apr-16

4.2.2 User Identification

After cleaning the data, the user should be identified for each transaction. There are two variables that can be used for the best estimation of user identity, those are IP address and users' underlying operating system and browser. IP address comes from the internet service provider where the client machine is located. The web log entries include the combination of the operation system and web browser used by the user and

it is known as useragent. If two requests have the same IP address and useragent, then it is estimated that those requests are used by the same user. If not, these are two different users. After user identification process, all the log entries are associated with the identified users. Figure 4.3 is the algorithm used for user identification.

Algorithm Name: User Identification

Input: Clean Web Log Data.
Output: Number of Distinct User.
Step1: Read records from Web Log File.
Step2: User's IP addresses of two consecutive entries are compared.
Step3: IF (IP address is same) THEN
Check user's browser and operating system
IF both are same THEN
Consider same user.
ELSE
Consider new user.
END IF
END IF

Step 4: Repeat above 2 steps until EOF (Web Log File).

Figure 4.3 Algorithm for User Identification

4.3 Sampling

Taking into account big volume of input data, 3.4 million records, sampling has been proposed to be used for data reduction. The objective of sampling is to use subset of the whole data set for data analysis which saves time and cost while maintaining accuracy of the output result. In this thesis, Stratified Random Sampling (SRS) is used for data reduction. Its strength includes minimizing sample selection bias and ensuring certain segments of the data are not over-represented or under-represented.

Stratified random sampling is a method of sampling that involves the division of a population into smaller groups known as strata. In stratified random sampling, the strata

are formed based on members' shared attributes or characteristics. A random sample from each stratum is taken in a number proportional to the stratum's size when compared to the population. These subsets of the strata are then pooled to form a random sample.

There are two factors that need to be taken into account when choosing the sample size, confidence level and confidence interval. Confidence level indicates the level of accuracy of the sampling result and confidence interval also called the margin of the error is plus or minus deviation of the result.

In this thesis, sampling calculator from a website provided by Creative Research Systems [16] has been used to estimate the subset of the data. Choosing 99% of confidence level and 0.5 confidence interval results 66,564. Out of 3.4 million, 66K is selected for pattern discovery process. It means that 1,960 records per day is used. The Table 4.3 shows chosen confidence level, confidence interval, sample size, number of days for all web logs and number of web log entries per day.

Table.4.3 Web Log Statistics Using Stratified Random Sampling

Confidence Level	99%
Confidence Interval	0.5
Sample size	66,564
# of days	34
# of web log per day	1,960

4.4 Clustering

Clustering is applied to the web log before discovering patterns as a pre-processing step. Clustering tends to group the users those share the similar access patterns. The ideas behind is that more relevant and meaningful patterns are discovered if association rule mining is applied on the clustered data than applying on the whole data set. In this thesis, Density Based Spatial Clustering of Applications with Noise (DBSCAN) is used to group users.

DBSCAN is selected for several reasons. First, the users don’t need to specify the number of clusters ahead. It is very clear that knowing the number of clusters in advance is not easy task. Second, it can find different shape of cluster. Third, it can identify noise when clustering and can remove them for sub-sequent process.

DBSCAN uses distance function to measure the similarity or dissimilarity among the object. Many distance functions are available in the literature and choosing the distance function is entirely rely on the type of data. To use distance function, first web log data need to be transform to certain data model which is compatible for distance function. In this thesis, user-page view binary matrix is formed using the user and the URL accessed. If a user accesses the specified URL, "1" value is used. If not, "0" value is used to form a binary matrix. Table 4.4 shows sample user-pageview binay matrix for eight users.

Table 4.4 User-pageview Binary Matrix

URL/User	1	2	3	4	5	6	7	8
/SahelSituation/admin/login-form.php	1	0	1	0	1	0	0	1
/SahelSituation/admin/rssfrom.php	0	0	0	0	0	0	0	0
/SahelSituation/country.php	0	0	0	1	0	1	0	0
/SahelSituation/documents.php	0	1	0	0	1	0	1	0
/SahelSituation/download.php	0	0	0	0	0	1	1	0
/SahelSituation/flash_read.php	1	0	1	0	1	0	0	0
/SahelSituation/highlights.php	0	1	0	0	0	0	0	1

Now data is ready to be used for as an input for distance function. Jaccard distance is recommended in this thesis as it can be used to measure the dissimilarity between two data binary objects. Jaccard distance between data binary objects can be calculated using the following the formula.

$$d_{(i,j)} = \frac{J_{01}+J_{10}}{J_{01}+J_{10}+J_{11}} \quad (4.1)$$

Where J_{11} is the number of occurrence in two data objects where both data values is 1. J_{01} or J_{10} is the number of occurrence in two data objects where either one value is 1.

Table 4.5 is sample Jaccard distance between eight users. The value 1 means that two objects are completely different while 0 represents that those objects are identical. For instance, the distance between the same users is 0. Once the distances between objects are calculated, it is ready to form clusters. DBSACN uses two parameters for clustering the data objects. The first parameter is called Epsilon which specifies how points should be closed to each other to be considered a part of a cluster; and minimum number of points, which specifies how many neighbors that a point

should have, to be included into a cluster. Choosing different parameters can lead to form different number of clusters. So, care should be taken when selecting parameters. Several clustering evaluation methods exist. In this system, purity score is used to examine whether selected parameters are good enough for clustering.

Table 4.5 Sample Jaccard Distance Matrix

User	1	2	3	4	5	6	7	8
1	0	0.9	0.25	1	0.25	1	1	1
2	0.9	0	0.9	0.9	0.9	1	1	1
3	0.25	0.9	0	0.9	0.9	1	1	1
4	1	0.9	0.9	0	1	1	1	1
5	0.25	0.9	0.9	1	0	1	1	1
6	1	1	1	1	1	0	1	1
7	1	1	1	1	1	1	0	1
8	1	1	1	1	1	1	1	0

To evaluate the performance of the clustering algorithm, validation measures should be used. There are two types of validation measures; they are internal validation measures and external validation measures. Purity score is one the external cluster evaluation methods to access the quality of clustering. It is a simple and transparent evaluation measure if compared to other evaluation methods. To compute purity score, each cluster is assigned to the class which is most frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned documents and dividing by total class members. The purity of each cluster is computed with the below formula.

$$purity(D_i) = max_j(Pr_i(c_j)) \tag{4.2}$$

The total purity of the whole clustering (considering all clusters) is

$$purity_{total}(D) = \sum_{i=1}^{k} \frac{|D_i|}{|D|} \times purity(D_i). \tag{4.3}$$

The below example demonstrates how to calculate purity score for three clusters. Cluster 1 has 2 class members (X,0) and maximum class label is X with the

number of count (5). Cluster 2 includes 3 attributes (+,0,X) in which 0 label has the highest count with 4 while cluster 3 has 3 purity score with (+) class label. To calculate purity score for all clusters, sum of each maximum purity score should be divided by total number of attributes from all cluster which is 17. The purity score for this clustering yields 0.71. Calculation of purity score can be found in the Table 4.6. Bad clustering has purity values close to 0, a perfect clustering has a purity of 1.

Table 4.6 Sample Calculation for Purity Score

Clusters	**Purity Score for Each Cluster**	**Purity Score for All Clusters**
{X,X,0,X,X,X}	X=5	5+4+3/17=0.71
{X,0,+,0,0,0}	0=4	
{+,+,+,X,X}	+=3	

Parameters for DBSCAN are selected using purity score which generates the highest result. Using the sample dataset 66,564, the purity score was calculated using the different parameters for epsilon and minimum number of points. The Table 4.7 shows the purity scores for different epsilon and minimum number of points. After comparing the data, purity score 0.38 is the highest result with 0.3 value for epsilon and 3 for minimum number of points for the sample dataset 66,564.

Table 4.7 Calculation of Purity Score for 66,564 Dataset

Minimum Number of Points	**Epsilon**								
	0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
2	0.25	0.34	0.38	0.27	0.19	0.19	0.17	0.2	0.19
3	n.a	0.24	**0.38**	0.26	0.14	0.17	0.17	0.2	0.19
4	n.a	0.23	0.34	0.37	0.14	0.14	0.17	0.2	0.19
5	n.a	0.25	0.31	0.37	0.14	0.14	0.17	0.2	0.19
6	n.a	0.25	0.31	0.37	0.14	0.14	0.17	0.2	0.19
7	n.a	0.28	0.31	0.36	0.14	0.14	0.17	0.2	0.19
8	n.a	0.28	0.32	0.35	0.14	0.14	0.17	0.2	0.19
9	n.a	0.28	0.35	0.35	0.15	0.14	0.12	0.2	0.19
10	n.a	0.29	0.35	0.35	0.15	0.14	0.13	0.2	0.19

n.a: Not applicable (no cluster)

Once required parameters for DBSCAN are identified, clustering process can be started. Epsilon (0.3) and minimum number of points (3) which generate the high score of purity are applied for the Jaccard distance matrix. DBSCAN picks up a point randomly from the dataset. Let us say user1 is selected and the process scans the neighborhood users. User3 and user5 are within the range of epsilon and the rest users are out of the range. As the process used 3 minimum number points to form a cluster, a cluster (C1) has been formed which includes user1, user2 and user3 and this can be found in the Table 4.8.

Table 4.8 A Sample Cluster with 3 Users

User	1	2	3	4	5	6	7	8
1	0	0.9	0.25	1	0.25	1	1	1
	C1	X	C1	X	C1	X	X	X

Within user2's range, there is no users within the epsilon range as all distance are bigger than 0.38. For user3, there is a user (user1) within the range but a cluster can be only formed with minimum 3 users. Then the process continues until all users are visited. At the end of DBSCAN, only a cluster is formed including user1, user3 and user5. All other users are identified as outliers and they are not taken into account for subsequent process. DBSCAN clustering result for eight users is shown in the Table 4.9.

Table 4.9 DBSCAN Result (0.3 epsilon, 3 Minpts)

User	1	2	3	4	5	6	7	8
1	C1	X	C1	X	C1	X	X	X
2	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X
4	X	X	X	X	X	X	X	X
5	X	X	X	X	X	X	X	X
6	X	X	X	X	X	X	X	X
7	X	X	X	X	X	X	X	X
8	X	X	X	X	X	X	X	X

4.5 Association Rule Mining

This is the final step of the whole process which generates the association rules for frequent itemsets. Apriori algorithm that is a well-known algorithm for association rule mining is used. This algorithm needs two parameters as well, minimum support which specifics how frequent the itemset is and minimum confidence which describes how strong a rule is. If minimum support is low, then all itemset which are not considered to be frequent are generated. Many numbers of rule can be generated, and which is not strong enough to consider "association rules" are generated if minimum confidence is too low. So, users can adjust the number of rules to be generated using these two parameters. To apply Apriori algorithm to the web log data, first data need to be converted to suitable form. Users can be considered as transactions and uniform resource locator (URL) can be itemset. User-pageview matrix is transformed to the form which is shown in the Table 4.10 where A,B,E and F refers to the URL accessed by users and URL to a single letter mapping is shown in the Table 4.11. Only those users (user1, user3 and user5) are fed into the association rule mining process.

Table 4.10 Users and URLs in Cluster 1

User	URL
1	A,B,F
3	A,F
5	A,E,F

Table 4.11 URLs Accessed by Users

A	/SahelSituation/admin/login-form.php
B	/SahelSituation/admin/rssfrom.php
C	/SahelSituation/country.php
D	/SahelSituation/documents.php
E	/SahelSituation/download.php
F	/SahelSituation/flash_read.php
G	/SahelSituation/highlights.php

Let minimum support is 0.7 (70%) and minimum confidence is 0.5(50%). Apriori algorithm use apriori knowledge, this means if an itemset is a frequent itemset, all subset of itemset are frequent as well. The algorithm uses this rule until no association rules are found. It works in two steps, first all frequent itemsets are generated which satisfies minimum support threshold. Second it generates all association rule mining from frequent itemsets which is equal to or greater than minimum confidence.

The process starts counting one-itemset L1 as below.

L1= {A:3,B:1,E:1,F:3}

Minimum support of A and F are 3/3=1(100%) while B, E has only 1/3=0.33(33%). As minimum support for this process is 70%, B and E are removed. Then, one frequent itemset F1 becomes

F1 = {A:3, F:3}

The process continues finding 2-itemset by joining two 1-itemset together. It is assumed that the dataset is ordered by alphabetical order. Then, two-itemset L2 becomes

L2 = {{A, F}:3}

Out of those 2-itemsets, if 1-itemset subset of 2-itemeset are not frequent, those 2-itemsets are removed from the process. As A, F are appeared altogether in 3 transactions out of 3 transactions. So, minimum support is 3/3=1(100%) and both subset of {A, F} itemset (A and F) are frequent itemsets in F1. Then, two frequent itemset becomes

F2 = {{A,F}:3}

As A, F is the only one 2-itemset, the process for finding frequent itemset is stopped here. The second step is to generate association rules from frequent itemset which comply the minimum confidence (50%). To generate rules for every frequent itemset f, all nonempty subsets of f are used. For each such subset α, rules are generated based on the below form

$$(f-\alpha)\rightarrow \alpha \; if \; confidence = \frac{f.count}{(f-\alpha).count} \geq minconf \qquad (4.4)$$

Where f.count (or (f - α).count) is the support count of f (or (f -α)). The below rule is generated according to the above form. Number of occurrences of A, F together is 3 which is already available in F2. A also appears in all 3 transactions. So,

A→F (3/3=1(100%))

/SahelSituation/admin/login-form.php → /SahelSituation/flash_read.php

(minsup: 100%, minconf: 100%)

After running Apriori algorithm, there is only one rule which satisfies both minimum support and minimum confidence provided by the users. Next topic will present how this system is implemented.

4.6 System Implementation

The system is implemented using python programming language and sqlite database is used as a storage layer. The system is designed to work as a portable software package. This mean users don't need to install this software, just copy and paste on their hard drive and start using it. The user interface is simply designed so that users can easily follow the steps. Once users start the application, two main menus (File and Help). Users can import the dataset using File menu and Help is for more information about the system (for example, python version and operation system being used). The application also displays four main tabs (Data Cleaning, User Identification, Clustering and Association Rule Mining). Except data cleaning tab, the rest tabs are disabled as data importing and cleaning is the first step needs to be followed by users. The Figure 4.4 shows the main interface of the system.

Figure 4.4 Main User Interface for Pattern Discovery Process

As mentioned earlier, data can be imported using Import submenu from File menu. Once users click the Import submenu, a message box is appeared to ask for the

location of the web log data. A single file or multiple files can be selected depending on users' needs. File dialog box showing users to choose log files can be seen in the Figure 4.5.

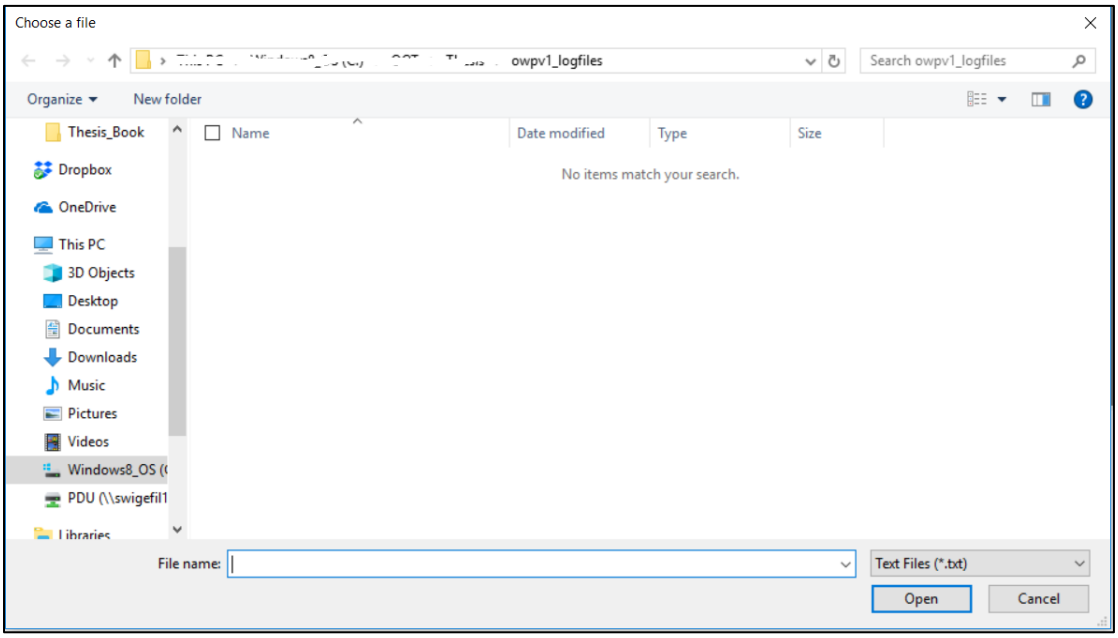


Figure 4.5 File Dialog Box to Choose the Web Log Data

If the file format is the same as the log file content mentioned in the Table 4.2, then the application continues importing and cleaning the data. If the format is different, the application displays the error message to the users. If data is successfully imported and cleaned by the application, it notifies users and raw web log data which is plain data from web log file and clean data removing unnecessary data according to the data cleaning algorithm. The Figure 4.6 can be seen once the data importing is successful. The application also displays number of raw log web entries without cleaning, number of clean log data and the percentage of reduced log entries in application console so that users are well informed about the process and it is shown in the Figure 4.7.

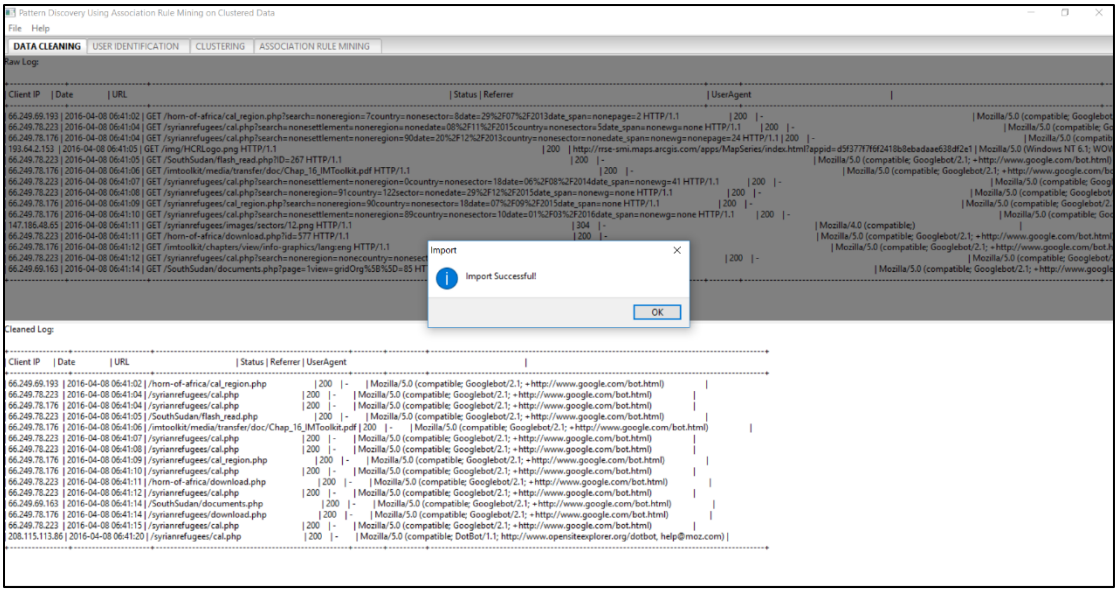


Figure 4.6 Data Importing and Cleaning

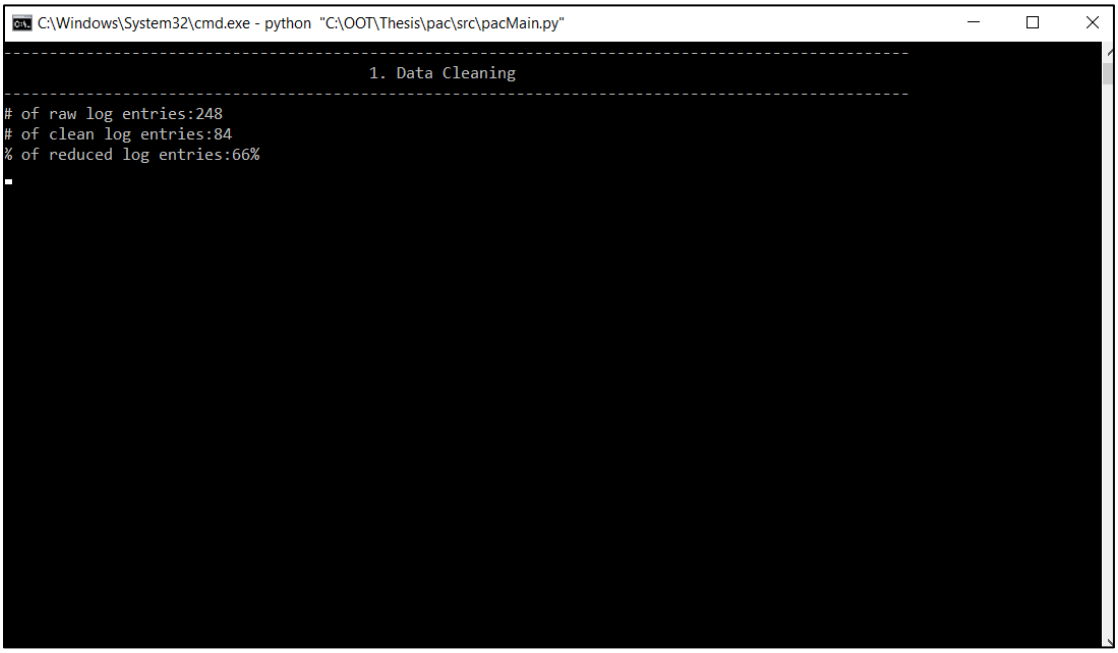
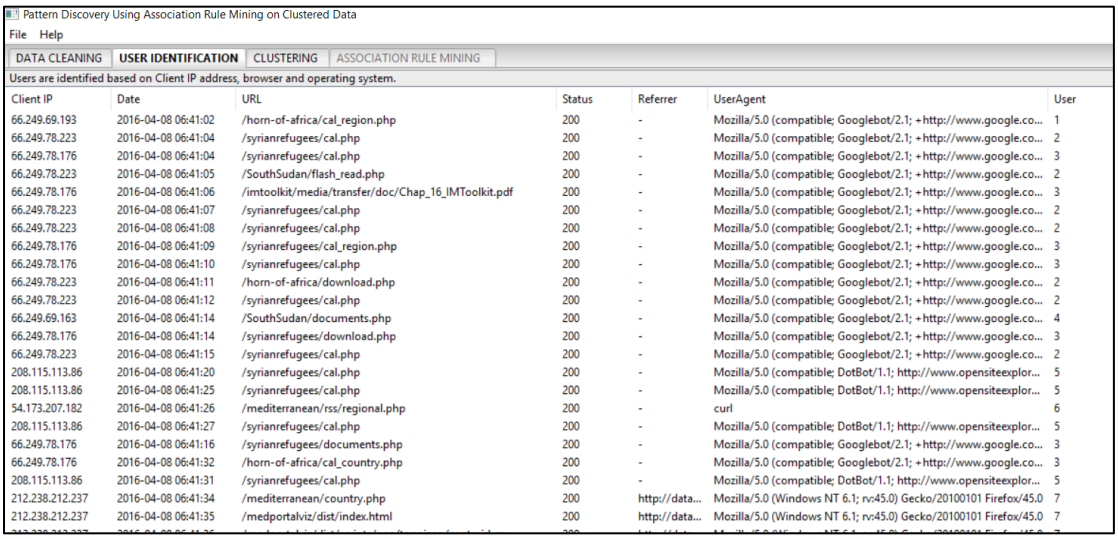


Figure 4.7 Log Entries for Data Cleaning

After data cleaning, web log data need to be segmented according to the users accessed to the website. Two parameters, client IP address and useragent, are used to estimate user identification. If both parameters are the same, it is assumed to be the same user while one of parameters is different, it is likely to be different user. After user identification process, the application displays web log data including the number

of users and the Figure 4.8 shows the interface for user identification. Total number of users identified is logged in application console and it is displayed in the Figure 4.9.



Client IP	Date	URL	Status	Referrer	UserAgent	User
66.249.69.193	2016-04-08 06:41:02	/horn-of-africa/cal_region.php	200	-	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.co...	1
66.249.78.223	2016-04-08 06:41:04	/syrianrefugees/cal.php	200	-	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.co...	2
66.249.78.176	2016-04-08 06:41:04	/syrianrefugees/cal.php	200	-	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.co...	3
66.249.78.223	2016-04-08 06:41:05	/SouthSudan/flash_read.php	200	-	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.co...	2
66.249.78.176	2016-04-08 06:41:06	/imtoolkit/media/transfer/doc/Chap_16_IMToolkit.pdf	200	-	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.co...	3
66.249.78.223	2016-04-08 06:41:07	/syrianrefugees/cal.php	200	-	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.co...	2
66.249.78.223	2016-04-08 06:41:08	/syrianrefugees/cal.php	200	-	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.co...	2
66.249.78.176	2016-04-08 06:41:09	/syrianrefugees/cal_region.php	200	-	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.co...	3
66.249.78.176	2016-04-08 06:41:10	/syrianrefugees/cal.php	200	-	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.co...	3
66.249.78.223	2016-04-08 06:41:11	/horn-of-africa/download.php	200	-	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.co...	2
66.249.78.223	2016-04-08 06:41:12	/syrianrefugees/cal.php	200	-	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.co...	2
66.249.69.163	2016-04-08 06:41:14	/SouthSudan/documents.php	200	-	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.co...	4
66.249.78.176	2016-04-08 06:41:14	/syrianrefugees/download.php	200	-	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.co...	3
66.249.78.223	2016-04-08 06:41:15	/syrianrefugees/cal.php	200	-	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.co...	2
208.115.113.86	2016-04-08 06:41:20	/syrianrefugees/cal.php	200	-	Mozilla/5.0 (compatible; DotBot/1.1; http://www.opensiteexplor...	5
208.115.113.86	2016-04-08 06:41:25	/syrianrefugees/cal.php	200	-	Mozilla/5.0 (compatible; DotBot/1.1; http://www.opensiteexplor...	5
54.173.207.182	2016-04-08 06:41:26	/mediterranean/rss/regional.php	200	-	curl	6
208.115.113.86	2016-04-08 06:41:27	/syrianrefugees/cal.php	200	-	Mozilla/5.0 (compatible; DotBot/1.1; http://www.opensiteexplor...	5
66.249.78.176	2016-04-08 06:41:16	/syrianrefugees/documents.php	200	-	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.co...	3
66.249.78.176	2016-04-08 06:41:32	/horn-of-africa/cal_country.php	200	-	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.co...	3
208.115.113.86	2016-04-08 06:41:31	/syrianrefugees/cal.php	200	-	Mozilla/5.0 (compatible; DotBot/1.1; http://www.opensiteexplor...	5
212.238.212.237	2016-04-08 06:41:34	/mediterranean/country.php	200	http://data...	Mozilla/5.0 (Windows NT 6.1; rv:45.0) Gecko/20100101 Firefox/45.0	7
212.238.212.237	2016-04-08 06:41:35	/medportalviz/dist/index.html	200	http://data...	Mozilla/5.0 (Windows NT 6.1; rv:45.0) Gecko/20100101 Firefox/45.0	7

Figure 4.8 User Identification

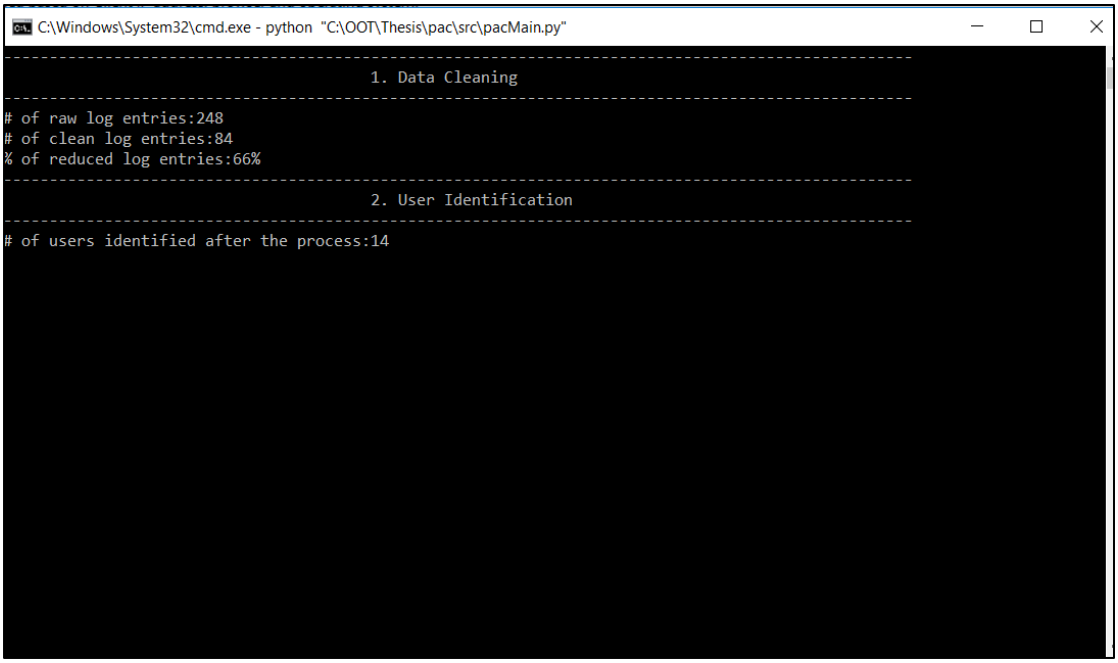
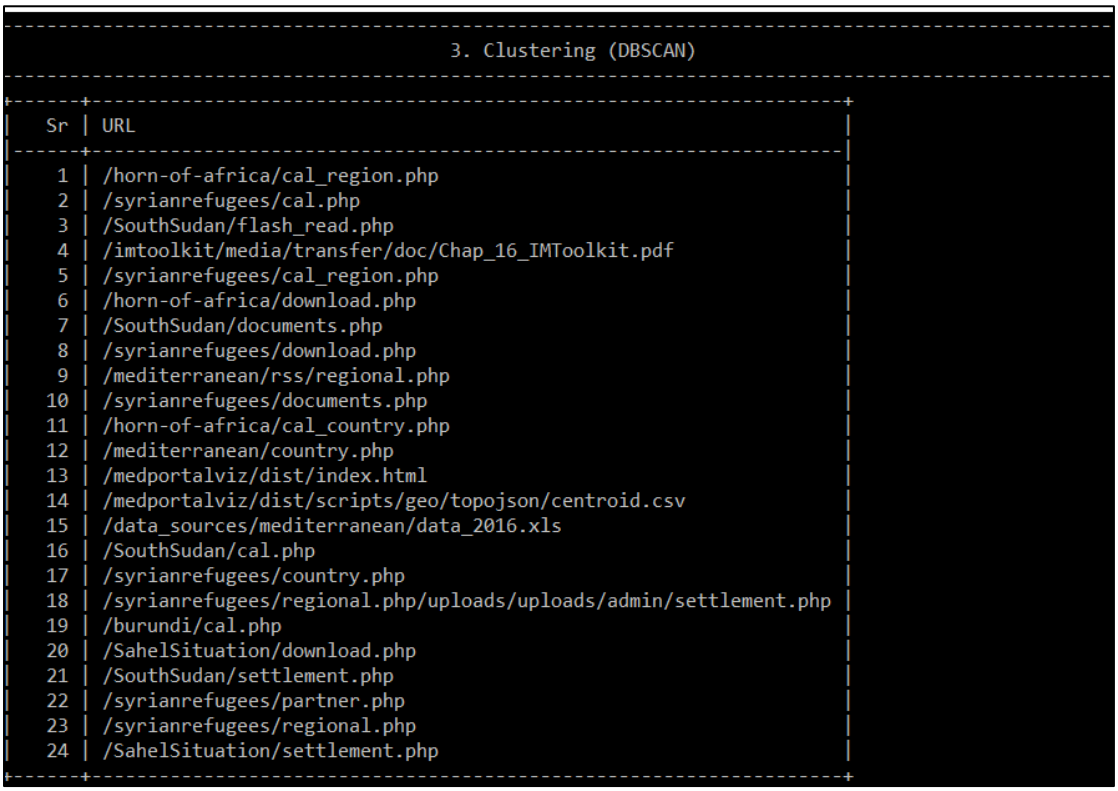


Figure 4.9 Log Entries for User Identification

After data is clean and users are identified, data is ready to be grouped based on their similarities. Clustering technique, DBSCAN is used to form clusters. DBSCAN is density based clustering algorithm and first it looks that the neighborhood objects around a point. If the area is dense enough according to the parameters specified by users, a cluster is formed. To measure the distance between objects, Jaccard distance

is used to calculate the distances between users. Before calculating the distance between objects, user-pageview matrix is converted. As URLs are long strings, it is easy to work with numbers rather than using the long strings. Using long strings may have impact on system performance as well. So, it was decided to encode them as numbers. The Figure 4.10 shows how the data are logged in application console during clustering process.



3. Clustering (DBSCAN)

Sr	URL
1	/horn-of-africa/cal_region.php
2	/syrianrefugees/cal.php
3	/SouthSudan/flash_read.php
4	/imtoolkit/media/transfer/doc/Chap_16_IMToolkit.pdf
5	/syrianrefugees/cal_region.php
6	/horn-of-africa/download.php
7	/SouthSudan/documents.php
8	/syrianrefugees/download.php
9	/mediterranean/rss/regional.php
10	/syrianrefugees/documents.php
11	/horn-of-africa/cal_country.php
12	/mediterranean/country.php
13	/medportalviz/dist/index.html
14	/medportalviz/dist/scripts/geo/topojson/centroid.csv
15	/data_sources/mediterranean/data_2016.xls
16	/SouthSudan/cal.php
17	/syrianrefugees/country.php
18	/syrianrefugees/regional.php/uploads/uploads/admin/settlement.php
19	/burundi/cal.php
20	/SahelSituation/download.php
21	/SouthSudan/settlement.php
22	/syrianrefugees/partner.php
23	/syrianrefugees/regional.php
24	/SahelSituation/settlement.php

Figure 4.10 Encoding URLs to Numbers

Once URLs are encoded as numbers, user-pageview matrix is built. The rows represent the users while the column headers are for URLs. When forming user-pageview matrix, 1 is used if the users access that URL, if not 0 is used instead. Jaccard distances are calculated among the users. The closer to 0 the distance is, the higher similarity among the users. The distance 1 means there is no similarity at all. The Figure 4.11 is logged in the application console how user-pageview matrix and Jaccard distances are calculated among the users.

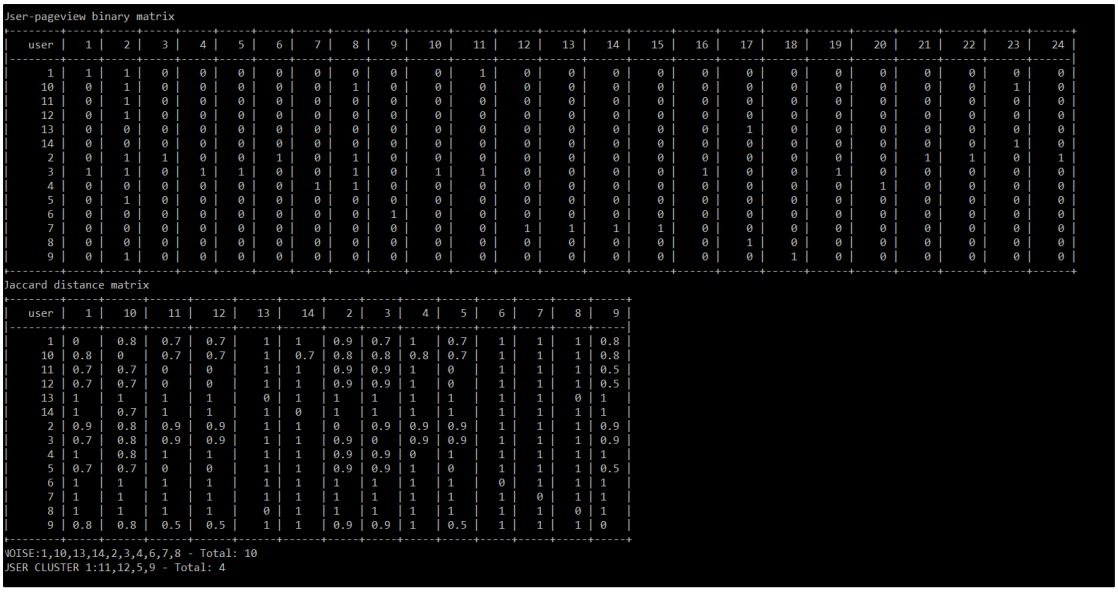


Figure 4.11 User-pageview and Jaccard Distance Matrix

All the above processes related to clustering are intended to display to the users how the application works internally. The user interface for clustering page requests two parameters, epsilon and minimum number of points. Once the parameters are selected and clustering process is started, the application shows not only the clustering result but also purity score which measure how goodness the clustering is. The score is between 0 and 1. The clustering process can be run several times by the users until the users satisfy the purity score and the clustering results. The clustering identifies which users are noise and which users are part of the clusters. Noise in another way called outliers are not taken into account for association rule mining process as these are dissimilar from other users according to the parameters provided by the users. The Figure 4.12 shows the DBSCAN result after running the clustering process.

Association rule mining which is the final step for this application can be proceeded after clustering. Association rules are generated based on clustered data, it means that each cluster is run for association rules separately. In this example, as there is only cluster, association rules are generated for a cluster. One of the well-known association rule mining algorithms called Apriori algorithm is used. It needs two parameters, minimum support and minimum confidence. Apriori algorithm is run using these two parameters and the result is displayed in the text. The internal process shows how Apriori algorithm works and the final result output to users is displayed in the Figure 4.13 and the Figure 4.14 respectively. These include, how web log data are

converted into the transactions so that Apriori algorithm can apply on this dataset, and frequent access patterns with their respective minimum support which is equal to or greater than the minimum support parameters provided by users. The last items in the application console are association rules with their minimum support and minimum confidence.

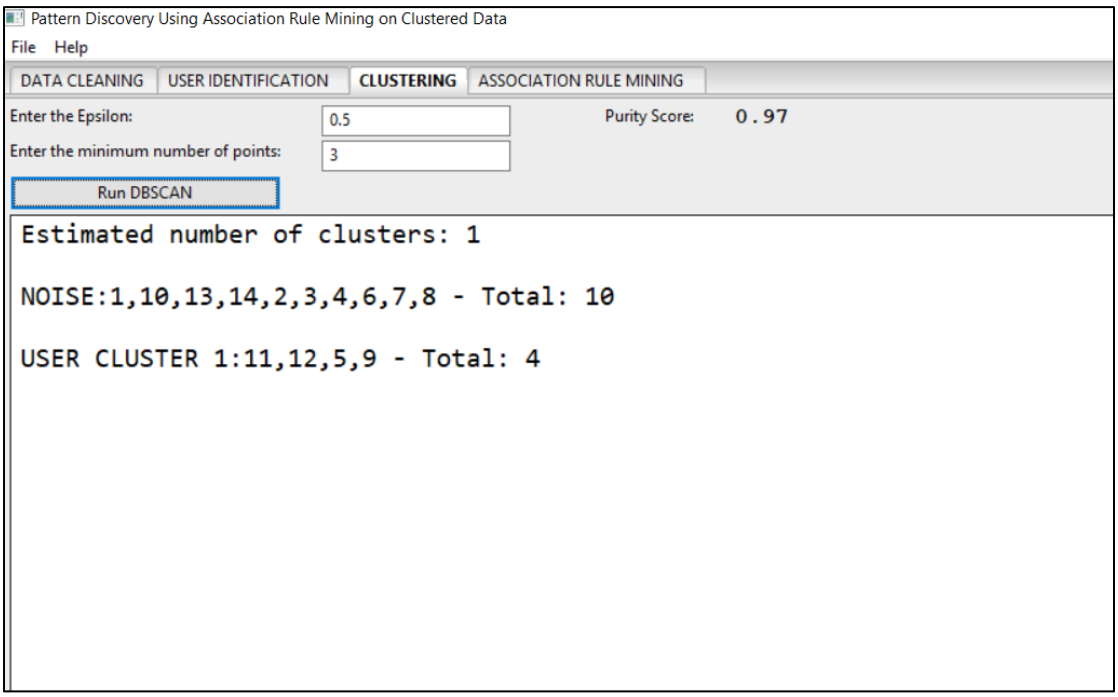


Figure 4.12 DBSCAN Clustering

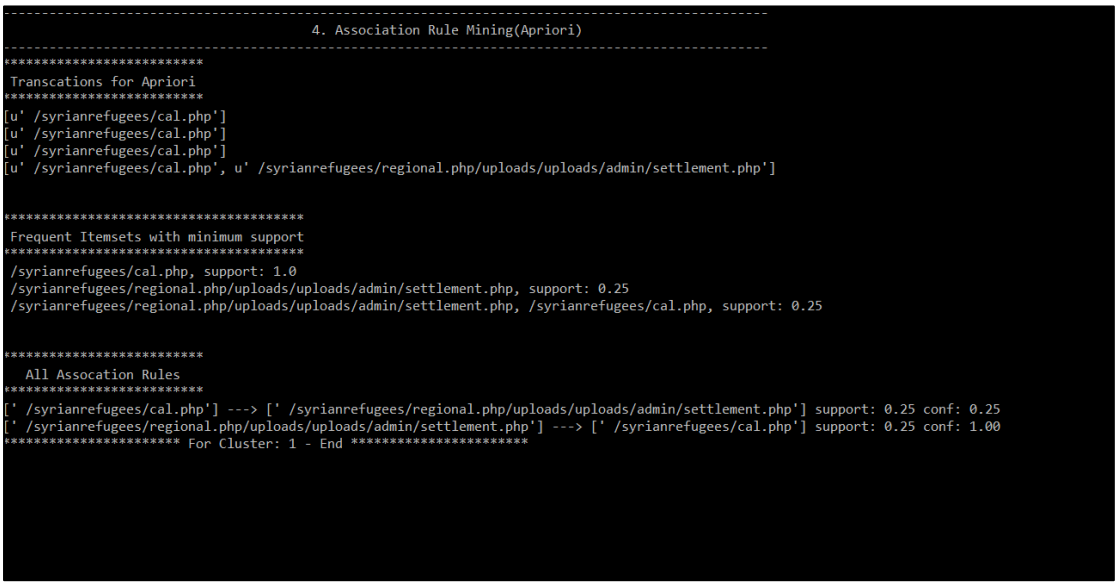


Figure 4.13 Logging for Apriori Algorithm

Figure 4.14 Association Rule Mining

4.7 Experimental Result

Two factors are considered when evaluation this system, the data reduction and number of rules generated.

4.7.1 Data Reduction

Before applying association rule mining, input data should be cleaned as much as possible. In literature, many different techniques have been proposed to reduce the data. In this thesis, DBSCAN algorithm is used as part of pre-processing step. After running DBSCAN algorithm on three clustered and non-clustered datasets those are D1, D2 and D3 experimental results proved that data is dramatically reduced. All three datasets are reduced by almost average 99% of the existing cleaned log entries which improves the efficiency of Apriori algorithm as less data needs less computation. This also fixes one of the major drawbacks of Apriori algorithm as the performance of the algorithm is directly proportional to the size of the input dataset. The Figure 4.15 displays the number of clean log entries before and after using DBSCAN clustering technique.

4.7.2 Generating Relevant and Concise Rules

One of the problems of association rule mining is generating too many rules and users are not able to find out which rules are more interesting and it doesn't produce no interesting rules with high minimum support and confidence at all. For this case, the

rules with low minimum support and confidence are not interesting for users. To overcome this problem, the below analysis shows that clustered association rule mining generates rules with high minimum support and confidence to user while non-clustered association rule mining doesn't generate any rules at all. Number of rules generated based on the provided parameters between clustered data and non-clustered data are displayed in the Figure 4.16.

Dataset	# of Clean Log Entries before Clustering	# of Clean Log Entries after Clustering	Reduction %	Note
D1	22,950	369	-98%	Using 0.3 epsilon and 3 minimum number of points which yield 0.56 purity score.
D2	43,758	604	-99%	Using 0.3 epsilon and 3 minimum number of points which yield 0.51 purity score.
D3	66,564	909	-99%	Using 0.3 epsilon and 3 minimum number of points which yield 0.38 purity score.

Figure 4.15 Number of Clean Log Entries before and after Clustering

Dataset	Minimum Support	Minimum Confidence	Number of Rules for Non-Clustered ARM	Number of Rules for Clustered ARM
D1	0.001	0.2	791	6,070
	0.01	0.01	2	394
	0.1	0.1	0	152
	0.5	0.5	0	26
	0.8	0.8	0	12
	0.9	0.9	0	10
	1	1	0	8
D2	0.001	0.2	479	16,690
	0.01	0.01	4	3,078

Dataset	Minimum Support	Minimum Confidence	Number of Rules for Non-Clustered ARM	Number of Rules for Clustered ARM
D2	0.1	0.1	0	2,750
	0.5	0.5	0	273
	0.8	0.8	0	52
	0.9	0.9	0	50
	1	1	0	50
D3	0.001	0.2	733	2,566,254
	0.01	0.01	2	3,473,511
	0.1	0.1	0	16,580
	0.5	0.5	0	115
	0.8	0.8	0	27
	0.9	0.9	0	25
	1	1	0	18

Figure 4.16 Number of Rules Generated on Clustered vs. Non-clustered data

CHAPTER 5
CONCULSION AND FURTHER EXTENSION

5.1 Conclusion

Web usage mining techniques are great area of research these days. The ultimate goal of web usage mining should provide users easily what they are looking for in websites. In this thesis, this goal is fulfilled by using association rule mining technique on clustered data. For clustering, DBSCAN is used to group users who share similar access patterns. When generating association rule mining, Apriori algorithm is deployed.

UNHCR is developing a new website and it is important to understand how users are using the current website. Discovering frequent user access patterns greatly improve the new website layout and those contents which are accessed frequently accessed together can be put closed to each other. For this study, 30 days web log files from UNHCR's operational data portal has been used. After compiling all 30 days web log files, the file size is very huge, and it is more than 8 GB including all invalid entries. Stratified sampling technique is used to select certain portion of the data so that the sample data is represented enough to make the conclusion. Online sample size calculator is used to estimate the sample size with 99% confidence level and 0.5 confidence interval. Three datasets have been used for this study. The size of the data is 4 MB, 8 MB and 13 MB respectively.

Using DBSCAN as a pre-processing step before generating association rules make a big difference. Based on experimental result using three datasets, DBSCAN reduces the data size by 99% in the current application domain. This means only users who have similar access patterns are used while users those have different access patterns from other users are removed. When applying Apriori algorithm on the clustered data, fewer number of rules with high minimum support and minimum confidence are generated. Without clustering, all web logs are used for association rule mining and too many rules are generated with low minimum support and confidence while there is no rule at all when users specify high minimum support and confidence. So, user access patterns with high minimum support and confidence are interesting for this study as these rules can be used to change the whole layout website structure.

Association rule mining may have drawback of generation of irrelevant rules, generation of too many rules leading to contradictory prediction resulting in reduction of accuracy. Clustering reduces data set for association rule mining and produces relevant frequent access patterns from each cluster. Running DBSCAN is entirely depending on two parameters, epsilon and minimum number of points. In this system, these two parameters are carefully chosen using purity score clustering evaluation technique. So, understanding underlying data is necessary and without the advance knowledge of the data, inappropriate parameters may be chosen by user and this may have impact on generating association rules. When generating the association rules, Apriori algorithm is applied to each cluster one by one. So, the processing time may take longer than using multi-thread programming model where Apriori algorithm is applied to all clusters all the same time.

5.2 Further Extension

This system can further be extended by selecting optimal parameters for DBSCAN automatically based on the nature of the data. In this thesis, the system allows the users to select the parameters manually by looking at purity score. Based on the parameters provided by the users, different clusters will be formed, and this may lead to generate different association rules though the same dataset has been used. To improve this, the system could calculate automatically the best parameters for DBSCAN and it could also reduce user intervention.

Another extension could be using multi-threaded programming model when finding association rules. In this system, association rule mining is applied on each cluster sequentially. This could result the processing time takes longer. By using multi-threaded programming model, Apriori algorithm can be applied to all different clusters at the same time and the processing time is expected to take lesser than the sequential running time.

PUBLICATION

[1] Htun Zaw Oo, Nang Saing Moon Kham, Pattern Discovery Using Association Rule Mining on Clustered Data, International Journal of New Technology and Research (IJNTR), ISSN:2454-4116, Volume-4, Issue-2, February 2018, pp. 07-11.

REFERENCES

[1] Aarti M Parekh, Anjali S Patel, Sonal J Parmar, Vaishali R Patel, Web usage mining: frequent pattern generation using association rule mining and clustering. Internatioinl Journal of Engineering Research & Technology (IJERT), ISSN 2278-0181, Vol. 4 Issue 04, pp. 1243 - 1246, April-2015.

[2] Anjali Nehete, Dipa Dixit, Kiruthika M, Rahul Jadhav, Rashmi J, Trupti Khodkar, Pattern discovery using assocation rules. International Journal of Advanced Computer Science and Applications, Vol. 2, No. 12, pp. 69 - 74, 2011.

[3] Bhaiyalal Birla, Sachin Patel, Hemlata Sunhare, Comprehensive Framework for pattern analysis through web logs using web mining. IJCSMC, Vol. 2, Issue 4, pp. 32 - 37, April 2013.

[4] Bing Liu, Web data mining – exploring hyperlinks, contents, and usage data, Springer, ISBN 978-3-642-19459-7, 2011.

[5] D Sharmila, R Suguna, Association rule mining for web recommendation. Internalation Journal on Computer Science and Engineering (IJCSE), Vol. 4, No. 10, pp. 1686 - 1690, 2012.

[6] J Just, A short survey of web data mining, WDS'13 Proceedings of Contributed Papers, Part I, ISBN 978-80-7378-250-4, pp. 59–62, 2013

[7] J Han, M Kamber, Data mining concepts and techniques. Morgan Kaufmann Publishers, San Francisco, ISBN 1558604898, 2001.

[8] Jiawei Han, Micheline Kamber, Data mining concepts and techniques. Second Edition, Elsevier, 2006, ISBN 978-1-55860-901-3, pp. 630.

[9] Khandakar Entenam Unayes Ahmed, Md Khairul Islam Bhuiyan, Md Mahamudur Rahaman, Shahnaz Parvin Nina, Pattern disocovery of web usage mining. International Conference on Computer Technology and Development, 2009.

[10] Lior Rokach, Oded Maimon, Data mining and knowledge discovery handbook. Chapter 15, pp. 321-352, Clustering Methods, Springer, 2005.

[11] M Zaki, Scable algorithms for assocation mining, IEEE Transactions on Knowledge and Data Engineering, vol. 12, no.3, pp. 372-390, 2000.

[12] Naresh Barsagade, Web usage mining and pattern discovery: A Survey Paper. December 8, 2003.

ONLINE DOCUMENTS

[1] Operational Data Portal is interagency information sharing and coordination tool provided by UNHCR.
Date of access: March 2017.
http://data.unhcr.org

[2] New version of Operational Data Portal is interagency information sharing and coordination tool provided by UNHCR
Date of access: July 2018.
http://data2.unhcr.org

[3] Sample size calculator provided by Creative Research Systems
Date of access: July 2018
http://www.surveysystem.com/sscalc.htm