# Strong Information Scent based Search Engine

# (Web Mining and Information Foraging)

A Dissertation submitted in partial fulfilment of the requirement for the

Award of degree of

**MASTER OF TECHNOLOGY**

**IN**

**INFORMATION SYSTEMS**

By

**PIYUSH KUMAR**

**Roll No. – 2K12/ISY/19**

Under the esteemed guidance of

**DR. O.P VERMA**
**PROFESSOR AND HEAD**
**DEPARTMENT OF INFORMATION TECHNOLOGY**


**Department of Information Technology**

**Delhi Technological University**

**2012-2014**

# Abstract

"Wealth of information creates the poverty of attention" ~Herb Simon. World wide web is the grid of valuable information which is extending with the enormous speed. World Wide Web consists of billions of web pages and their urls and thousands of web pages keeps adding every day. Search engines provide the way to access these web pages and their links based on the query fired by the users and in return, search engine provides the users with several links of the web pages which may consist of the user's information need. But, due to the ample of links provided by the search engine in response to the fired query, user may get confused which link to follow, which will contain his/her information need. In the proposed method, we are embedding a novel optimization technique for search engine which considers the web log mining of the different user's log and the information foraging theory concept i.e. information scent in order to provide the user with the more optimized links of the web pages in order to reduce their information snacking time.

# ACKNOWLEDGEMENT

I express my sincere thanks and deep sense of gratitude to my project guide**, Dr. O.P. Verma**, Professor and Head of Department, Department of Information Technology, Delhi Technological University, for his valuable motivation and guidance, without which this study would not have been possible. I consider myself fortunate for having the opportunity to learn and work under his supervision and guidance over the entire period of association.

I humbly extend my words of gratitude to other faculty members and my friends for providing their valuable help and time whenever it was required.

Piyush Kumar

Roll No.2K12/ISY/19

M.TECH(Information Systems)

E-mail: piyush.k9013@gmail.com

# CERTIFICATE


This is to certify that the thesis entitled **"Strong Information Scent based Search Engine (Web Mining and Information Foraging)"** submitted by **Piyush Kumar (2K12/ISY/19 )** to the Delhi Technological University, New Delhi for the award of the degree of **Master of Technology** is a bonafide record of research work carried out by him under my supervision.

To the best of my knowledge, the contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

(Project Guide)

**Dr O.P VERMA**

Professor and Head of Department

Dept. of Information Technology

Delhi Technological University

Bawana Road, New Delhi-110042

# Table of Contents

**Title** **Page No.**

# List of Figures

# List of Tables

# Chapter 1: Introduction

We are living in the era of information age. Every day, multi-set of web pages ,terabytes or petabytes of data drain into the computer networks, world wide web and various storage devices from science and engineering, medicine, business society and almost from every aspect of daily life.

This explosion of the available data volumes is the repercussion of the computerization of our society and rapid development of the powerful data collection and storage tools. The catalogue of the data sources that generate the ample amount of data is endless. Hence, there is an indispensible need of the powerful and versatile tools which can uncover the valuable information from the tremendous amount of data and to transform such data into organized knowledge. This necessity has led to the need of the web search engines

A web search engine is a specialized computer server that hunts for the information from the web. It receives millions of queries every day. Each query can be viewed as the transaction where the user describes his or her information need. Web search engines are actually very large web mining applications. It uses various web mining techniques like web content mining, web log mining, web structure mining and web crawling[1],[2].

But, web search engine poses grand challenges to web mining i.e. it has to handle a huge and ever-growing data and returning the result from such a huge data source is very tedious and over that returning the optimized result is even more fugacious.

## 1.1 Motivation

The optimization techniques of the search engine such as Page Rank algorithm, Weighted Page Rank algorithm, HITS (Hypertext Induced Topic Search) has improved the searching from, such a vast amount of web data, to a very large extent. These algorithms find the rank of each page which defines the quality of a page and use these ranks to return the web result to the user in

response to their query. The results are returned in the decreasing order of their ranks. But still, the problem is that the user will get large number of links in return to their queries. The users information may be satisfied by the top rank search engine results or may not be. Eventually, the user has large number of returned links but user may not be able to decide which link to follow. He/she may follow different different links and probably won't get the required information. The user may get confused which link to navigate when addressed by the search engine web result. This only increases the user's navigation time to fetch the information. So, we want some novel approach to return the subset of the web search result of the search engine in which the probability of satisfying the user's information need to a great extent. This lead to save the user's navigation and information snacking time.

## 1.2 Research Objective

In the proposed method, we are trying to embed the information foraging theory concept i.e. information scent with the web mining techniques in order to evolve a search engine which will return the optimized result to the user. In this method, first, the query log file is mined to discover the similarity between the query keywords of different users and the clicked urls. In second step, the query clusters and the clicked url clusters are created. In third step, the combined similarity measure of query clusters and the clicked urls clusters are used to recommend the strong information scent based urls to the user. It means the clusters which closely approximate the information need of the input query of the user are used to return the urls from the cluster with similar information need for a given query. The returned urls will be strong information scent based urls which will reduce the navigation time.

## 1.3 Organization of thesis

The thesis is organised in five chapters. In chapter 1, the motivation of the thesis and research objective is listed. In chapter 2, data mining and its types followed by web mining and its techniques, accomplishments and applications of web mining are explained in detail. This chapter also includes definition of information foraging, details of theory of information foraging

and concept of information scent. Working of search engine optimization and its techniques are also explained in this chapter followed by the different algorithms and formulas reviewed such as K-means clustering, frequent Itemset mining algorithm, page rank formula, information scent metric, similarity and dissimilarity formula for measuring data, data normalization formula. In chapter 3, proposed methodology is listed in which its architecture, details of architecture and similarity analyzer is explained. In chapter 4, experimental result is shown. Chapter 5 concludes the thesis.

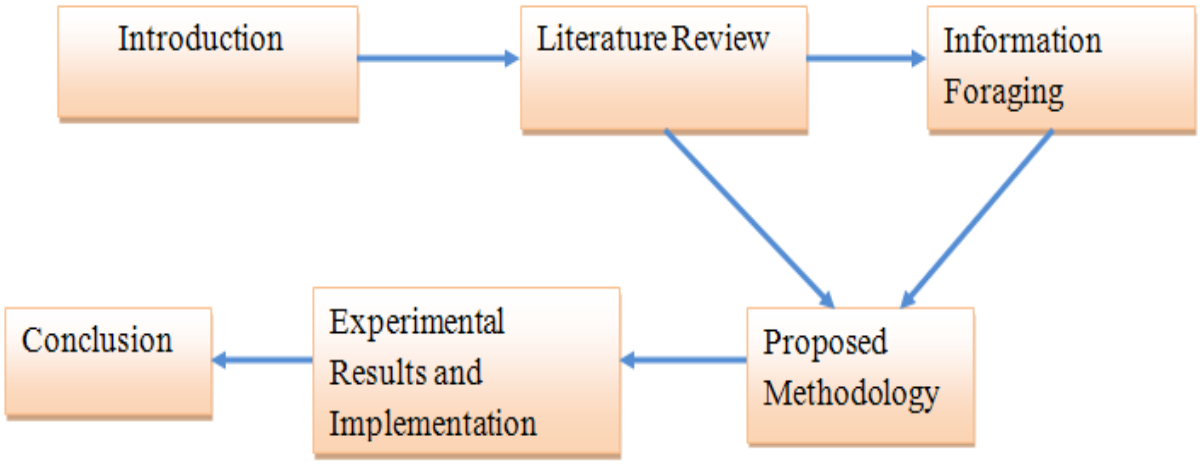


**Figure 1.1:** Thesis Organisation Road Map

## 2.1 Data mining

Data mining [5] is the knowledge discovery process. It is used to uncover the hidden patterns and relationships, and discover knowledge in the raw data that we never suspected to exist in the data.

### 2.1.1 Phases of Knowledge Discovery Process

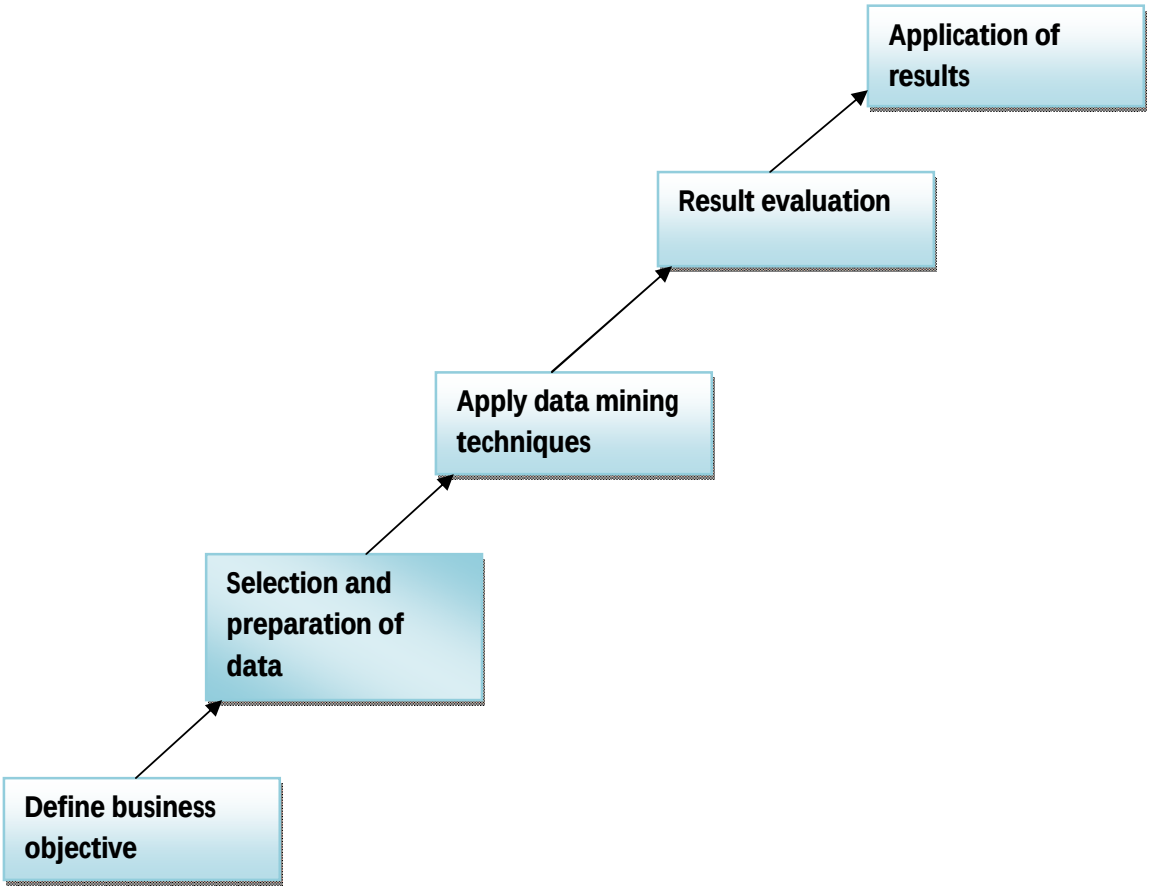


**Figure 2.1:** Knowledge Discovery Process

- **Define business objective:** In this phase, first of all determine whether we really need data mining. Then, clearly define the objectives e.g. Do you want to detect fraudulent activity in credit card usage etc, for data mining. Also, define how the final result will be presented and will be used in the operational systems.
- **Selection and preparation of data:** This phase comprises of data selection, preprocessing, data transformation. First, use the business objectives to select the data to be extracted from the data warehouse. Also, include the appropriate metadata which describes the selected data. Second, preprocessing of the selected data is needed to enrich the quality of the selected data and remove the unwanted noisy data. Third, transform the data into the proper format for the mining process.
- **Apply data mining techniques:** This is the pivotal phase of knowledge discovery process. In this phase, selected data mining algorithm is applied in order to gain patterns and relationships. This step and the next step i.e. result evaluation are performed in the iterative manner. The result to each iteration can be used to set the data for the next iteration.
- **Result evaluation:** In this phase, the patterns and relationships that are discovered in previous steps are evaluated, examined and unwanted patterns are filtered out and select only promising patterns. This phase also visualizes the results in graphs, charts or free-format texts.
- **Application of results:** This phase involves the results of knowledge discovery process to be applied in order to take some necessary actions which can be exploited to improve business.

### 2.1.2 Data Mining Techniques

There are various kinds of data mining techniques which assist mining the data according to the different criteria. Following sections explain some famous data mining techniques in detail.

#### (A) Cluster Analysis

Cluster analysis or clustering [3], [5] is the process of portioning a set of data objects (or observations) into subsets. Each subset represents a cluster, such that data objects in a cluster are similar to each other and are dissimilar to the data objects in other cluster. Clustering is also

known as unsupervised learning because the analysts have no prior knowledge of what they are looking for and how the result is likely to be.

Clustering can be used for :-

- **Data Segmentation**: Clustering segregates the large data sets into groups according to their similarities.
- **Outlier Detection**: Clustering used to detect the outliers(values which is far away from any cluster [3], [5]). It filters out any noisy data. Example-fraud detection in credit card usage etc.

**Requirements of clustering in data mining**

- **Scalability:** Highly scalable clustering algorithms are needed. Since clustering on the large data set objects does not give promising results whereas clustering on small data set objects give fine results. Hence, highly scalable clustering algorithm is required in order to get unbiased and fine results irrespective of the amount of data.
- **Ability to deal with the different types of attributes:** Clustering algorithm must be designed in such a manner that it can work with any type of data types such as binary, nominal, ordinal, graphs, sequences, images, and documents.
- **Discovery of arbitrary shaped clusters:** The cluster algorithms must be able to detect the clusters of the arbitrary shape. Many clustering algorithms determine cluster on the basis of Euclidean distance which results in the spherical clusters with small size and density. Hence, misses some important data sets. Therefore, clustering algorithm must develop clusters of arbitrary shape.
- **Requirement for domain knowledge:** Many clustering algorithm requires user to provide the domain knowledge in the form of input parameters [3]. Clustering results are susceptible to the input parameters passed to clustering algorithm.
- **Ability to deal with noisy data:** Clustering algorithm must be sensitive to detect the noisy data and filter out such unwanted data. Many clustering algorithms are not susceptible to the noisy data, hence give erroneous results.
- **Incremental clustering and insensitivity to input order:** In many applications, clustering algorithms do not incorporate the incremental updates, which may arrive in the

application, into the existing clustering structure. Also, clustering may also be sensitive to the input data order. Clustering may give different result according to the input order. Hence, clustering algorithm must be incremental and insensitive to the input order.

- **Capability of clustering high dimensional data:** Many clustering algorithms work well with low dimensional data such as data set with two or three dimensions. But, do not work well with high dimensional data. Hence clustering algorithm must be able to handle and work fine with high dimensional data.
- **Constrained based clustering:** Clustering algorithms must also consider all the constrains that are required to perform clustering.

## Types of clustering techniques

- **K-means clustering:** K-means clustering [3], [5] is the centroid based partitioning technique. It considers the centroid of a cluster to form clusters. Centroid is the center point of the cluster which represents the mean value and the cluster is evolved around centriod. Centroid is the mean value of the data objects in the cluster. In this clustering, first, it randomly selects 'n' objects from the data set objects 'D', where n<=D. Each of the 'n' selected objects initially represents a centroid of a cluster. For each of the remaining data objects, an object is assigned to the cluster to which it is most similar based on the Euclidean distance between the object and cluster mean [3], [5]. Then, the k-means clustering, then, iteratively improves the cluster. It computes the new mean for each cluster using the objects assigned to the cluster in previous step. All the objects are reshuffled using the new cluster centroids. This whole process will keep iterating till the centers get stabled.

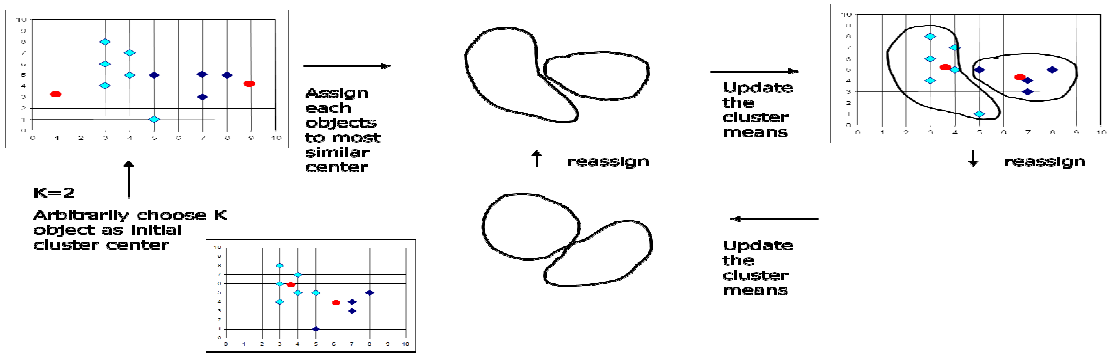


**Figure 2.2:** Cluster Analysis

- **Partitioning around medoids(PAM):** This clustering is done around the representative objects called medoids. In this process, initially 'n' objects from the data set objects 'D' are selected randomly called medoids(or representative objects). Then, assign each remaining object to the cluster with the nearest representative object. Then, randomly select the non representative object from the cluster. Swap this object with representative object and compute the total cost(C) of swapping. Keep swapping until we get the cost C which is less than zero. The total cost (C)(also called as absolute error criterion) is given by :-

$$\sum_{i=1} \sum_{p \in c(i)} dist(p, n(i)) \qquad (1)$$

where dist is the Euclidean distance,

c(i) is the ith cluster,

p is the non representative object,

n(i) is the representative object.

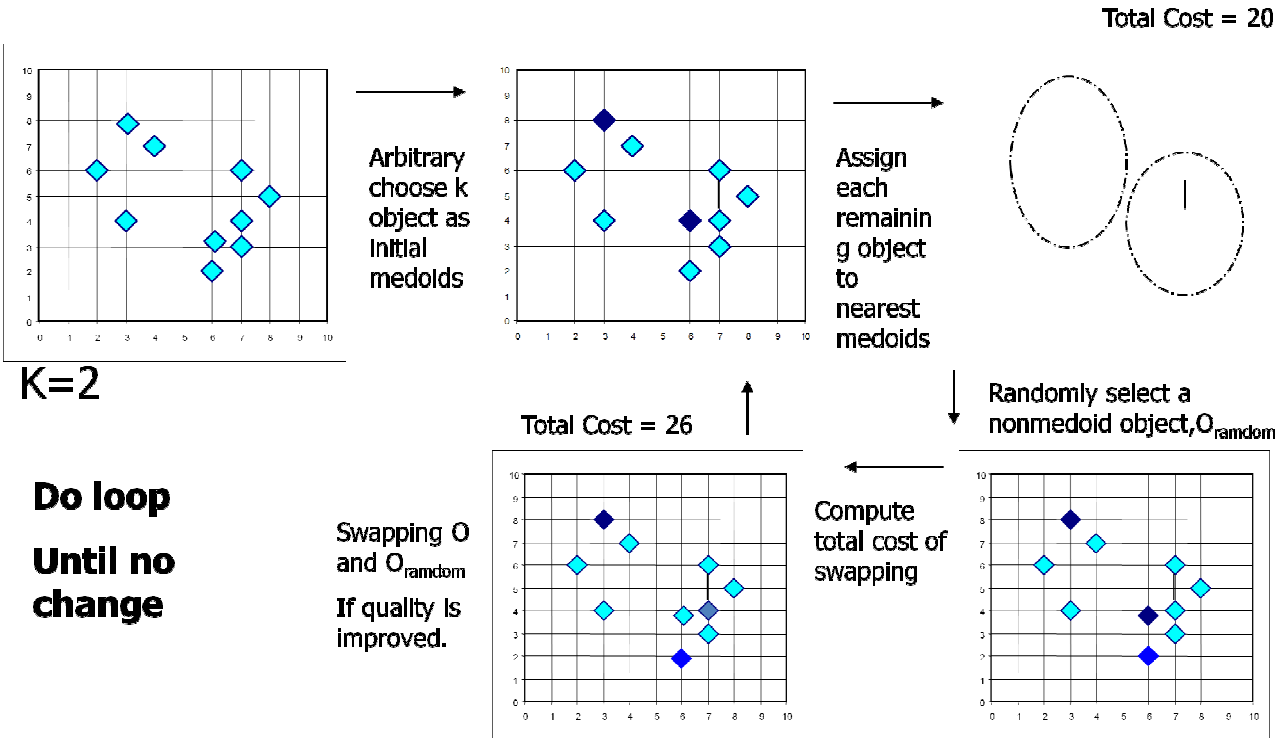


**Figure 2.3:** Partitioning around Medoids

- **Clustering large applications:** PAM works well with the small datasets but does not scalable with the large data sets. To deal with the large data sets, a sampling based method called CLARA (Clustering Large Application) can be used. CLARA, instead of considering the whole data set, CLARA uses a random sample of the data set. The sample must represent the original data set. Then, PAM algorithm is applied to compute the best medoids from the sample. CLARA builds clusterings from multiple random samples and returns the best clustering as the output [3].

**(B) Decision Trees**

Decision tree is the simplest data mining technique for classification and prediction. Decision tree considers the hierarchal structure for classification with root and the child nodes. Decision trees are drawn with the root at the top and contains the question which best classifies the coming data set. The data set enter the tree at the root and filters down till the leave nodes which exactly classifies the data set into classes. By traversing the tree we can decipher the rules and understand why the data set is classified in certain way.

**Example**

Figure 2.4 shows a decision tree for the all electronics customer to buy a computer [3]. Here each internal node depicts the attribute (or test) (age, student, credit_rating) and the leaf nodes depicts the class. In this case, there are two classes i.e. buy_computer= no or buy_computer =yes.

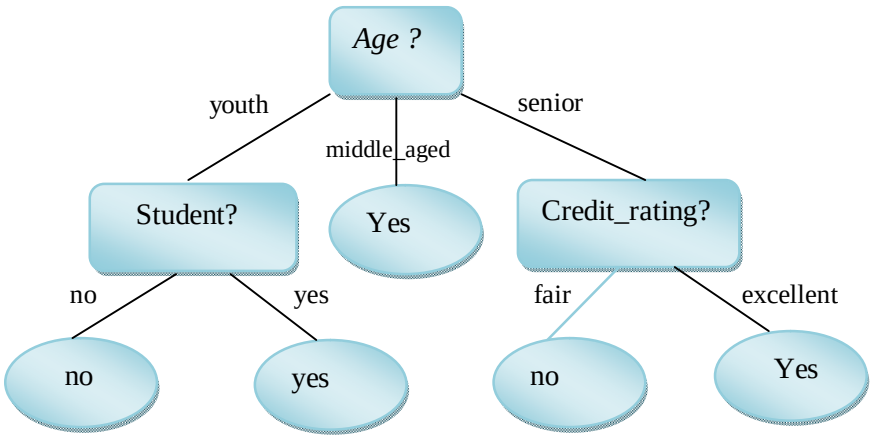


**Figure 2.4:** Decision Tree for Classification [3]

Classification using decision tree is a two steps process:

- Develop the decision tree using Decision Tree Induction.
- For each test case, use the decision tree to predict its class.

Major factors influencing the performance of decision tree are:

- Splitting attribute must be chosen carefully i.e. it should be feasible. Splitting attributes are the attributes which are used to partition the test cases into the classes.
- Sequence of splitting attributes i.e. order in which splitting attribute is selected. .
- Tree structure also affects the performance of the decision tree i.e. whether tree is balanced or unbalanced. There should be minimum levels if possible.
- Stopping criteria which defines that the creation of tree should be concluded when training data set is classified perfectly.

**(C) Memory based Reasoning**

Memory based reasoning exploits the previous known data records to predict the unknown data records. It uses the characteristics of the previously analyzed data records to classify the new records. When a new record arrives for evaluation, the algorithm finds the neighbors similar to the new record, then characteristics of neighbors are used for prediction and classification or new record [5].

**Key components of Memory based Reasoning**

- **Distance function:** Distance function calculates the distance between the incoming records and the training data set record.
- **Combination function:** Combination function combines the various distance functions results to obtain the final result.

Whenever the new record arrives, first, the distance function of the data mining tool calculates the distance between the new record and the record in the training data set[5]. The outcome of the distance function determines which record from the training set is similar to new incoming record and qualified to become the neighbour of the new record. Next, combination function of

the data mining tool is used to obtain the aggregated value of various distance functions to get the final answer.

**Example**

Consider an example of predicting the last book read by the new student based on the data set of known students. In fig 2.5, let's assume source is the student reader of different age group with the known last book read by them. Suppose a new student belongs to certain age group comes and we want to predict the last book read by him/her. By using MBR, first the distance between the new student with unknown last book read and the previous student with known last book read by them is calculated and on the basis of this, nearest neighbour is found which is used to predict the last book read by him/her.

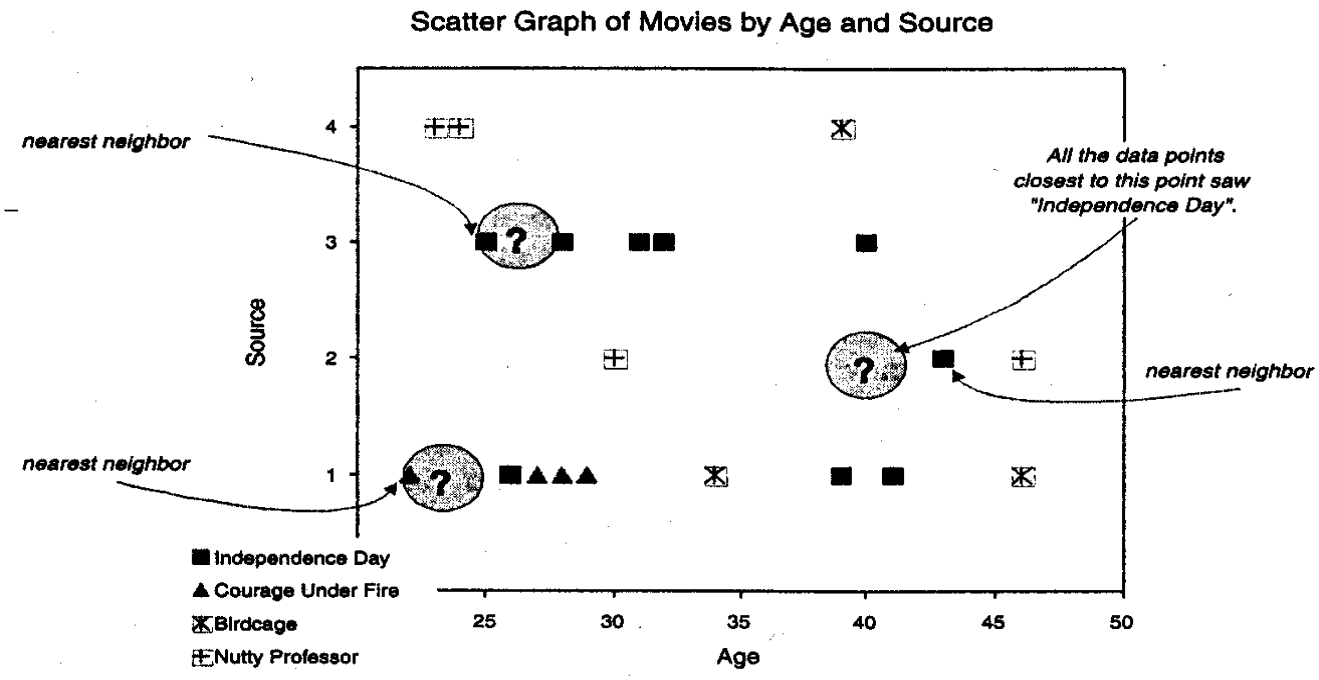


**Figure 2.5:** Memory based Reasoning (MBR)

**Issues in solving a problem with Memory based Reasoning**

- **Choosing the appropriate set of historical records**

  The historical records, represents the training set .

  The training set provides the necessary information to predict the nearest neighbour.

- **Representing the historical records**

  MBR performance in making predictions depends on how the training set is composed in the computer.

- **Determining the distance function and the combination function**

  The distance function, combination function, and number of neighbors are the key components in evaluating how good MBR is at producing results.

**(D) Frequent Pattern Mining**

Frequent patterns [3], [5] are the itemsets, subsequence or substructure that appears frequently in a data set. For example, milk and bread that are brought together appear frequently in a transaction data set is frequent itemset. A subsequence such as buying a computer, then printer and then, a digital camera, if it occurs frequently in the transactions then, it is a frequent sequential pattern [2], [3], [5]. Frequent pattern mining is used for discovering associations, correlations and many other interesting relationships among the data sets. Hence, also called as association rule mining.

A typical example of frequent pattern mining for discovering associations and correlations among items in large transactional data sets is **market basket analysis**.

Market basket analysis process mines the customer buying habits to gain insight information and find relationship between the different items that customer place in their shopping basket. By gaining the insight information about the buying habits of the customer and discovering which items are purchased frequently, it will help retailers to develop marketing strategies in order to grow their sales.

For instance, if customer is buying milk, then how likely the customer will buy bread, eggs, butter or any other item on the same trip to supermarket? This information about the buying pattern of the customer can help the retailers to take initiatives to develop some schemes to increase their sales and also, it helps the retailers to plan their shelves.

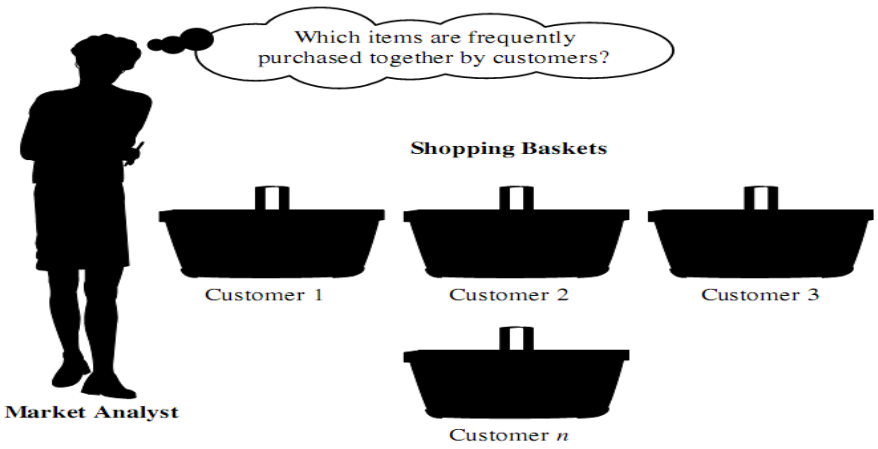


**Figure 2.6:** Market Basket Analysis

**Example**

Suppose, as the manager of an electronics shop, he wants to know the buying habits of the customer i.e. he wants to know which are the group of items often purchased by the customer in one round trip to the shop? To answer these questions market basket analysis can be performed on the database of the customer transactions done at the shop. By using the results of the market basket analysis, manager can draw some marketing strategies, endorsements or design the new layout for the shop. For instance market basket analysis may help to put the new interior layout for the shop. In one strategy, the items that are more often purchased by the customers can be put together to further increase the combined sale of these items [3]. If a customer purchases the computers also tends to buy the printer at the same time, then placing these two together may help increase the sale on both or to further increase sale may give discount on the purchase of both and also, after the purchase of computer the customer may tend to buy the antivirus, then, placing the hardware and software display together may help increase the sale of both the items.

In an alternative strategy, placing the hardware display and the software display opposite to each other may tempt the customers who wants to purchase them, to pick up other items along the way [3]. For instance, after purchasing the expensive computer and while heading towards the software display to buy anti-virus, he finds security system which he may wants to purchase them as well.

**Support and Confidence**

- In the previous example, if we think items that are available in the store as the universal set of items.
- Represent each item has the boolean variable associated with it which denotes the presence and absence of the item.
- Each basket can then be denoted by the boolean vector of values assigned to these item's boolean variables.

The boolean vector can then be analyzed to find the buying patterns which reflect items that are frequently purchased together. These patterns can then be represented in the form of association rules.

For instance, the information about the customers who purchase the computers also tend to buy the printer at the same time is represented by the following association rule:

$$Computer \Rightarrow printer\left[support = 3\,\%,\ confidence = 70\%\right] \qquad (2)$$

**Support** and **confidence** are the two factors for evaluating the association rule. In equation (2), support of 3% means that 3% of all the transactions shows that the computer and the printer are bought together. Confidence of 70% means that 70% of the customer who bought the computer also bought the printer. Association rules are considered usable if support and confidence both satisfy the minimum support threshold and minimum confidence threshold. These thresholds can be set by the domain experts and analysts.

**Formulating Association Rule**

Let IS={$is_1$,$is_2$,$is_3$….$is_n$} be the item set(a set of items is called item set). Let D be the task relevant data, be a set of database transactions where each transaction T is a nonempty item set such that T is a subset of IS. Let P be an set of items and the transaction T is said to contain P if P is the subset of T.

An association rule is generally of a implication form i.e. A=>B where set(A) is a subset of IS, set(B) is a subset of IS and A intersection B= null. The association rule A=>B has support s, if s is the percentage of transactions in D that contains set(A) union set (B) .This can also be taken as

P(A union B). The rule A=>B has confidence c, if c is the percentage of transactions in D that contain A which also contain B. This is the conditional probability, P(A/B).This can be summarized as:

$$Support(A => B) = P(A \ U \ B) \tag{3}$$

$$Confidence(A => B) = P(B / A) \tag{4}$$

An association rule that satisfy both minimum support threshold and minimum confidence threshold are referred to as **strong**.

A set of items is referred to as an **item set**. An item set that holds k items is a k-item set. The set{*computer, printer*} is a 2-item set.

The number of transactions that contain the item set is referred to as **occurrence frequency of an item set**. This is also known as **frequency**, **support count**, **count or absolute support** of the item set. Eq. (3) also called as **relative support** whereas occurrence frequency is **absolute support.** If absolute support of item set is more than **minimum support count threshold**, then, item set is frequent item set.

The eq. (4) can also be written as :-

$$\begin{aligned} Confidence(A => B) = P(B / A) &= support(A \ U \ B) / support(A) \\ Or = \ support\_count(AUB) &/ support\_count(A) \end{aligned} \tag{5}$$

Association rule mining (frequent pattern mining) can be generalized in two step process:

- **Find all frequent item sets**: Finding all the item sets that satisfy the minimum support count i.e. each item set is occurring as frequently as minimum support count.
- **Strong association rules must be generated from the frequent item set**: Association rules must satisfy minimum support threshold and minimum confidence threshold.

## 2.2 Web Mining

Web mining is the process of applying the data mining techniques to discover patterns and extract usable information from the web data i.e. web content, web structure, web usage data [1], [2], [6].

### 2.2.1 Categorization of web mining

Web mining can be categorized into three distinct categories, according to the kind of data to be mined: web content mining, web structure mining, web usage mining.

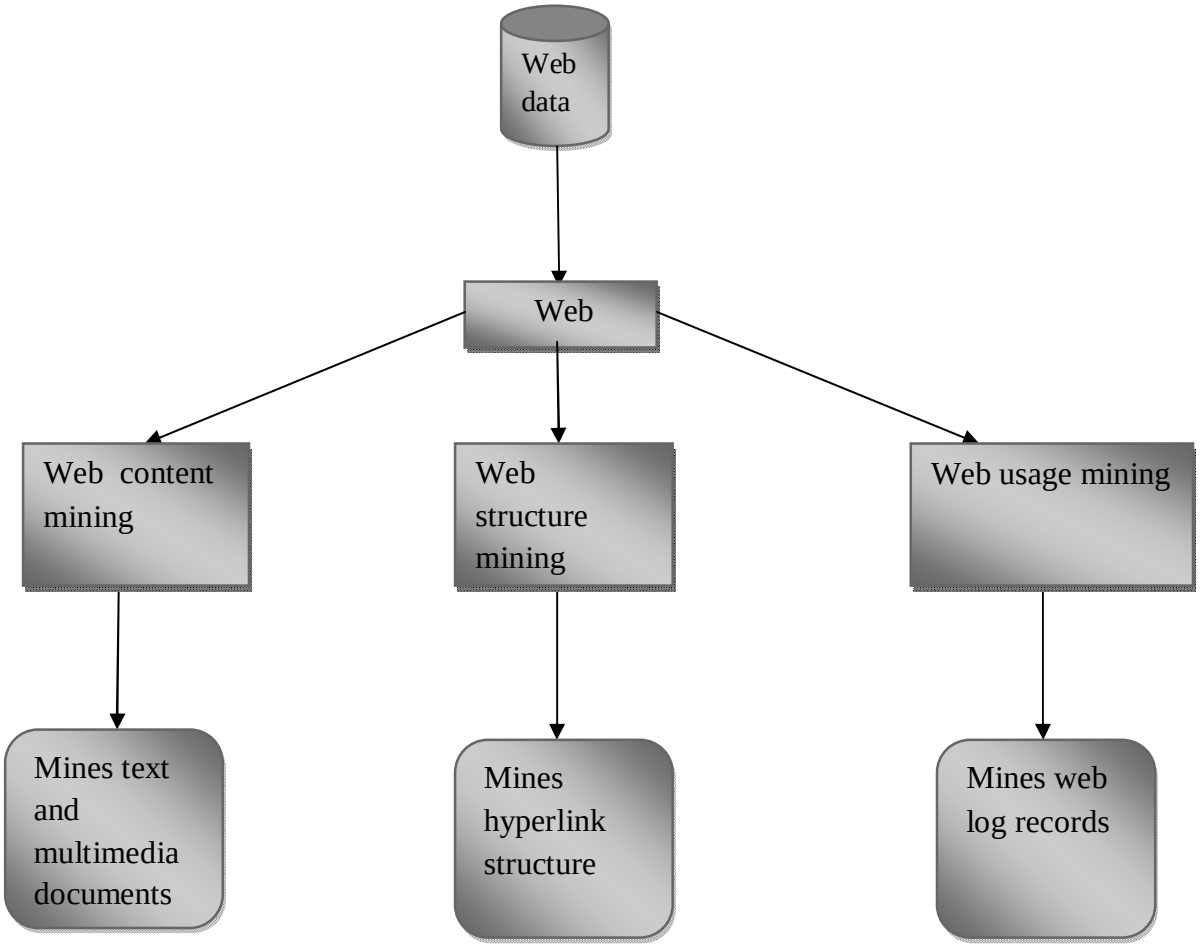


**Figure 2.7:** Web Mining Categorization

**(A)Web content mining**

Web content mining is the technique of web mining which focuses on the discovery of the usable knowledge from the web contents. Web contents correspond to the contents that are designed to

convey the information to the users. It may consist of text, images, audio, video, structured records such as lists and tables. It is also called as text mining. Web mining has many applications which spans in various areas. It provides the users an efficient mechanism to seek the information they need. Web mining is most widely used in the extracting association patterns, topic discovery, clustering of web documents and classification of web pages [1], [2].

**(B) Web structure mining**

Web structure mining is the process of discovering the structure information from the web. It studies the model that comprises of the link structures of the web. Such information are used to infer important knowledge about web pages. Hyperlinks among the web pages are usually the indicators of high relevance or good quality web pages. Using the information of structure of the web, the document retrieval can be made more efficient [1], [2]. Web structure mining can be further divided into two categories based on the kind of structural data being used.

- **Hyperlinks:** Hyperlinks are the structural units that either connects one part of the web page to another part or one web page with another web page. A hyperlink that connects one part of the web page with another part is called intra-document hyperlink. A hyperlink that connects one web page with another web page is called inter-document hyperlink [1], [2].

- **Document Structure:** The content within the web page can also be organized in a tree-structured format, based on the various HTML and XML tags within the page. Mining here focuses on automatically extracting the document object model (DOM) structures out of documents [1], [2].

**(C) Web usage mining**

Web usage mining is the process of discovering the useful pattern and hidden knowledge from the users log data i.e. users history. It is the process of finding the trajectory of the user over the web i.e. how he/she is moving over the web and what the user is searching for. Some users may be interested in finding the textual data whereas some others are interested in multimedia data. It is also called as web log mining [2]. Web usage mining involves three steps, which are as follows:

- **Preprocessing:** It involves data cleaning (filtering the unwanted data from the log file), session identification and data conversion (converting the log data into the proper format for mining).
- **Pattern discovery:** It applies the frequent pattern discovery methods to log files in order to uncover the hidden information and discovers the frequent patterns, sequences, item sets or sub-trees. For pattern discovery phase, the data must be converted to the appropriate format. For this reason, preprocessing must be done.
- **Pattern analysis:** Analyzing and understanding the result obtained in pattern discovery phase and then, drawing conclusions.

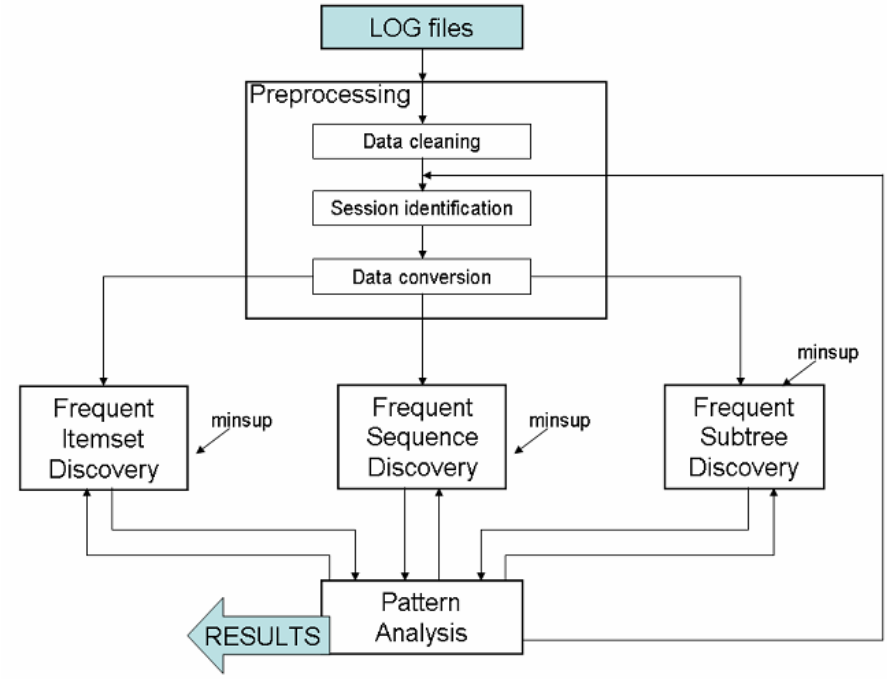


**Figure 2.8:** Process of Web Usage Mining [2]

### 2.2.2 Accomplishments of Web mining

Web mining has given genesis to many applications and algorithms. Its applications and accomplishments are as follows:-

#### (A) Page Rank Metric-Google's Page Rank Function

Page Rank is a link analysis algorithm which ranks the hypertext documents, which in turn, used to determine the quality and the relative importance of the document. It assigns numerical

weights to the hypertext documents which is called as the rank of the hypertext document. The principle of this algorithm is that a page will posses a high rank, if many highly ranked pages will point to it. So, the rank of the page depends upon the ranks of the pages pointing to it.

The rank of the web page is given by:-

$$PR(p) = d/n + (1-d)\sum_{(q,p)\in G} \frac{PR(q)}{Out\deg ree(q)} \qquad (6)$$

Here,

- n is number of web pages indexed in search engine.
- Outdegree (q) is the number of links (hyperlinks) on page q.
- The term d/n in equation (6) denotes that the user arrives at a page p by typing the url or set the page as the bookmark and select that bookmark or may have that page set as the home page.
- d is the damping factor which is the probability of the random surfer keep clicking the links. Higher the value of d (lies between 0-1), higher is the tendency that user is clicking the click.
- Term 1/n (from d/n) denotes uniform probability that a random surfer chooses the page p from the complete set of n pages on the web.
- The second term in the equation (6) user arriving at a page by traversing a link. Summation corresponds to the sum of the rank contributed by all the pages that point to the page p.

**(B) Hypertext Induced Topic Search (HITS)**

Like Page Rank algorithm, HITS also identifies the significant web pages. HITS algorithm uses the hubs and authorities to define a recursive relationship between the web pages. Hubs and authorities can be viewed as the bipartite graph. Authoritive pages contain useful information

and are supported by several links pointing to it i.e. these pages are highly referenced [1], [2]. Hubs pages are the web pages that points to authoritive pages [1], [2].

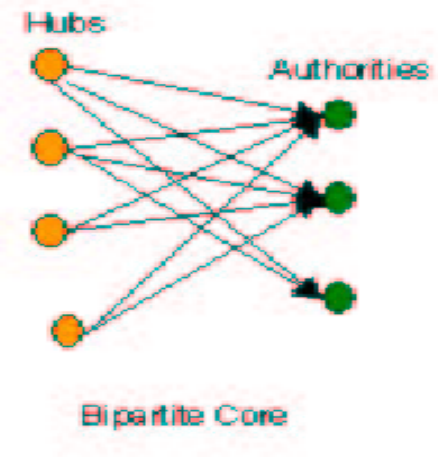


**Figure 2.9:** Hubs and Authorities Bipartite Graph [1]

The nodes on the left in the above graph represent hubs and nodes in the right represent authorities.

### 2.2.3 Applications of web mining

#### (A) Google Search Engine

Google is the most famous web search engine. It has indexed around 2.5 billion web pages on its server. It allows user to access information from these indexed web pages. It uses the web structure mining for link analysis which provides the importance of the web pages in terms of ranks called as Page rank. This makes google to work more faster, as compared to earlier search engine which concentrated on web content mining to return relevant pages to a query, and return more quality web pages based on the fired query. Working of google search engine is accomplished by three components which are as follows:-

- **Googlebot:** It is a web crawler that finds and fetches the web pages.
- **Google's Indexer:** It scraps each word from every web page and sorts them alphabetically and indexes them in its database. With each indexed word, google indexer

stores the list of web pages in which the word occurs and also, it stores the location in the index where that word occurs. This indexing scheme allows users rapid access to the web pages.

- **Query Processor:** It compares the user query with the index and recommends the relevant pages on the basis of the match and orders the page on the basis of the page rank.

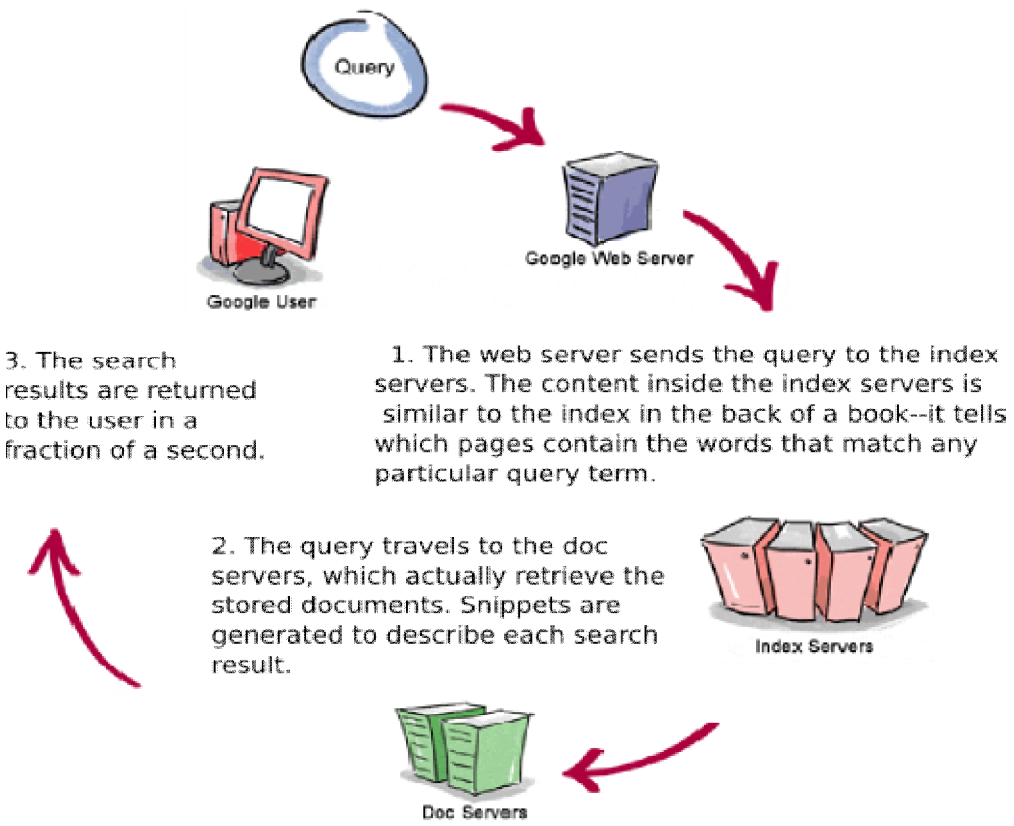


**Figure 2.10:** Working of Google Search Engine

**(B) Personalized Instant Recommendations System - Amazon**

Amazon has used the several mining techniques e.g. association rule mining, clicked path analysis, clustering web content mining, web structure mining and web log mining to gather the information about the user visiting pages over site to provide the personalized instant recommendations to the users.

**(C) Personalized Portal for the Web – My Yahoo** [1]

Yahoo provides the 'personalized portal' to a user i.e. the website will have contents personalized to the user's needs. The look and feel of the website will be according to the end user. Mining the MyYahoo usage log provides valuable information about the end user which helps to know more about the user behavior, which in turn, assist yahoo to provide user with more alluring personalized content.

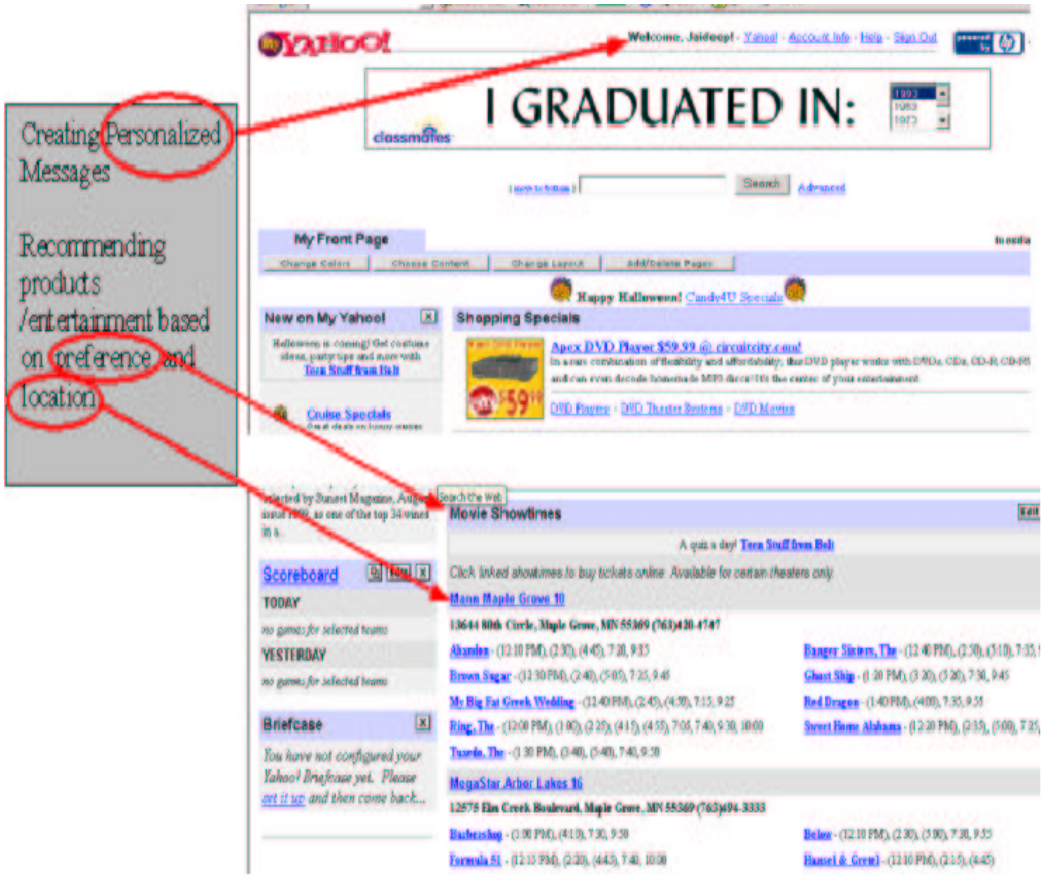


**Figure 2.11:** Personalized Web Page of MyYahoo [1]

## 2.3 Search Engine Optimization

Search engine optimization is the process of optimizing the web pages in the search result in response to a query. It optimizes the ranks of the web pages, which is used to evaluate the quality of a web page. It helps in optimizing the usability and visibility of a website to rank well in a web search engine result.

### 2.3.1 Working of Search Engine

Search engine performs the following tasks:

- **Crawling:** It is the process of fetching the web pages and storing it in a cache for later analysis. This task is performed by spider (or crawler). For example, googlebot is the crawler of google.

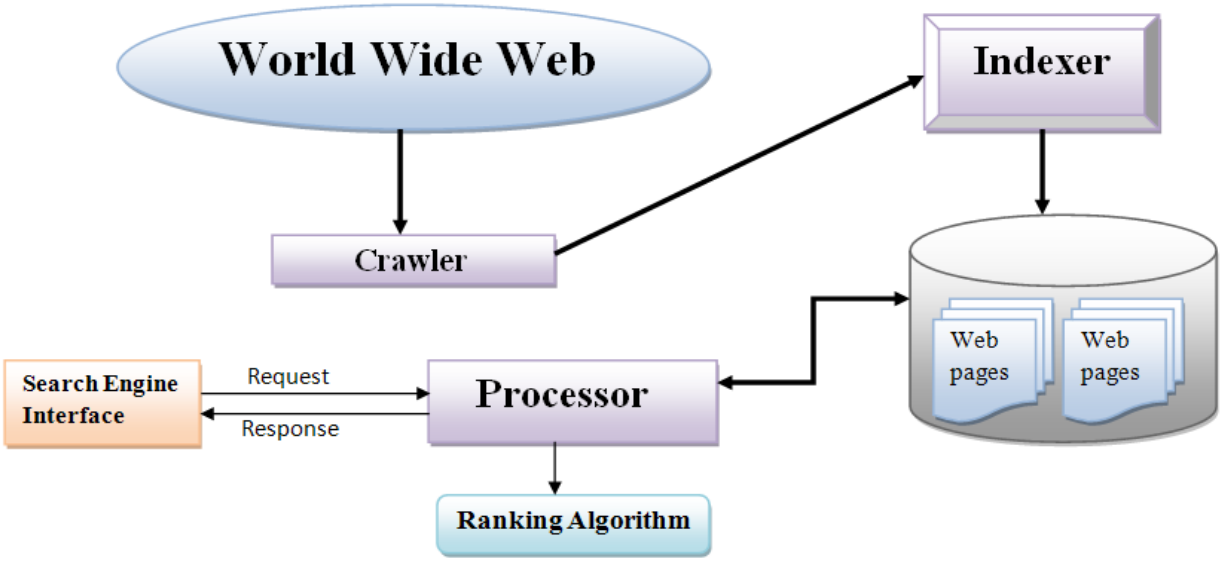


**Figure 2.12:** Search Engine Process

- **Indexing:** It is the process of indexing all the web pages available in web to a database from where it can be retrieved later. Also, this process helps in identifying the keywords and expressions that best describes the page and index the page with these keywords and expressions.
- **Processing:** In this phase, search engine processes the search request. It compares the contents of the fired query with the indexed pages in database.
- **Calculate Relevancy:** It calculates the rank of the page which determines the relevancy of the page with the fired query. The search results are returned in the decreasing order of the rank.

### 2.3.2 Search Engine Optimization Techniques

Search engine optimization techniques are classified into two categories:

- **White Hat SEO:** White Hat SEO technique is a technique that search engine recommends as a good design and are according to the guidelines of the search engine.

This technique involves no deceiving content. This technique ensures that the content present over web page is meant for the users not for search engine.

- **Black Hat SEO:** Black Hat SEO technique is a technique which search engines do not recommend as a good design. It is also called as spamdexing. Black Hat SEO technique redirects the users from a page ranked in search engine to a different page which is not listed in the search engine database. It serves one version of web pages to a search engine spider bots and another version to the users. It does **meta tag stuffing** which means using repeated keywords in meta tag. Using keywords which are not related to the site content called as **keyword stuffing** which is practiced in this technique. It produces **doorway pages** in which low quality web pages i.e. which contain very little content, are populated with very similar and famous keywords in order to rank it well in search engine. It also does **page hijacking**. Page hijacking is the process which creates the copy of a popular website and shows its contents to the web crawler but redirects the surfer to the malicious or unrelated website when surfer clicks its link.

## 2.4 Algorithms and Formulas Reviewed

In this section, some of the algorithms and formulas are discussed. The algorithms such as K-means clustering algorithm, frequent itemset mining algorithm, page rank algorithm and information scent metric are explained here. These algorithms and formulas are used in the proposed methodology.

### 2.4.1 K-means Clustering Algorithm

K-means clustering [3], [5] is the centroid based partitioning technique. It considers the centroid of a cluster to form clusters. Centroid is the center point of the cluster which represents the mean value and the cluster is evolved around centriod. K-means algorithm is given as follows:

**Input**

- **C**: the number of clusters.
- **D**: a data set comprising of n objects.

**Output**

- A set of C clusters.

**Method**

**(1)** Choose randomly C objects as initial clusters.

**(2) Repeat Loop**

**(3)** Assign or reassign each object to the cluster to which it is the most similar based on the centroid of the cluster and the similarity measure.

**(4)** After (re)assigning new objects to a cluster, recalculate the mean value of each cluster.

**(5) End loop**

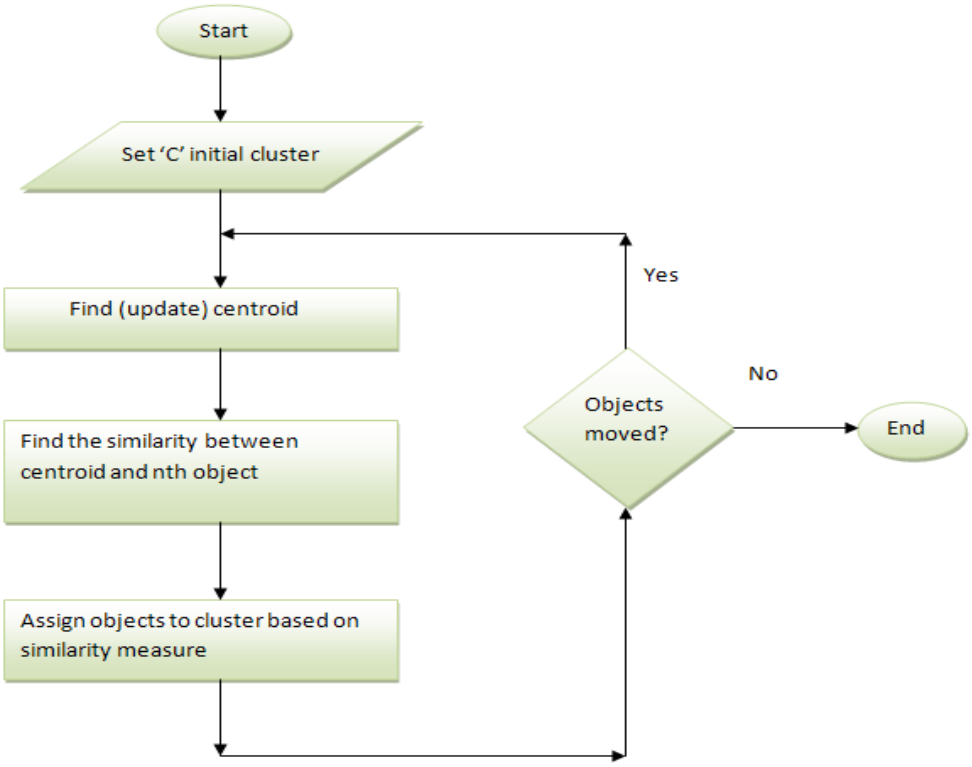


**Figure 2.13:** K-means Clustering Flowchart

**K-means: Step-By-Step Numerical Example**

Consider the table 2.1 data set consisting of the scores of two individuals 'P' and 'Q'.

| Subject | P | Q |
|---|---|---|
| 1 | 1.0 | 1.0 |
| 2 | 1.5 | 2.0 |
| 3 | 3.0 | 4.0 |
| 4 | 5.0 | 7.0 |
| 5 | 3.5 | 5.0 |
| 6 | 4.5 | 5.0 |
| 7 | 3.5 | 4.5 |

**Table 2.1:** Two Individual Scores used For K-means clustering

Let's randomly pick two initial clusters say, cluster 1 and cluster 2:

| | Subject | Mean Vector (centroid) |
|---|---|---|
| **Cluster 1** | 1 | (1.0, 1.0) |
| **Cluster 2** | 4 | (5.0, 7.0) |

**Table 2.2:** Randomly Picked Clusters Formed

The remaining subjects are now evaluated and are assigned to the clusters to which it is most similar which is measured by calculating their Euclidean distance. The mean value (i.e. centroid) is recalculated for each cluster. This step is performed in iteration until clusters centroid becomes stable. This leads to the following series of steps:

| | Cluster 1 | | Cluster 2 | |
|---|---|---|---|---|
| **Step** | **Subject** | **Mean Vector (centroid)** | **Subject** | **Mean Vector (centroid)** |
| 1 | 1 | (1.0, 1.0) | 4 | (5.0, 7.0) |
| 2 | 1, 2 | (1.2, 1.5) | 4 | (5.0, 7.0) |
| 3 | 1, 2, 3 | (1.8, 2.3) | 4 | (5.0, 7.0) |
| 4 | 1, 2, 3 | (1.8, 2.3) | 4, 5 | (4.2, 6.0) |
| 5 | 1, 2, 3 | (1.8, 2.3) | 4, 5, 6 | (4.3, 5.7) |
| 6 | 1, 2, 3 | (1.8, 2.3) | 4, 5, 6, 7 | (4.1, 5.4) |

**Table 2.3:** Cluster Analysis Steps

The two clusters at this stage have the following characteristics:

| | **Subjects** | **Mean Vector (centroid)** |
|---|---|---|
| **Cluster 1** | 1, 2, 3 | (1.8, 2.3) |
| **Cluster 2** | 4, 5, 6, 7 | (4.1, 5.4) |

**Table 2.4:** Cluster Formed

Now, even after assignment of the subjects to the cluster 1 & 2, we cannot say for sure the subjects are assigned to the right clusters. Now, we calculate distance of each subject's score with its cluster mean value score and also, with its opposite cluster mean value score. And, we get following values:

| **Subject** | **Distance to mean (centroid) of Cluster 1** | **Distance to mean (centroid) of Cluster 2** |
|---|---|---|
| 1 | 1.5 | 5.4 |
| 2 | 0.4 | 4.3 |
| 3 | 2.1 | 1.8 |
| 4 | 5.7 | 1.8 |
| 5 | 3.2 | 0.7 |
| 6 | 3.8 | 0.6 |
| 7 | 2.8 | 1.1 |

**Table 2.5:** Distance to Centroid of Clusters

Here, in table 2.6, subject 3 which belongs to cluster 1 is more nearer to mean value score of the opposite cluster (cluster 2). Since, each subject's distance to its own cluster mean value should be less than the subject's distance to the opposite cluster mean value score. Hence, subject 3 will be reshuffled to cluster 2 in new partitioning:

| | Individual | Mean Vector (centroid) |
|---|---|---|
| **Cluster 1** | 1, 2 | (1.3, 1.5) |
| **Cluster 2** | 3, 4, 5, 6, 7 | (3.9, 5.1) |

**Table 2.6:** Final Clusters

### 2.4.2 Frequent Itemset Mining Algorithm

The most effective algorithm for mining frequent itemset is **Apriori algorithm.** Apriori algorithm [3] is used for mining frequent itemsets for boolean association rules. The algorithm uses the prior knowledge of frequent itemset properties. Hence, called Apriori algorithm. It is an iterative approach which does level wise search, where n-itemset is used to evaluate n+1 itemsets. In this algorithm, first, frequent 1-itemsets (frequent itemset containing only one element) is found by traversing the dataset and keeping the count of each item, and collecting only those items that satisfy the minimum support count. Let's say resultant set is F1. Next, F1 is used to find the frequent 2-itemset (frequent itemset containing only one element), say F2 and so on, until no more frequent k-itemset is found.

An **Apriori property** [3] is used to improve the efficiency of generation of frequent itemsets which is :

*All nonempty subsets of a frequent itemset must also be frequent*

**Apriori Algorithm**

**Input:**

- A dataset of transactions say 'D'.
- Minimum support count say *min_sup*.

**Output:**

- Frequent itemsets 'L' in dataset of transactions 'D'

**Method:**

```
(1)   L1 = find_frequent_1-itemsets(D);
(2)   for (k = 2; Lk−1 ≠ φ; k++) {
(3)       Ck = apriori_gen(Lk−1);
(4)       for each transaction t ∈ D { // scan D for counts
(5)           Ct = subset(Ck, t); // get the subsets of t that are candidates
(6)           for each candidate c ∈ Ct
(7)               c.count++;
(8)       }
(9)       Lk = {c ∈ Ck | c.count ≥ min_sup}
(10)  }
(11)  return L = ∪k Lk;
```

```
procedure apriori_gen(Lk−1:frequent (k − 1)-itemsets)
(1)   for each itemset l1 ∈ Lk−1
(2)       for each itemset l2 ∈ Lk−1
(3)           if (l1[1] = l2[1]) ∧ (l1[2] = l2[2])
                  ∧... ∧ (l1[k − 2] = l2[k − 2]) ∧ (l1[k − 1] < l2[k − 1]) then {
(4)               c = l1 ⋈ l2; // join step: generate candidates
(5)               if has_infrequent_subset(c, Lk−1) then
(6)                   delete c; // prune step: remove unfruitful candidate
(7)               else add c to Ck;
(8)           }
(9)   return Ck;
```

```
procedure has_infrequent_subset(c: candidate k-itemset;
          Lk−1: frequent (k − 1)-itemsets); // use prior knowledge
(1)   for each (k − 1)-subset s of c
(2)       if s ∉ Lk−1 then
(3)           return TRUE;
(4)   return FALSE;
```

**Example**

Consider the following transaction table of the departmental store.

| Transaction Id | List of Items Purchased Id |
|---|---|
| **T1** | { $IS_1$, $IS_2$ } |
| **T2** | { $IS_3$, $IS_4$ } |
| **T3** | { $IS_2$, $IS_3$, $IS_5$, $IS_4$ } |
| **T4** | { $IS_1$, $IS_2$, $IS_3$, $IS_5$ } |
| **T5** | { $IS_3$, $IS_5$, $IS_4$ } |

| | |
|---|---|
| **T6** | { $IS_2$, $IS_3$ } |

**Table 2.7:** Transaction Table of Departmental Store

**Step 1:** Scan all the transactions and collect the count for 1-itemsets (i.e. set containing only one item) in the transactions.

| Itemset | Support_count |
|---|---|
| **$IS_1$** | 2 |
| **$IS_2$** | 4 |
| **$IS_3$** | 5 |
| **$IS_4$** | 3 |
| **$IS_5$** | 3 |

**Table 2.8:** 1-Itemset Table with Support Count

**Step 2:** Now, prune the itemsets generated in step 1 by comparing their count with the minimum support count say, minimum support count is 2. Table 2.9 will remain same after pruning since all the itemset's support count is greater than equal to 2.

| Itemset | Support_count |
|---|---|
| **$IS_1$** | 2 |
| **$IS_2$** | 4 |
| **$IS_3$** | 5 |
| **$IS_4$** | 3 |
| **$IS_5$** | 3 |

**Table 2.9:** Pruned 1-Itemset Table

**Step 3:** In this step, generate a 2-itemsets by self joining the itemsets gained in step 2 and collect their count by scanning Table 2.9.

| Itemset | Support_count |
|---|---|
| **$IS_1$ , $IS_2$** | 2 |
| **$IS_1$ , $IS_3$** | 1 |
| **$IS_1$ , $IS_4$** | 0 |

| | |
|---|---|
| **$IS_1$, $IS_5$** | 1 |
| **$IS_2$, $IS_3$** | 3 |
| **$IS_2$, $IS_4$** | 1 |
| **$IS_2$, $IS_5$** | 2 |
| **$IS_3$, $IS_4$** | 3 |
| **$IS_3$, $IS_5$** | 3 |
| **$IS_4$, $IS_5$** | 2 |

**Table 2.10:** 2-Itemset Table with Support Count

**Step 4:** Prune the itemsets generated in the step 3 by comparing it with the minimum support count which is 2.

| Itemset | Support_count |
|---|---|
| **$IS_1$, $IS_2$** | 2 |
| **$IS_2$, $IS_3$** | 3 |
| **$IS_2$, $IS_5$** | 2 |
| **$IS_3$, $IS_4$** | 3 |
| **$IS_3$, $IS_5$** | 3 |
| **$IS_4$, $IS_5$** | 2 |

**Table 2.11:** Pruned 2-Itemset Table

**Step 5:** Generate a 3-itemsets by self joining the itemsets gained in step 4 and collect their count by scanning Table 2.11.

| Itemset | Support_count |
|---|---|
| **$IS_2$, $IS_3$, $IS_5$** | 2 |
| **$IS_3$, $IS_4$, $IS_5$** | 2 |

**Table 2.12:** 3-Itemset Table with Support Count

**Step 6:** Prune itemsets generated in step 5 by comparing their count with minimum support count. We will get the same table as Table 2.12.

| Itemset | Support_count |
|---|---|
| **$IS_2$, $IS_3$, $IS_5$** | 2 |
| **$IS_3$, $IS_4$, $IS_5$** | 2 |

**Table 2.13:** Pruned 3-Itemset Table

**Step 6:** Generate a 4-itemsets by self joining the itemsets gained in step 5. We will get only one 4-itemset which is $\{ IS_2, IS_3, IS_4, IS_5 \}$ . This 4-itemset will be pruned according to the **Apriori Property** as discussed earlier. Since the subset of 4-itemset is not frequent itemset, hence , it's superset will not be frequent itemset. Hence, we get two frequent itemset as listed in Table 2.14 i.e. $\{ IS_2, IS_3, IS_5 \}$, $\{ IS_3, IS_4, IS_5 \}$.

**Generating Association Rule from Frequent Itemset**

It is very easy to generate the strong association rules, once we have found the frequent itemset. Association rule can be easily generated from frequent itemset. We will find association rule by finding confidence by using the frequent itemset which is given by:-

$$Confidence(A => B) = P(B/A) = \frac{\sup port\_count(AUB)}{\sup port\_count(a)} \quad (7)$$

Here,

SUPPORT_COUNT(AUB) is the number of transactions containing (AUB) itmesets and SUPPORT_COUNT(A) is the number of transactions containing itemset(A) [3]. Based on this equation, association rule can be generated as follows [3] :-

- For each frequent itemset, say 'f', generate all non-empty subsets, say 's' of 'f'.
- For every non-empty subset s of 'f', output the rule "s=>f-s" if $\frac{\text{SUPPORT\_COUNT(f)}}{\text{SUPPORT\_COUNT(s)}} \geq$ minimum confidence threshold.

**Example**

We will generate association rule by finding confidence by using frequent itemset. Consider frequent itemset generated in previous section i.e. $\{ IS_2, IS_3, IS_5 \}$.

| A=>B | Confidence(A=>B) |
|---|---|
| **$IS_2$,$IS_3$=>$IS_5$** | 2/3=66.6% |
| **$IS_2$, $IS_5$=> $IS_3$** | 2/2=100% |
| **$IS_3$, $IS_5$=> $IS_2$** | 2/3=66.6% |
| **$IS_2$=>$IS_3$,$IS_5$** | 2/4=50% |
| **$IS_3$=> $IS_5$, $IS_2$** | 2/5=40% |
| **$IS_5$=> $IS_2$,$IS_3$** | 2/3=66.6% |

**Table 2.14:** Association Rule Table

If minimum confidence threshold is, say 80%, then only second rule will the strong association rule. The rule confidence ($IS_2$, $IS_5$=> $IS_3$) = 100% means that 100% of the customer who have bought items $IS_2$, $IS_5$, have also bought $IS_3$.

### 2.4.3 Page Rank

As discussed earlier in section 2.2.2.1, Page Rank [1] is a link analysis algorithm which ranks the hypertext documents, which in turn, used to determine the quality and the relative importance of the document. It is given by :-

$$PR(p) = d / n + (1 - d) \sum_{(q,p) \in G} \frac{PR(q)}{Out\deg ree(q)} \quad (8)$$

**Working of PageRank Algorithm**

Consider the fig 2.14

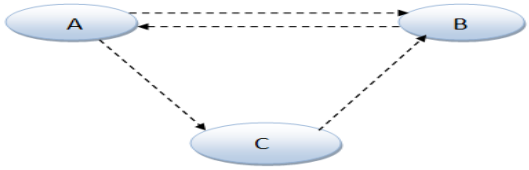


**Figure 2.14:** Hyperlink structure for three pages

We assume, here, initially the page rank of each page is 1 and d=0.85 as suggested by Larry Page and Sergey Brin ( inventors of Page Rank algorithm).

Finding page ranks of each page:

Iteration {

PR(A)=(1-0.85) + PR(B)/OUTDEGREE(B) = 0.15 + 1/1 = 1.15

PR(B)=(1-0.85)+PR(A)/ OUTDEGREE(A)+ PR(C)/OUTDEGREE(C)=0.15 + 1.15/2+1=1.725

PR(C)=(1-0.85)+ PR(A)/ OUTDEGREE(A)=0.15+1.15/2=0.725

}

This above process will keep on iterating until the ranks of the page won't get stabled.

### 2.4.4 Information Scent Metric

Web consists of bags of information where user searches for information by navigating from one page to another. Their surfing pattern is guided by their information need. Information scent is the measure by which we can determine whether the page is satisfying the information need or not i.e. the user is getting the information that he/she is looking for or not. If the user has got the information in the clicked page, then, that page reflects strong information scent, otherwise, weak.

Information scent metric is used to assign weight to each clicked page (visited page) by inferring user information need.

There are two factors used to determine information scent metric:

- **Page access weight (PF.IPF).**
- **Time.**

Each of these factors is used to quantify the information scent associated with the page. First factor is page access (PF.IPF) where PF is the access frequency of the clicked page in given query session and IPF is the ratio of total query sessions in log file to the number of query

sessions in which this page is clicked [11]. The second factor is the time spent on a page in a given query session.

Information scent metric helps in finding those pages with which strong information scent is associated i.e. page is more relevant to the user's information need.

Information scent metric is given in [11] which is :

$$I_{sc} = PF.IPF\left(P_{id}\right) * Time\left(P_{id}\right) \tag{9}$$

$$PF.IPF\ \left(P_{id}\right) = \frac{f(Pid)}{\max(f(pid))} * log(\frac{Q}{q(Pid)}) \tag{10}$$

PF is the normalized frequency $f(P_{id})$ of a page $P_{id}$ in a given query session. IPF is the ratio of the total number of query sessions Q in the whole log file to the number of query sessions $q(P_{id})$ in which given page $P_{id}$ is clicked.

$Time(P_{id})$ is the ratio of time spent on the page $P_{id}$ in a given query session to the total duration of the query session.

### 2.4.5 Measuring Similarity and Dissimilarity of Data

There are two measures used for measuring proximity (closeness): Similarity and Dissimilarity.

- **Similarity:** Similarity measure between two objects say, i and j finds the degree of closeness between them. Similarity measure will return value 0, if two objects are unlike and return 1, if objects are identical. Higher the value of similarity measure, more the two objects are identical.
- **Dissimilarity:** Dissimilarity measure between two objects say, i and j finds the degree of dissimilarity between them. Dissimilarity measure will return value 0, if two objects are like and return 1, if objects are unlike. Higher the value of dissimilarity measure, more the objects are un-identical.

There are two data structures used for finding similarity and dissimilarity measure: Data matrix and Dissimilarity matrix.

- **Data Matrix(or object-by-attribute matrix):** Data matrix is used to store data multiple attributes objects. Suppose, we have n objects (such as person, house, occupation etc) and their attributes represented by p(such as age, height or age) . Then, objects will be represented by x[i][j], where i denotes $i^{th}$ object and j denotes its $j^{th}$ attribute.

| | | | |
|---|---|---|---|
| **x11 ….** | **x12 ……** | **x13 …….** | **x1p** |
| **x21 ….** | **X22 ……** | **x23 …….** | **x2p** |
| **x31 ….** | **x32 ……** | **x33 …….** | **x3p** |
| **x41 ….** | **x42 ……** | **x43 …….** | **x4p** |
| **xn1 ….** | **xn2 ……** | **xn3 …….** | **xnp** |

**Figure 2.15:** Data matrix( 'n' object -x- 'p' attributes)

- **Dissimilarity Matrix (or object-by-object structure):** This matrix stores dissimilarity values for pairs of objects.

| | | | |
|---|---|---|---|
| **0** | | | |
| **D(2,1)** | **0** | | |
| **D(3,1)** | **D(3,2)** | **0** | |
| **D(n,1)** | **D(n,2)** | **D(n,3)** | **0** |

**Figure 2.16:** Dissimilarity matrix

In this matrix, D(i,j) represents the difference between objects i and j. If D(i,j) is a non-negative number and close to zero, then, objects i and j are similar to each other, otherwise, dissimilar.

Similarity [3] can be expressed in terms of dissimilarity measure as:

$$sim(i, j) = 1 - D(i, j) \tag{11}$$

Where sim(i,j) is the similarity between objects i and j.

**Proximity Measure for Nominal Attributes**

Nominal attribute is an attribute which can take more than one state. For example, subject can be math, science, English etc.

The dissimilarity [3] between two objects i and j can be given by:

$$D(i, j) = \frac{p - m}{p} \tag{12}$$

Where m is the number of matches i.e. attributes for which i and j are in the same state and p is the total number of attributes of an object.

**Example**

**Proximity measure for Numeric Attributes**

Numeric attributes are those attributes which contains a numeric value. Dissimilarity between numeric objects is given by Euclidean distance and Manhattan distance.

Euclidean distance is the most popular distance measure and is given by:

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + ..... + (x_{ip} - x_{jp})^2} \tag{13}$$

Where i object is given by xip and j object is represented by xjp, p is the total number of attributes.

Manhattan distance is given by:

$$d(i, j) = \left|x_{i1} - x_{j1}\right| + \left|x_{i2} - x_{j2}\right| + ..... + \left|x_{ip} - x_{jp}\right| \tag{14}$$

**Cosine Similarity**

Document normally consists of thousands of word(or attributes). In such cases, document is represented by the term-frequency vector or document vector. Term frequency vector is a matrix

which keeps track of the count of each term in the document which is called as occurrence frequency of a term.

| Document | Data | Retrieval | Search | Web | Mining |
|---|---|---|---|---|---|
| **Document 1** | 10 | 6 | 5 | 20 | 15 |
| **Document 2** | 7 | 2 | 0 | 15 | 13 |
| **Document 3** | 23 | 0 | 17 | 21 | 12 |

**Table 2.15:** Term Frequency Vector Table

Table 2.15 is the term frequency vector in which each term that appears in the doument are used as attributes and their corresponding counts are collected. In dcoument 1, data appears ten times while retrieval appears six times and so on.

Such term frequency vectors are normally very large and sparse (i.e. they may have large number of zero enteries in vector). Traditional distance measures do not work well for such vectors.For example, two documents may have a term whose count is zero that means these two documents do not share this variable and do not make them similar but traditional distance measure will report the similar if count is zero for particular term in both documents. Hence, cosine similarity is used for such vectors which give very effective results.

Cosine similarity [3] is a measure of similarity which is used to compare two documents or ranks the document with respect to a given vector of query words.Let ‘p’ and ‘q’ be two vectors for comparison. Cosine similarity is given by:

$$sim(p,q) = \frac{p.q}{\|p\|\|q\|} \tag{15}$$

where ||p|| is the Euclidean norm of vector p=(p1,p2,p3……,pn) defined as $\sqrt{p1^2 + p2^2 + \cdots pn^2}$ .Similarly, ||q|| is the Euclidean norm of vector q.The measure computes the cosine angle between p and q. If value is 0, then, its cosine angle is 90 degrees between two vectors which means two vectors have no similarity. If value is closer to 1, then, angle between two vectors are smaller and they have higher similarity.

**Example**

Consider two vectors from Table 2.15 ,document 1 say ‘p’ and document 2 say ‘q’.

p=(10,6,5,20,15) and q=(7,2,0,15,13)

p.q=10x7 + 6x2 + 5x0 + 20x15 + 15x13=377

$||p||=\sqrt{10^2 + 6^2 + 5^2 + 20^2 + 15^2}= 28.03$

$||q||=\sqrt{7^2 + 2^2 + 0^2 + 15^2 + 13^2}= 21.14$

$$\text{sim(p,q)} = \frac{377}{28.03 \text{ x } 21.14} = 0.63$$

### 2.4.6 Data Normalization

Data normalization is the process in which the value of data can be made to fall in a certain range. Data normalization also known as standardization. Normalization helps to avoid dependency on the measurement units. For example, changing measurement units from meters to kms for road or from kilograms to pounds for weight may lead to different result, hence, in such cases normalizing the value within a certain range makes the value to be independent of any unit.

Normalization is particularly useful in classification algorithms such as neural network, clustering, and nearest-neighbor classification. For distance based methods such as clustering or nearest-neighbor, normalization prevents attributes with large values(say income) from overriding smaller ranges values(say binary attributes) .

Method for data normalzation: min-max normalization, z-score normalization, normalization by decimal scaling .

- **Min-max Normalization:** It maps the value from one range to another range. It is given by:

$$\frac{v_i - \min_A}{\max_A - \min_A}(new\_\max_A - new\_\min_A) + new\_\min_A \qquad (16)$$

Here,

$min_A$ is a minimum range value of a given value $v_i$ and $max_A$ is the maximum range value of a given value $v_i$.

$new\_min_A$ is new minimum range value for a given value and $new\_max_A$ is new maximum range value for a given value $v_i$.

**Example**

Suppose minimum and maximum value for an salary attribute are 60,000 and 80,000 respectively. We want to map a salary 65,000 in the range [0.0-1.0]

$$v_i' = \frac{65000-60000}{80000-65000}(1.0-0.0) + 0.0 = 0.333$$

The value 65000 is mapped to 0.333 in the new range[1.0-0.0]

- **Z-score Normalization:** A value can be normalized based on the mean and standard deviation of the given value.

$$v_i' = \frac{v_i - \bar{A}}{\sigma_A} \quad (17)$$

- **Normalization by Decimal Scaling:** In this method, value is normalized by moving the decimal point. The number of decimal points moved depends on the absolute value of the given value.

$$v_i = \frac{v_i}{10^j} \quad (18)$$

# Chapter 3: Information Foraging

## 3.1 Information Foraging

Web is overwhelmed with the ample amount of information which is increasing with an astonishing speed. Retrieving relevant information from the web in minimum is tedious task. Information foraging aims at forage for relevant information from the web in minimum time.

### 3.1.1 Definition of Information Foraging

Information foraging [14] is the theory that applies the concept of optimal foraging theory to the idea about how human users search for information. The theory is based on the assumption that, when humans searching for information, humans use "built-in" foraging mechanisms in order to forage as much as relevant information as possible like animals find food and grab as much as possible. The theory assumes that user, when possible, will modify their strategies or the structure of the environment to maximize their rate of gaining valuable information [8].

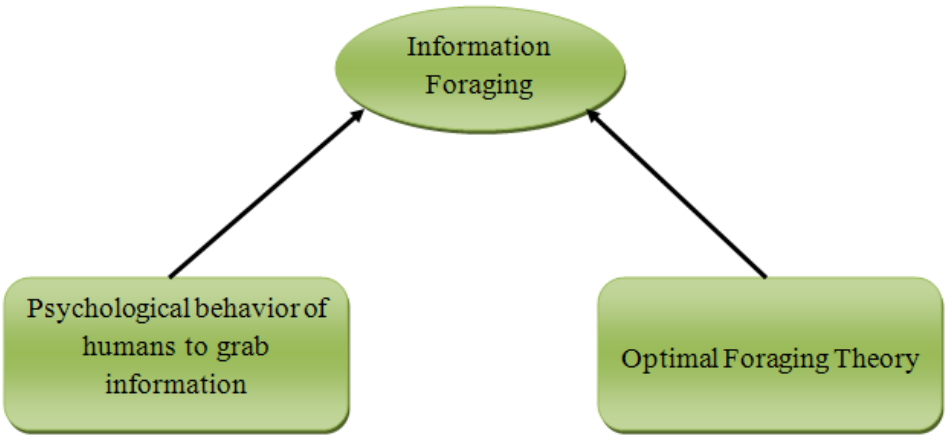


**Figure 3.1:** Information Foraging

There is a similarity between the user's information seeking patterns and animal food foraging strategies. Information seekers use the same strategies as used by animals to seek information

over web while navigating through links in order to seek as much as relevant information as possible.

### 3.1.2 Details of Theory of Information Foraging

- **Informavores:** Organisms that consumes information. Informavores constantly makes decision on what kind of information to look for, whether to stay at the current site to try to find additional information or whether they should move to the other site, which path or link to follow to the next information site.

- In **optimal foraging theory**, our emphasis is on maximizing the energy per unit time.

$$Optimal\ Foraging\ =\ maximize[\frac{ENERGY}{TIME}] \tag{19}$$

- In the same way, in **information foraging theory**, our emphasis is on maximizing the useful information per unit time

$$Information\ Foraging\ =\ maximize[\frac{USEFULINFORMATION}{TIME}] \tag{20}$$

- **Information Scent:** Information scent is a part of information foraging theory. It refers to the extent that users can predict what they will find pursuing a certain path through a website or other source of information.

## 3.2 Information Scent

It is the most pivotal concept in information foraging. Like animals, rely on the scents which indicate the possibility of finding the prey and guides them to the promising patches, in the same way web surfers rely on the various cues on the links in the web search area. Their surfing patterns are guided by these cues and their information needs.

Information scent refers to the term which is used to indicate how users will evaluate the links option they have, when they are looking for the information on the web [8], [14]. It is the

measure which tells whether the user who is surfing over the web finds the information that he/she looking for while clicking various links. It is of two types which are as follows:-

- **Strong information scent:** It refers to the measure which tells that, when user presented with the list of option which option they will select which gives the clearest indication of the information they are looking for and whether they find the required information after clicking this option.
- **Weak information scent:** It refers to the measure which evaluates the links which does not provide the clear indication of the user information need .Hence confuses the user which link to follow.

**Example**

Suppose following are the navigation options provided to user in the online shopping site

| ---------Select options-------- |
|---|
| **Audio and TV** |
| **Books** |
| **Computing** |
| **Fashion** |
| **Furniture** |
| **Gardening** |

**Table 3.1:** Select option for evaluating information scent

If user wants to look for information about i-pods which option he/she will choose here. Here, all the navigation options are clearly understandable, but which option user will select to gain information on i-pods? Since i-pods can be listed in either Audio and TV, or computing section. In this case, user is more likely to select the wrong option which makes the user get frustrated on not finding information. The user backtrack the links or eventually leave the site. This is referred to as weak information scent.

### 3.2.1 Information Scent Metric

Web consists of bags of information where user searches for information by navigating from one page to another. Their surfing pattern is guided by their information need. Information scent is the measure by which we can determine whether the page is satisfying the information need or not i.e. the user is getting the information that he/she is looking for or not. If the user has got the information in the clicked page, then, that page reflects strong information scent, otherwise, weak.

Information scent metric is used to assign weight to each clicked page (visited page) by inferring user information need.

There are two factors used to determine information scent metric:

- **Page access weight (PF.IPF).**
- **Time.**

Each of these factors are used to quantify the information scent associated with the page. First factor is page access (PF.IPF) where PF is the access frequency of the clicked page in given query session and IPF is the ratio of total query sessions in log file to the number of query sessions in which this page is clicked [11]. The second factor is the time spent on a page in a given query session.

Information scent metric helps in finding those pages with which strong information scent is associated i.e. page is more relevant to the user's information need.

Information scent metric is given in [11] which is:

$$I_{sc} = PF.IPF\ (P_{id}) * Time\ (P_{id}) \tag{21}$$

$$PF.IPF\ (P_{id}) = \frac{f(Pid)}{\max(f(pid))} * log(\frac{Q}{q(Pid)}) \tag{22}$$

PF is the normalized frequency $f(P_{id})$ of a page $P_{id}$ in a given query session. IPF is the ratio of the total number of query sessions Q in the whole log file to the number of query sessions $q(P_{id})$ in which given page $P_{id}$ is clicked.

Time ($P_{id}$) is the ratio of time spent on the page $P_{id}$ in a given query session to the total duration of the query session.

## 4.1 Architecture of Proposed Search Engine

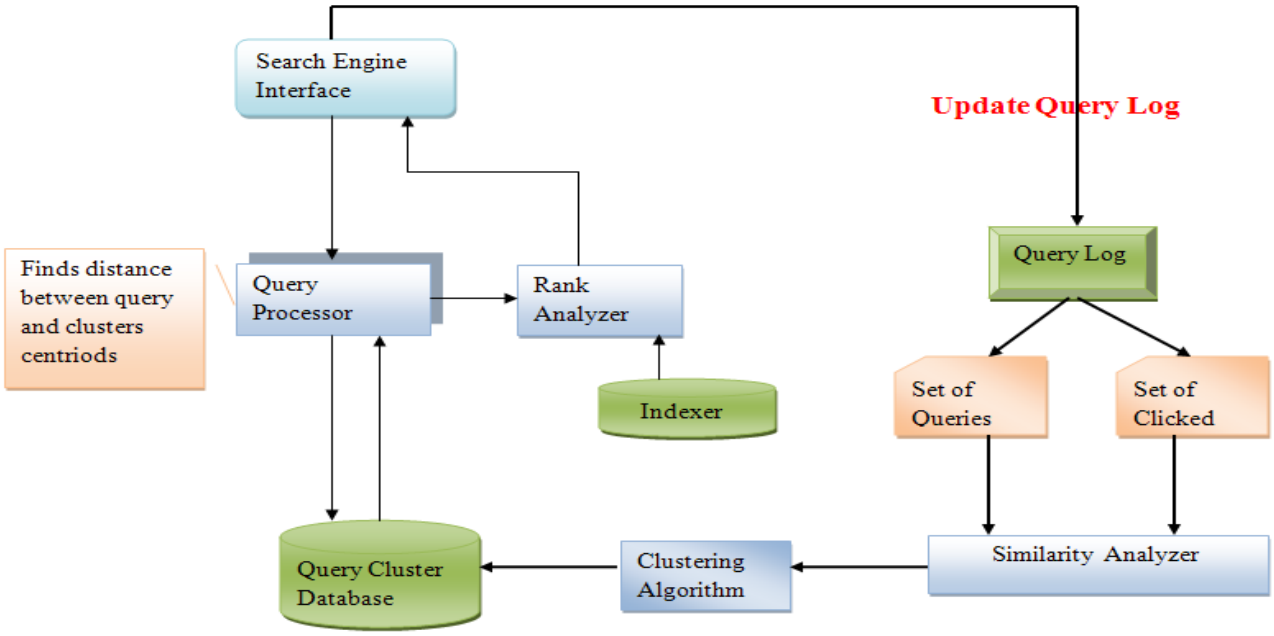


**Figure 4.1:** Architecture of Strong Information Scent based Search Result

## 4.2 Details of the Proposed Search Engine

The architecture is comprises of following components:

- **Search Engine Interface:** User fires the queries at the search engine interface.
- **Query Processor:** It processes the request of the query. It finds the Euclidean distance (D) between the query and the centroid of each cluster and fetches the urls based on the Euclidean distance (D) ≥ Threshold. The fetched urls are returned according to their ranks set by rank analyzer.

- **Query Log:** Query log is the log file which comprises of the database of all the queries fired by all users, user ID, time of query, clicked urls. The query log provides necessary details about the users which is used for mining and helps discovering their surfing pattern.
- **Similarity Analyzer:** In my proposed methodology, similarity analyzer will find the similarities between the queries of different users and similarities between the clicked urls after query is fired. In similarity analyzer, time spent on the clicked urls is also considered. It will be explained in later section in detail.
- **Clustering Algorithm:** It will perform the clustering on the basis of found similarity. It will form the cluster of similar queries with their clicked urls. It will be explained in later section in detail.
- **Query Cluster Database:** It will make the database of the clusters of the queries with their urls and which will return the strong information scent web pages on the fire of user's query.
- **Rank Analyzer:** Rank Analyzer will rank the urls fetched by the query processor from the query cluster database. It will work on the Page Rank algorithm.

## 4.3 Similarity Analyzer

Similarity analyzer works on three similarity criteria:

- Similarity between queries.
- Similarity between clicked urls.
- Time spent on a clicked web page.

**Similarity between users queries**

Similarity between the users queries in the log file is calculated by finding the similarities between the queries content words. This similarity is assisted by using the cosine similarity as discussed earlier in section 2.4.5. It is given by

$$Sim_{query}(q_1, q_2) = \frac{q1.\ q2}{\|q1\| \, |q2|}} \quad (23)$$

Where $q_1$ and $q_2$ are the two queries. $\|q_1\|$ is the Euclidean norm of vector $q_1=(q_{1a},q_{1b},q_{1c},\ldots\ldots,q_{1z})$ defined as $\sqrt{q1a^2 + q1b^2 + \cdots q1z^2}$ as discussed earlier in section 2.4.5.Similarly, $\|q_2\|$ is the Euclidean norm of vector q.

**Similarity between Clicked Urls**

Two queries are similar, if it lead to the selection of the same or similar web pages. It is given by:

$$Sim_{clicked}(q_1, q_2) = \frac{\text{commom urls clicked in two queries}(q1, q2)}{\text{maximum}(\text{clicked urls in queries}(q1, q2))} \quad (24)$$

**Total Time spent on Common Clicked Web page between Two Queries**

In my proposed methodology, I am also considering the time spent on a web page to give more weight to the similarity measure. Here, I am assuming that if a user is spending time which is greater than some threshold time, then, that page reflects strong information scent i.e. the quality of a web page is good where user's information need is satisfied.

**Proposed Similarity Metric**

Similarity metric, by combining all the above similarities is given by:

$$S_m = Sim_{query}(q_1, q_2) + Sim_{clicked}(q_1, q_2) + 1(\textit{for each common url if } T > \textit{threshold}) \quad (25)$$

Here,

$Sim_{query}(q_1,q_2)$ is Similarity between users queries

$Sim_{clicked}(q_1,q_2)$ is Similarity between Clicked Urls.

+1 is added to the metric if Time spent on common clicked urls is greater than some threshold.

## 4.4 Clustering Algorithm

Clustering algorithm is used to cluster the query along with their corresponding clicked urls and time spent on these urls on the basis of the proposed similarity metric. Following is the proposed clustering algorithm used.

**Given:** set of n queries and their corresponding clicked urls and time spent on the clicked urls.

Minimum similarity threshold (£)

**Output:** set of clusters C= {$C_1$, C2.....$C_p$}, each cluster is stored in separate array.

**Clustering_Query (n, £)**{

For (each query $q_1$ € n queries){

Set cluster_Id ($q_1$)= $C_p$

$C_p$= {$q_1$}

For(each query $q_2$ €{n-$q_1$} queries){

Find $S_m$(proposed similarity metric given in section 4.3)

If($S_m \geq$ €) then{

$C_p$= $C_p$ U {$q_2$}

}

Else

{ Continue }

}

p = p+1

}

Return $C_p$ }

Now, the mean value of each cluster is calculated which act as the mean centroid of each cluster

The algorithm for finding mean centroid is as follows :-

**Input:** set of clusters, C = {$C_1$,$C_2$.....$C_P$},where each cluster comprises of query keywords, their common clicked urls and the time spent on each clicked url.

**Output:** Mean centroid of each cluster i.e. mean $S_m$, distinct keywords

Mean_Centroid_Calculate(C){

For (each cluster $C_p$ € C){

Find mean centroid i.e. mean $S_m$ value of each cluster and distinct keywords in each cluster

Return mean centroid

}

}

Mean centroid of each cluster is calculated because when a user fire the query in the search box, the similarity($S_{query}$) between the fired query and the distinct keywords in each cluster is calculated and then, Euclidean distance(D) between the mean $S_m$ and Squery is calculated. If D ≥ threshold, the urls from the corresponding cluster (clicked urls in each cluster are strong information scent based urls) are fetched and shown to the users according to their ranks. This whole clustering process discussed here, is highly iterative process.

# Chapter 5: Experimental Results

The proposed methodology is performed over the following table which comprises of following attributes:

- **Rowid:** Which defines the unique row identity.
- **Sno:** Which define the unique identity of the query.
- **Userid:** Which defines the unique user id.
- **Clicked url:** Which defines the hyperlinks clicked after gaining the result by firing query in the search engine.
- **Time Spent:** Which defines the time spent on the clicked pages.

| Rowid | Sno | Query | Userid | Clicked url | Time Spent(minutes) |
|---|---|---|---|---|---|
| 1 | 1 | web mining | 123 | http://www.xyz.com | 1 |
| 2 | 1 | web mining | 456 | http://www.abc.com | 5 |
| 3 | 2 | data mining over web | 789 | http://www.abc.com | 10 |
| 4 | 2 | data mining over web | 189 | http://www.xyz.com | 5 |
| 5 | 3 | web data mining | 198 | http://www.pqr.com | 3 |
| 6 | 3 | web data mining | 123 | http://www.rst.com | 8 |
| 7 | 3 | web data mining | 201 | http://www.lmn.com | 10 |
| 8 | 4 | mining example | 198 | http://www.xyz.com | 5 |
| 9 | 4 | mining example | 111 | http://www.rst.com | 6 |

**Table 5.1:** Search engine log file example

## 5.1 Proposed Similarity Metric

The proposed similarity metric given by equation 25, will be calculated for each query with all other queries, which is as follows:

**Consider Sno 1 query and Sno 2 query, say:**

**q1:** web mining

**q2:** data mining over web

**Step 1:** finding $Sim_{query}(q_1,q_2)$

$$Sim_{query}(q_1,q_2) = \frac{q1 \cdot q2}{||q1||\,||q2||} = \frac{2}{\sqrt{2}\,x\,2} = 0.707$$

**Step 2:** finding $Sim_{clicked}(q_1,q_2)$

$$Sim_{clicked}(q_1,q_2) = \frac{\text{commom urls clicked in two queries(q1,q2)}}{\text{maximum(clicked urls in queries(q1,q2))}} = \frac{2}{2} = 1$$

**Step 3:** Finding total time spent on the clicked urls common to q1 and q2.

T= 1+5 (for http://www.xyz.com) , 5+10 (for http://www.abc.com) = 6,15(consider time threshold to be 5minutes) = +1,+1

**Step 4:** Summing the result of step1, step 2 and step 3 to obtain proposed similarity metric

$S_m = Sim_{query}(q_1,q_2) + Sim_{clicked}(q_1,q_2) + 1$(for each common url if T > threshold)

=0.707 + 1 + 1 + 1 = 3.707

**Consider Sno 1 query and Sno 3 query,say:**

**q1:** web mining

**q2:** web data mining

**Step 1:** finding $Sim_{query}(q_1,q_2)$

$$Sim_{query}(q_1,q_2) = \frac{q1 \cdot q2}{||q1||\,||q2||} = \frac{2}{\sqrt{2}\,x\,\sqrt{3}} = 0.816$$

**Step 2:** finding $Sim_{clicked}(q_1,q_2)$

$$Sim_{clicked}(q_1,q_2) = \frac{\text{commom urls clicked in two queries(q1,q2)}}{\text{maximum(clicked urls in queries(q1,q2))}} = 0$$

**Step 3:** Finding total time spent on the clicked urls common to q1 and q2.

T= 0

**Step 4:** Summing the result of step1, step 2 and step 3 to obtain proposed similarity metric

$S_m = Sim_{query}(q_1,q_2) + Sim_{clicked}(q_1,q_2) + 1$(for each common url if T > threshold)

=0.816 + 0 + 0 = 0.816

**Consider Sno 1 query and Sno 4 query,say:**

**q1:** web mining

**q2:** mining example

**Step 1:** finding $Sim_{query}(q_1,q_2)$

$$Sim_{query}(q_1,q_2) = \frac{q1 \cdot q2}{||q1||\,||q2||} = \frac{1}{\sqrt{2}\,x\,\sqrt{2}} = 0.5$$

**Step 2:** finding $Sim_{clicked}(q_1,q_2)$

$$Sim_{clicked}(q_1,q_2) = \frac{\text{commom urls clicked in two queries(q1,q2)}}{\text{maximum(clicked urls in queries(q1,q2))}} = \frac{1}{2} = 0.5$$

**Step 3:** Finding total time spent on the clicked urls common to q1 and q2.

T= 1+5 (for http://www.xyz.com) =6 (consider time threshold to be 5 minutes) = +1

**Step 4:** Summing the result of step1, step 2 and step 3 to obtain proposed similarity metric

$S_m = Sim_{query}(q_1,q_2) + Sim_{clicked}(q_1,q_2) + 1$(for each common url if T > threshold)

=0.5 + 0.5 + 1 = 2

**Consider Sno 2 query and Sno 3 query,say:**

**q1:** data mining over web

**q2:** web data mining

**Step 1:** finding $Sim_{query}(q_1,q_2)$

$$Sim_{query}(q_1,q_2) = \frac{q1 \cdot q2}{||q1||\,||q2||} = \frac{3}{\sqrt{4}\,x\,\sqrt{3}} = 0.866$$

**Step 2:** finding $Sim_{clicked}(q_1,q_2)$

$$Sim_{clicked}(q_1,q_2) = \frac{\text{commom urls clicked in two queries(q1,q2)}}{\text{maximum(clicked urls in queries(q1,q2))}} = 0$$

**Step 3:** Finding total time spent on the clicked urls common to q1 and q2.

T= 0

**Step 4:** Summing the result of step1, step 2 and step 3 to obtain proposed similarity metric

$S_m = Sim_{query}(q_1,q_2) + Sim_{clicked}(q_1,q_2) + 1$(for each common url if T > threshold)

=0.866 + 0 +0=0.866

**Consider Sno 2 query and Sno 4 query,say:**

**q1:** data mining over web

**q2:** mining example

**Step 1:** finding $Sim_{query}(q_1,q_2)$

$$Sim_{query}(q_1,q_2) = \frac{q1 \cdot q2}{||q1||\,||q2||} = \frac{1}{\sqrt{2}\,x\,2} = 0.353$$

**Step 2:** finding $Sim_{clicked}(q_1,q_2)$

$$Sim_{clicked}(q_1,q_2) = \frac{\text{commom urls clicked in two queries(q1,q2)}}{\text{maximum(clicked urls in queries(q1,q2))}} = \frac{1}{2} = 0.5$$

**Step 3:** Finding total time spent on the clicked urls common to q1 and q2.

T = 5+5 (for http://www.xyz.com) =10(consider time threshold to be 5 minutes) = +1

**Step 4:** Summing the result of step1, step 2 and step 3 to obtain proposed similarity metric

$S_m = Sim_{query}(q_1,q_2) + Sim_{clicked}(q_1,q_2) + 1$(for each common url if T > threshold)

=0.353 + 0.5 + 1 = 1.853

**Consider Sno 3 query and Sno 4 query,say:**

**q1:** web data mining

**q2:** mining example

**Step 1:** finding $Sim_{query}(q_1,q_2)$

$$Sim_{query}(q_1,q_2) = \frac{q1 \cdot q2}{||q1||\,||q2||} = \frac{1}{\sqrt{2} \times \sqrt{3}} = 0.408$$

**Step 2:** finding $Sim_{clicked}(q_1,q_2)$

$$Sim_{clicked}(q_1,q_2) = \frac{\text{commom urls clicked in two queries(q1,q2)}}{\text{maximum(clicked urls in queries(q1,q2))}} = \frac{1}{3} = 0.333$$

**Step 3:** Finding total time spent on the clicked urls common to q1 and q2.

T= 8+6 (for http://www.rst.com) =14(consider time threshold to be 5 minutes) = +1

**Step 4:** Summing the result of step1, step 2 and step 3 to obtain proposed similarity metric

$S_m = Sim_{query}(q_1,q_2) + Sim_{clicked}(q_1,q_2) + 1$(for each common url if T > threshold)

=0.408 + 0.333 + 1 = 1.741

**Tabularizing the results**

| q1 | q2 | $S_w$ (Proposed Similarity Metric) |
|---|---|---|
| (1) Web mining | (2) Data mining over web<br>(3) Web data mining<br>(4) Mining example | 3.707<br>0.816<br>2 |
| (2) data mining over web | (3) web data mining<br>(4) mining example | 0.866<br>1.853 |
| (3) web data mining | (4) mining example | 0.408 |

**Table 5.2:** Proposed Similarity Metric Result Table

## 5.2 Clustering

The clustering algorithm discussed in section 4.4 is applied to the table 5.2, we will get following clusters, considering $S_m$ threshold is 1:

**$C_1$:** {(1),(2),(4)}

**$C_2$:** {(2),(4)}

**$C_3$:** {(3)}

Mean centroid of each cluster is:

**Mean Centroid of C1:**

mean $S_m = S_m(1,2) + S_m(1,4)/2 = 2.8535$

Distinct keywords = {web, mining, data, over, example}

**Mean Centroid of C2:**

mean $S_m = S_m(2,4) = 1.853$

Distinct keywords = {data, mining, over, web, example}

**Mean Centroid of C3:**

Since there is only one element in cluster 3. Hence, mean $S_m = 0$

Distinct keywords = {web, data, mining}

## 5.2.1 Comparing the Clustering with Varying Thresholds

The result in the table 5.2 is evaluated for clustering with different threshold values.

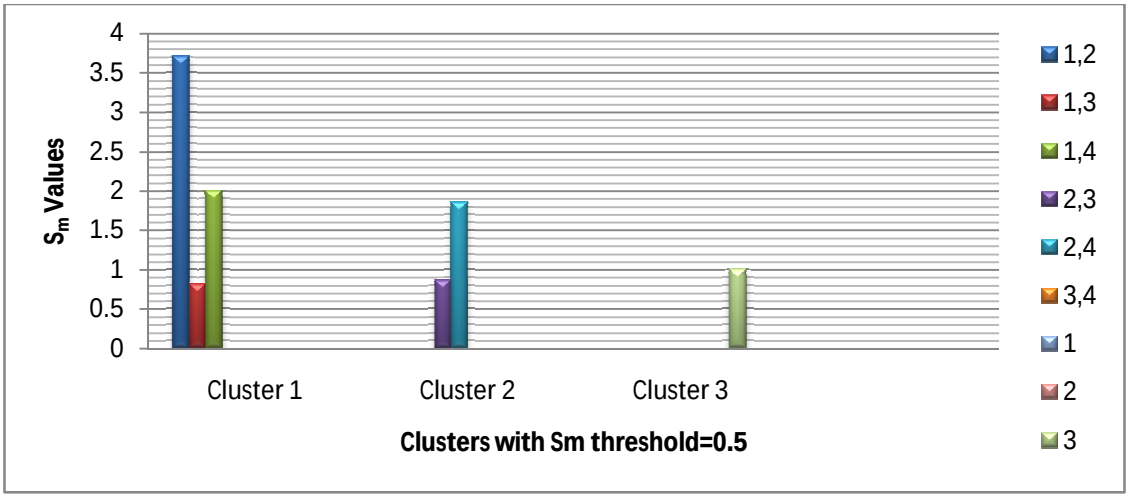


**Figure 5.1:** Clusters formation with $S_m$ Threshold = 0.5

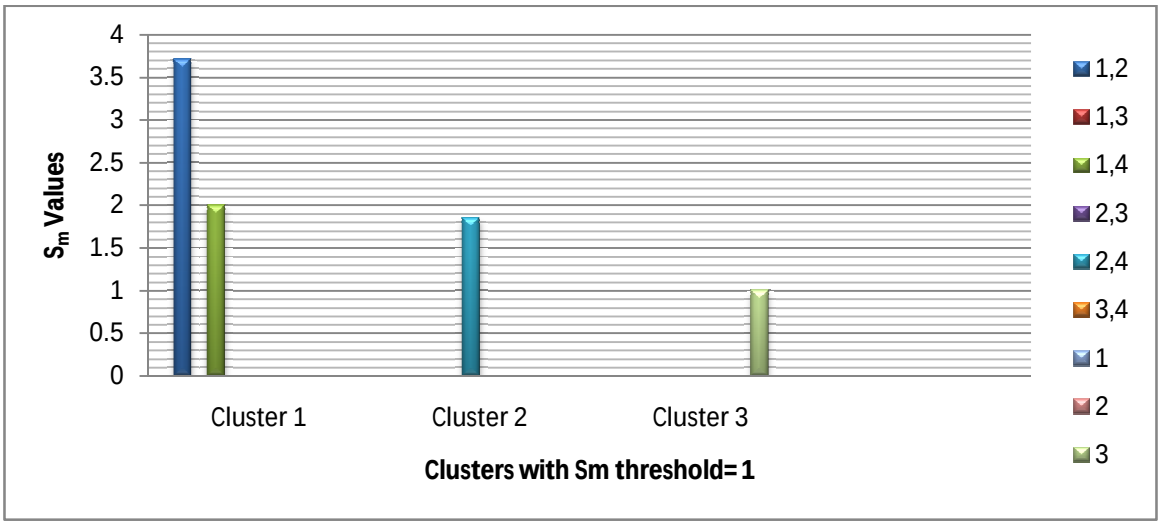


**Figure 5.2:** Clusters formation with $S_m$ Threshold = 1

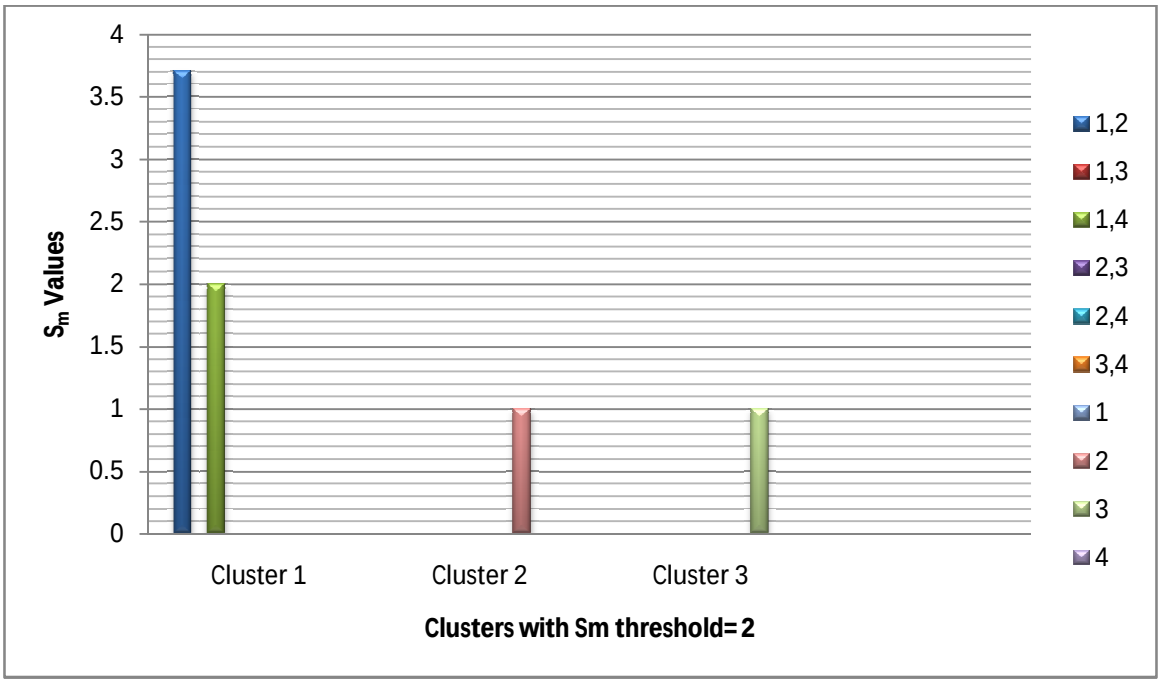


**Figure 5.3:** Clusters formation with $S_m$ Threshold = 2

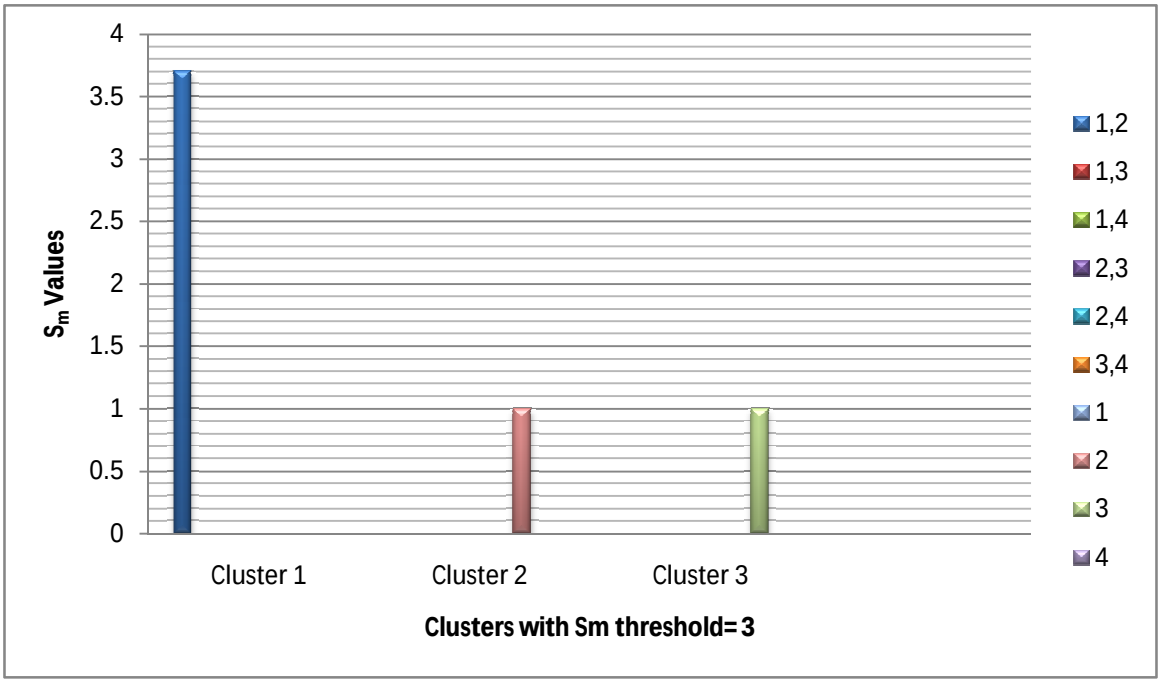


**Figure 5.4:** Clusters formation with $S_m$ Threshold = 3

# Chapter 6: Conclusion and Future Work

Search engines, indeed, the best application of web mining which returns the links in return of the query. Previous research works on optimizing the result of search engine are based on the page rank algorithms, weighted page rank algorithm, Hypertext Induced Topic Search (HITS) which has improved the result of search engine to a large extent. But still, users gets huge list of links in response to their queries which may confuses the user which one to navigate and increases the information seeking time of the surfer. The current work embeds the web mining techniques with the theory of information foraging and focuses on how to optimize the web search result in order to reduce the surfer's navigation time and returning the user with the web pages which satisfies his/her information need. The proposed work makes use of web log mining, clustering, cosine similarity and concept of information scent to accomplish the task of providing the user with the strong information scent based web pages to the surfer. The proposed work is assisted by the experimental results which concludes the result with the formation of query clusters along with their common clicked urls and time spent on these urls.

## 6.1 Future Work

There is a wide scope for the future exploration of the work. These scopes are summarized below:

- The proposed approach can be applied to the large dataset with multidimensional attributes and then, analyzing the result.
- The page rank of the web pages can be updated based on the clusters formed in this approach.
- Frequent pattern mining algorithm can be used to evaluate the frequent patterns of the urls and can be provided weight which can be used to enhance the page rank of a web page.

# Bibliography


[1] Jaideep Srivastava, Prasanna Desikan and Vipin Kumar,"Web mining- Accomplishments and Future Directions"

[2] Renata Ivancsy and Istvan Vjak, "Frequent Pattern Mining in Web Log Data", Acta Polytechnica Hungaria, Vol. 3, No. 1, 2006

[3] Jaiwei, H., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann, San Francisco (2006)

[4] Joni Pajarinen, "Page Rank/HITS algorithm ", March 19, 2008

[5] Paul Raj Punia,"Data Warehousing and Fundamentals",Wiley, 2007

[6] Raymond Kosala and Hendrik Blockeel,"Web Mining Research: A Survey ", ACM SIGKDD, Volume 2 Issue 1, july 2000.

[7] Jefreey Heer, Ed. H. chi,"Identification of Web User Traffic Composition using Multi-Modal Clustering and Information Scent", Xerox Pero Alto Research Center, CA 94304, 2000

[8] Peter Perolli and Stuard K. Card,"Information Foraging ", Psychological Review, UIR Technical Report, January 1999.

[9] Peter Perolli, "Information Foraging and Information Scent: Theory, Models and Applications", Xerox Pero Alto Research Center

[10] Ed. H Chi, Peter Perolli, Kim Chen and James Pitkow,"Using Information Scent to Model User Information Needs and Actions on the Web", Xerox Pero Alto Research Center, CA 94304, ACM, 2001

[11] Suruchi Chawala and Dr. Punam Bedi,"Improving Information Retrieval Precision by Finding Related Queries with Similar Information Need using information Scent", IEEE,2008

[12] Sue Warcup and Prof. Don Zimmerman,"The Relevance of Information Scent to Information Seeking on the Web ", IEEE, 2009

[13] Yukio Horiguchi, Shinsu An, Tetsuo Sawaragi and Hiroaki Nakanishi"Analysis of Menu Selection Behavior using Information Scent Model", IEEE, 2012

[14] Wikipedia.org/information Foraging

[15] A.K Sharma, Neha Aggarwal, Neelam Duhan and Ranjan Gupta,"Web Search Result Optimization by Mining the Search Engine Query Logs", IEEE,2010

[16] R. Umagandhi, Dr. A.V.Senthil Kumar,"Time Independent Query Recommendations from Search Engine Query Logs",IEEE

[17] Zhou Hui, Qin Shigang, Liu Jinhua, Chin Jianli,"Study on Website Search Engine Optimization", International Conference on Computer Science and Service System, IEEE, 2012

[18] Wing Shun Chan, Wai Ting Leung and Dik Lun lee, "Clustering Search Engine Query Log containing Noisy Clickthroughs", IEEE, 2004

[19] Mehdi Hosseini and Hassan Albolhassani,"Mining Search Engine Query Log for Evaluating Content and Structure of a Website", IEEE, 2007

[20] http://www.tutorialspoint.com/searchengineoptimization

[21] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery: An overview," in *Advances in Knowledge Discovery& Data Mining*, U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds. Cambridge, MA: MIT Press, 1996, pp. 1–34.

[22] Agarwal C.C. and P.S. Yu., "Data Mining Techniques for Associations, Clustering and Classification", Proceedings of the Third Pacific-Asia Conference, PAKDD-99, Beijing, China, April, 1999.

[23] D. J. Hand, "Pattern detection and discovery," in Pattern Detection and Discovery, ser. Lecture Notes in Computer Science, D. J. Hand, N. Adams, and R. Bolton, Eds. Berlin, Germany: Springer-Verslag, 2002, vol. 2447, pp. 1–12.

[24] P.Ravi Kumar and Ashutosh Kumar Singh,"Web Structure Mining: Exploring hyperlinks and algorithms for information retrieval ", American Journal of Applied Sciences, pp. 840-845, ISSN 1546-9239, 2010

[25] Pooja Sharma and Asst. Prof. Rupali Bhartiya,"An Efficient Algorithm for Improved Web usage Mining ", International Journal of Computer Technology and Application, Vol. 3 (2), pp. 766-769, ISSN: 2229-6093

[26] Usama Fayyad, Padhraic Smyth, Gregory,"The KDD process for extracting useful knowledge from volumes of data", ACM,Volume 39 Issue 11, Nov. 1996 Pages 27-34

[27] Tamanna Bhatia,"Link Analysis Algorithm for Web Mining", IJCST, VOL. 2, Issue 2, ISSN:2229-4333, June 2011

[28] Viswanath, P.; Sarma, T.H., "An improvement to k-nearest neighbor classifier," *Recent Advances in Intelligent Computational Systems (RAICS), 2011 IEEE* , vol., no., pp.227,231, 22-24 Sept. 2011

[29] J. R. Quinlan, "Generating production rules from decision trees," in Proc. Int. Joint Conf. Artificial Intelligence, San Francisco, CA, 1987, pp. 304–307.

[30] Shesh Narayan Mishra, Alka Jaiswal and Asha Ambhaikar, "An Efficient Algorithm for Web Mining based on topic sensitive link analysis", IJARSCSSE, Volume 2, Issue 4, ISSN: 2277 128X, April 2012