# Building Dynamic E-Business Systems Architecture

**P. S. Lokhande[1] B. B. Meshram[2]**
[1]Assistant Professor [2]Professor
[1,2]Department of Computer Engineering
[1]AIKTC, Navimumbai, India [2]VJTI, Matunga, Mumbai, India

*Abstract*— Web personalization is the process of customizing a web site as per user's preferences by using the data mining, taking advantage of the knowledge acquired from the analysis of the user's navigational behavior. With personalization, when a particular person visits a Website, the recommendation can be designed specifically for that person. The Dynamic E-Business website can be designed using web personalization. The proposed E-Business Web personalization model consists of preprocessing, Knowledge Discovery using data mining and recommendation engine. The proposed preprocessing algorithm consists of three subroutines, namely Cleaning non-required server log, Identification of User from IP address and Log, Creating session for every user and preparing Transaction file. The knowledge discover module consists of Aprori Algorithm for TRP association rule, Sequence Pattern Algorithm for Mining log and Access Log and ID3 Algorithm for Classification of the registered Users of the website. Depending on the Sequence pattern, association rule and user class the recommendation module recommends the desired web page link to the user.

***Key words:*** E-business architecture, Dynamic web e-business system, Web mining, Data mining, Web Personalization, Data mining algorithms

## I. INTRODUCTION

Today E-commerce evolved as a biggest online market for selling goods online. Reach and rise of internet supported the online shopping to grow at faster pace. The ability to track user's browsing behavior down to individual mouse clicks has brought the vendor and end customer closer than ever before. It is now possible for vendor to personalize the product message for individual customers at a massive scale which is referred as Web Page Personalization.

Various techniques are used for personalization task[2]. Personalized pages engage visitors to a web site at a very early stage (i.e. before registration or authentication) personalization tools must rely primarily on click stream data captured in web server logs. The problem with the www is the user's behavior is dynamic in nature, which changes depending on the user's interest and many constraints. The size of the data to be handled is in terms of terabytes. To deal with such problem the data mining techniques is one solution, which is applied on the WWW database and called as web mining.

### *A. Need for Data Mining:*

The web contains the rich and dynamic collection of hyperlink information and web page access and usage information, providing the rich sources of data mining .So one of the popular applications of data mining is World Wide Web Mining.

With web page personalization, advertisements to be displayed to a potential customer are chosen based on specific knowledge concerning that customer. The goal here is to entice a current customer to purchase something he or she may not have thought about purchasing. Personalization is almost the opposite of targeting. Personalization include such techniques as use of cookies, use of databases and more complex data mining and machine learning strategies. Web usage mining focuses on techniques that could predict user behavior while the user interacts with the web. One of the most successful and widely used technologies for building the personalization system is record collaborative filtering (CF) [1]. Given a target user's record of activity, CF based technique, such as k-nearest neighbor (KNN) approach compare the record with the historical record of the other users in order to find top k neighbor who have similar test or interest. But these techniques suffer through some of its basic limitation, which reduces the efficiency, scalability etc. Using data mining technique is one of the solutions to it.

The goal of the web personalization system is to provide the user most likely links of the web page depending upon user's current navigated links [2]. The system will use the data mining techniques like association rule and sequence pattern matching on the web data and transaction data to find out the likely accessed pattern. Until now whatever is the model given for the Web Personalization System is based only on the server log data and the current user-browsing pattern. The system can be designed, which have combination of market basket analysis and server log rule extraction.

Principal elements of Web Page Personalization(WPS) includes modeling of web objects like pages etc. and subjects (users), categorization of objects, matching between and across objects and/or subjects and determination of the set actions to be recommended for personalization .

Existing systems used by many companies, as well as approaches based on collaborative filtering rely heavily on getting human input e.g. user profile for determining the personalization action.

## II. PROPOSED SYSTEM

### *A. Statement of the Problem:*

The problem with the www is the user's dynamic behavior, which changes depending on the user's interest and many constraints. The size of the data to be handled is huge. To deal with such problem the data mining techniques is one solution, which is applied on the WWW database[3].

Use of data mining technique on the web data to serve the personalization task is one of the best choices to solve the problem. The main task in Web Personalization System is to provide the most likely visiting link in the future for particular user depending on various users current navigated browsing pattern. The job is to design the Web Personalization System for an e-commerce site.

In our project the site is of the superstore shop which sells everything from food to electronic goods. The sellers want the site to be modified:

- To increase the cross sells.
- To provide the different advertising policy to different user.
- To provide the ease of surfing the site to the user.
- To get the group of user of particular interest.
- To suggest customer an alternative product similar to his interest.
- To show customer products based on his age and income group.

This will help the seller to get potential customer and new market policy in future. Here the shop is maintaining the data for the purchased items and the customers of the shop. It is also maintaining server log data. The task is to mine both these data and get the association rule from these and provide the recommendation to most likely visiting link on the site.

### *B. Proposed Software Architecture*

*1) System Functional Diagram:*

The figure 2.1 shows detailed architecture of the web personalization system, it consists of Preprocessing module, Data Mining Module and Recommendation Module.

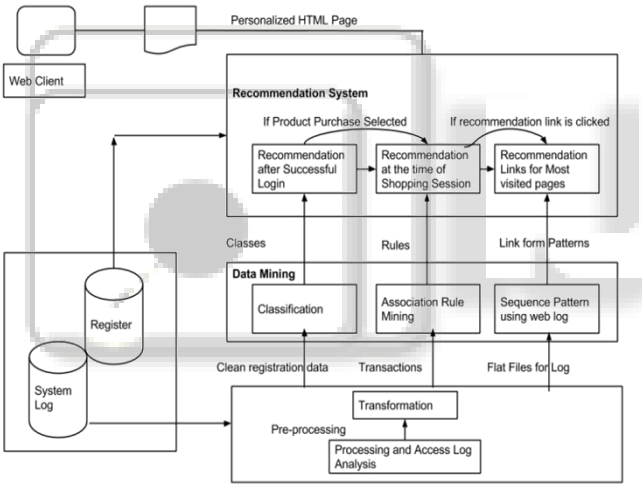



Fig. 2.1: System Functional Diagram

*2) Preprocessing Module:*

System Log is processed at preprocessing module and transformed in to the meaningful information such as registration data, transaction details and user activity log.

*3) Data Mining Module:*

In this module following data mining algorithms are implemented on the received

It shows that classes created by classification algorithm will be used by the site to create personalized homepage of each user. Association rules will be used when user starts a session for shopping. This time site will suggest links to the user for new purchase along with his choice. If the user shows interest in the suggested links and clicks on the link, the new web page will have recommended links from sequence pattern algorithm.

### *C. Proposed Software Architecture:*

A component diagram models the pieces of software, embedded controllers etc. that will make up a system. A component diagram has a higher level of abstraction than a class diagram-usually a component is implemented by one or more classes. Component can encompass a large portion of a system. The component diagram for the proposed system is as shown below fig. no. 3.15. Homepage.java is the main program which creates the default user interface to the web site. The Data mining module is the separate module which is run by the administrator to update the information for personalization like new association rules, classes of new user's and sequence pattern's. InitApriori.java, ID3mod.java and Updatelog.java are the programs for mining the data from the database Transaction, Register and AccessLog respectively. The output of the mining algorithm will be stored in the TRP(transaction processing output), Classification and Mininglog database tables[APR].

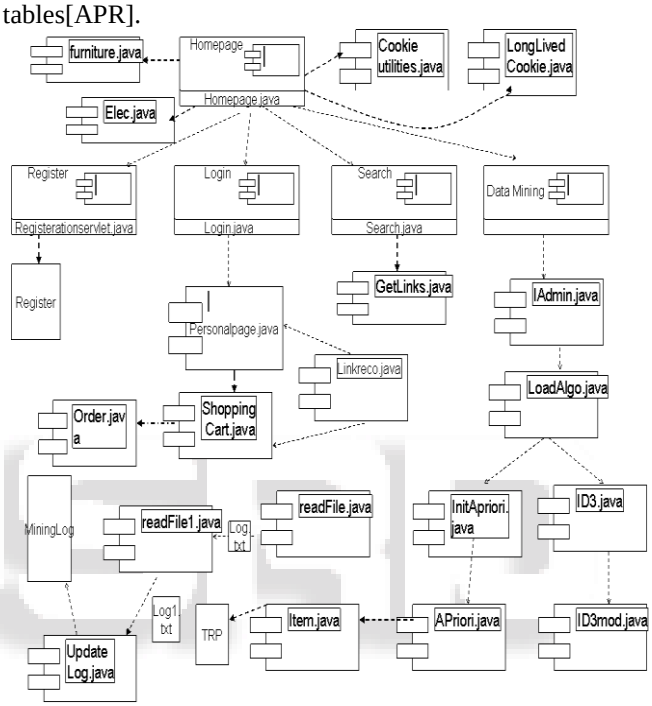



Fig. 2.2: Component Diagram for Web Personalization System

### *D. Data Structure Design*

For the Web personalization system various databases are maintained which are as follows:

*1) Login Table: Here login information about the user is stored*

Login (Name, Emailid, pswd, voice)

The fields of the table are as follows:

Emailid: this field stores the emailed of the user which should be unique per user.

Pswd : this field stores the password of the user.

Voice: this field stores the address of the welcome voice file created for the particular user.

*2) Advertisements:*

Advertisements to be shown to user are stored in this table.

Advertisement(addid,Title,Pagelink,Image,Descriptipon)

where

Addid: is the field for the identification of advertisement number.

Title: This field have values like sport, music, reading etc. to identify their category.

Pagelink: this field stores the address of the html page to which program should go after clicking the link.

Image: this field has address of the image file to be displayed to the user for advertisement.
Description: is the text to be displayed alongwith image for the product advertisements.

*3) Register:*

Here the registration information provided by user is stored. After registration form is completed data will directly go to the REGISER table.
Register (Fname, Lname, Emailid, pswd, Sport, Shopping, Sport, Music, Reading, Shopping, Gender, Age, Profession)

These are all field which stores the information collected from the registration form.
FNAME: first name of the user is stored in this field.
LNAME: Last name of the user is stored in this field.
EMAILID: Email id of the user is stored in this field.
PSWD: password entered by the user is stored in this field.
SPORT: if user selects sport as his interest in the form, 'Sport' value will be entered in the table.
MUSIC: if user selects music as his interest in the form, 'Music' value will be entered in the table.
READING: if user selects Reading as his interest in the form, 'Reading' value will be entered in the table.
SHOPPING: if user selects shopping as his interest in the form, 'shopping' value will be entered in the table.
GENDER: Gender value entered by the user will be stored in the field.
AGE: Age value selected from the drop down box from the registration form will be stored.
PROFESSION: profession value selected from the registration form will be stored.

*4) Product:*

In this table product details are stored
The field description is as below:
Product( Productid, SD, LD, Cost, Productname, Productpage)
PRODUCTID: each product will have unique id & will be stored in this field.
SD: Product's short description or title is stored in this table.
LD: Product's detailed or Long description is stored in this table.
COST: cost of the product is stored in this table.
PRODUCTNAME: name of the product is stored in this field.
PRODUCTPAGE: If any separate html link is there for the particular product, then that address is stored in this field.

*5) Classification:*

In this table output from ID3 algorithm is stored. The registered users will be classified.
Classification (Emailid, Fname, Class, )
The classification table have following fields.
EMAILID: Email id of the user is stored in this field.
FNAME: FNAME: first name of the user is stored in this field.
CLASS: The class of the user found from the classification algorithm i.e. ID3 is stored in this field. The class will help to personalize the web page of user faster.

*6) Access Log:*

Log data mined from the web server is stored in this table.
Log data after preprocessing is stored in this table.
Accesslog (IPaddress, Datetime, GMT, Method, Pagelink, Protocol, Port, ID)
IPADDRESS: Ipaddress of the user who visited the web site is stored in this field.
DATETIME: date and time when user accessed the site are stored here.
GMT: GMT value from the log for particular record is stored in this field.
METHOD: Method of accessing particular page link GET or POST is stored here.
PAGELINK: The accessed page by the user is stored here.
PROTOCOL: protocol used for accessing the web page of the site is stored here
PORT: port number used to access the web page is stored.
ID: Idenetification number of the user session is store.

*7) MiningLog: Here the second iteration output from the access log is stored.*

The fields of the table are as follows
Mininglog ( IPaddress, Date, Hour, MIN, Seconds, Pagelink)
IPADDRESS: Ipaddress of the user who visited the web site is stored in this field.
DATE: date when user accessed the web page is stored here.
HOUR: At what hour user accessed the web page is stored here.
MIN: minute value when user accessed the web page is stored here.
SECONDS : second value when user accessed the web page is stored here.
PAGELINK: The accessed page by the user is stored here.

*8) Transaction:*

Transaction performed on the web site will be stored in the Transaction table.
Transaction (TID, Television, Refridgerator, Computer, Laptop, Livingroomset, Diningtable, Computer Table, Bed)
Transactions on the Web site are stored in this table. The descriptions of the fields are as follows.
TID: transaction identification number is stored here.
TELEVISION: If TV is purchased during shopping this field have value "1", otherwise "0".
REFRIDGERATOR: If fridge is purchased during shopping this field have value "1", otherwise "0".
COMPUTER: If computer is purchased during shopping this field have value "1", otherwise "0".
LAPTOP: If laptop is purchased during shopping this field have value "1", otherwise "0".
LIVINGROOMSET: If living room set is purchased during shopping this field have value "1", otherwise "0".
DININGTABLE: If dining table is purchased during shopping this field have value "1", otherwise "0".
COMPUTER TABLE: If computer table is purchased during shopping this field have value "1", otherwise "0".
BED: If bed is purchased during shopping this field have value "1", otherwise "0".

*9) TRP:*

Output from the apriori algorithm with support value is stored in the table. The field's are storing the association rules and support value will be stored in the support field. The item1 to Item8 fields get filled as we find out association rule's between them. Item1 field is for TV, Item2 for Fridge and so on.

*10) PageLinkID:*

This table is used to store the pagelinks and their path.
Pagelink ( Pagelink, ID)

This table is helpful for giving the identification to the pages for sequence pattern matching.
PAGELINK: This stores the value of the page address which is there in access log.
ID: The ID which we want to allot to the page.

## III. ALGORITHM

### *A. Object Model for Preprocessing System:*

The preprocessing system, collects and cleans the data. The data is again stored back in the various database. The preprocessing system reads the web usage log files which are stored at the web server and uses string tokenizer class to separate the fields of the text file so that it can be stored in the database. The non required field will be removed from the data and only useful information like IPAddress, session no. , Page visited, Date time hour etc. will be separated and stored in the database. This preprocessing is required for the data mining algorithm to which these tables will be given as input.

The class diagram shown in Fig. 3.1 for the preprocessing module of the system. The classes like user identification, sitemapper will collect data for preprocessing. Sitemapper will collect the data for visited links of HTML pages. User identification will use the cookies to identify the new user and previous users which will help for the personalization process.

### *B. Knowledge Discovery:*

The term Data Mining generally refers to a process by which accurate and previously unknown information can be extracted from large volumes of data in a form that can be understood, acted upon, and used for improving decision processes. Data Mining is most often associated with the broader process of Knowledge Discovery in Databases (KDD), "the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data"[KDD].

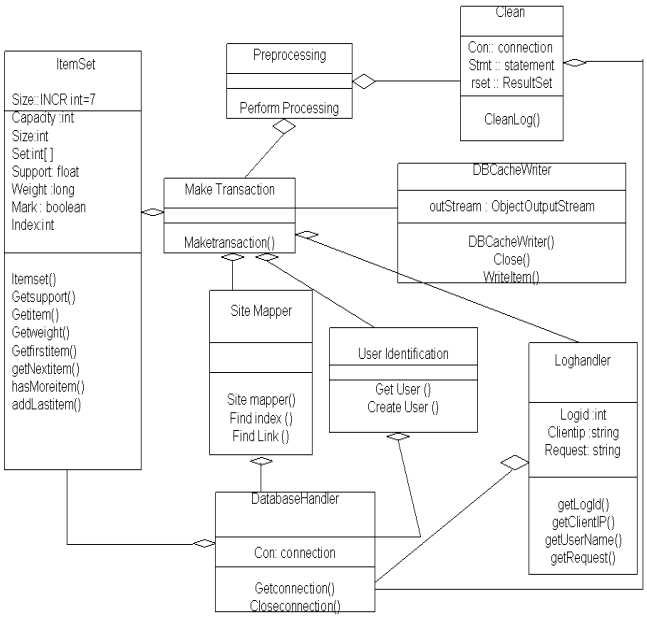



Fig. 3.1: Class Diagram for Preprocessing System

### *1) Knowledge Discovery Definition:*

The entire process of applying a computer-based methodology, including new techniques, for discovering knowledge from data. Many people treat DM as synonym for another popular used term, knowledge discovery in database. In other view DM is just a part of KDD process. KDD process consists of an interactive sequence of the following

1) Data Integration: In this multiple data sources can be combined for the further easier processing.
2) Data selection: In this data relevant to the analysis task are retrieved from the database.
3) Data Transformation: In this the data are transformed or consolidated into forms appropriate for the mining process by performing the summary or aggregation operations for instance.
4) Data mining: In this the intelligent methods are applied in order to extract data patterns.
5) Pattern evolution: In this step the truly interesting patterns representing knowledge based on some interesting measures.
6) Knowledge representation: In this stage the various techniques are used to representing the knowledge to the user

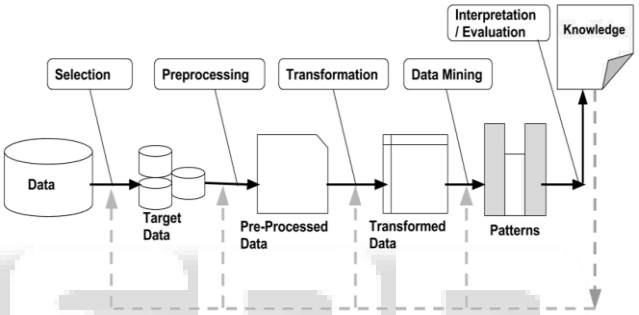



Fig. 3.2: Steps comprising the KDD process.

### *C. Recommendation:*

In the diagram shown for recommendation module, homepage is main class which starts the application. If login process completes successfully, system will create personal page for user depending on the output of classification algorithm. Also Homepage will check the cookies, if already present it will greet the user accordingly. Otherwise new cookie will get created.

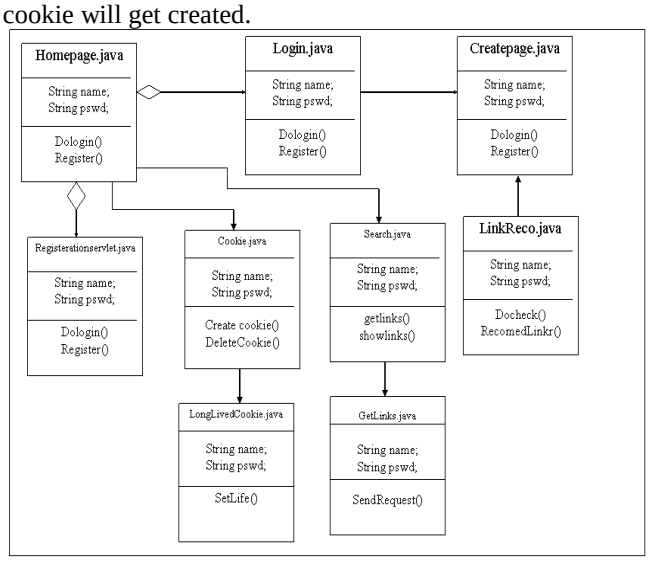



Fig. 3.3: Class Diagram for recommendation module.

The class diagram shows the basic activities and classes involved in the shopping process on the superstore web site. The classes linkreco.java will recommend the links to the user. Login.java will create login session for each user

and personalize web site according to interest of each user and output of classification algorithm output.

*D. Data Mining Algorithms used for Web Personalization*

We begin by describing market basket analysis, the context in which data mining was first proposed. Then, we discuss efficiency considerations, a topic of particular importance given the large size of event histories that must be mined. Last, we show how traditional data mining relates to event mining[3].

*1) Data Mining Task of Classification:*

This classification algorithm is used for classifying the users who have registered on the web site. On the basis of age and gender, the users are classified as old, young, middle, the recommendations for the advertisement is shown accordingly

*E. Most Common Algorithms:*

ID3, C4.5, C5 but we have used the ID3 Algorithms for the implementation of personalization project.

ID3 algorithms introduced by Quinlan for inducing Classification Models, also called Decision Trees. ID3 builds a decision tree from a fixed set of examples. The resulting tree is used to classify future samples.

*1) The ID3 Algorithm*

The ID3 algorithm is used to build a decision tree, given a set of non-categorical attributes C1, C2, .., Cn, the categorical attribute C, and a training set T of records.

function ID3 (R: a set of non-categorical attributes,
C: the categorical attribute,
S: a training set) returns a decision tree;)
begin
If S is empty, return a single node with value Failure;
If S consists of records all with the same value for the categorical attribute,
return a single node with that value;
If R is empty, then
return a single node with as value the most frequent of the values of the categorical attribute that are found in records of S; [note that then there will be errors, that is, records that will be improperly classified];
Let D be the attribute with largest Gain(D,S) among attributes in R;
Let {dj| j=1,2, .., m} be the values of attribute D;
Let {Sj| j=1,2, .., m} be the subsets of S consisting respectively of records with value dj for attribute D;
Return a tree with root labeled D and arcs labeled d1, d2, .., dm going respectively to the trees
ID3(R-{D}, C, S1),
ID3(R-{D}, C, S2), .., ID3(R-{D}, C, Sm);
end ID3;

A statistical property called information gain is used. Gain measures how well a given attribute separates training example into targeted classes. The one with the highest information is selected. In order to define gain, we first borrow an idea from information theory called Entropy. Entropy measures the amount of information in an attribute.

- Entropy: measure of impurity of an arbitrary collection of examples

Given a collection S of c outcomes

$$Entropy(S) = \sum_{i=1}^{c} - p_i \log_2(p_i)$$

Where pi is the proportion of S belonging to class i
Note that if the target attribute can take on c values then the max value of entropy is log(c)

- Information Gain: of an attribute A is the expected reduction in entropy caused by partitioning the examples according to their values for A.

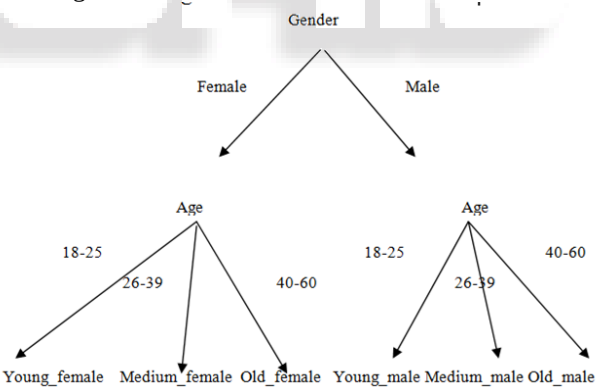
$$Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Where
$\sum$ is each value v of all possible value of attribute A
$S_V$ =Subset of S for which attribute A has value V
$|S_V|$ =number of elements in $S_V$ in
$|S|$ =number of elements in S
Example:
Consider the following set of input for the decision tree algorithm.

Table Editor - "SYSTEM"."REGISTER"

| FNAME | LNAME | EMAILID | PSWD | SPORT | MUSIC | READING | SHOPPING | SEX | AGE | PROFESSION |
|---|---|---|---|---|---|---|---|---|---|---|
| Dipti | Thakur | dthakur@yahoo... | dipti | | Music | | Shopping | Female | 26-38 | Engineer |
| Rajesh | Dushinge | rajesh_d@rediff... | rajesh | Sport | | Reading | | male | 18-25 | Student |
| Kamini | Nalavade | knalavade@ya... | kamini | | Music | Reading | Shopping | Female | 26-38 | Engineer |
| Dipali | Patil | dpatil@yahoo.c... | dipali | | | Reading | | Female | 18 | Student |
| shubhangi | morey | smore@gmail.... | shubhangi | null | null | Reading | Shopping | Female | 18-25 | Student |
| sangita | choudhary | sangita_choud... | sangita | Sport | Music | | | Female | 26-38 | Engineer |
| teena | verma | teenaverma@g... | teena | Sport | Music | | | Female | 18-25 | Engineer |
| swati | patil | spatil@gmail.c... | swati | | Music | Reading | | Female | 26-38 | Engineer |
| Chandrakant | Nalavade | cdnalavade@y... | chandrakant | Sport | | | | male | 51-65 | Lwyer |
| peter | jackson | peterjack@gm... | peter | | | | Shopping | male | 65 | Doctor |
| poonam | nalavade | pnalavade@g... | poonam | Sport | Music | Reading | Shopping | Female | 18-25 | Student |
| Lilavati | Nalavade | lila@yahoo.com | lila | | Music | | Shopping | Female | 65 | Housewife |
| Neelam | Gorhe | naalarn@infy.c... | neelam | | | | Shopping | Female | 51-65 | Housewife |
| Manoj | Sankhe | contactmanoj@... | manoj | Sport | Music | | | male | 39-50 | Engineer |
| Sachin | Tendulkar | sachintendulka... | sachin | Sport | | Reading | | male | 39-50 | other |

The tree generated for the above data is




Gain (S, Gender) = .151
Gain (S, Gender) = .940 – (7/14).985 – (7/14).592
= .151
Gain (S, Age) = .048
Gain (S, Age) = .940 – (8/14).811 – (6/14)1.0
= .048
Looking at the gain values highest gain values will be selected. So Gender will be selected as root. The rules like
If (Gender=Male) and (18<Age<26) then class is young_female.
Will be generated from the tree.
The ID3 algorithm after execution classifies the given input into predefined classes.
As shown in the figure it will generate a decision tree for the given input and place the input into appropriate class.
The entropy value for each classifier is also calculated.

*2) Data Mining Task of Clustering*

Definition – Set of methodologies for automatic classification of samples into a number of groups using a measure of association, so that the samples in one group are similar and samples belonging to different groups are not similar.

For example in Retail shop you will have to identify significant customer segments. Customers in your segment liked this product so you will too view the data as a point in multidimensional space.

Clustering is a form of unsupervised learning because we don't know what defines a class or how many there are.

How is this different from Association rules?: Association rules are more local and clustering is global.Association rules look dependencies between variables over the whole set and classification tries to group different elements based on all their attributes.

Since the customers are to be classified into 3 clusters namely Great shopper, medium shopper and small shopper, the hierarchical clustering algorithm suits the best. Hence we choose hierarchical clustering for this purpose.

Clustering is an example of data mining task that fits in the descriptive model of data mining. The use of clustering enables you to create new groups or classes. Clustering technique is otherwise known as unsupervised learning or segmentation. All those data items that resembles more closely with each other are clubbed together in a single group, also known as clusters. The clustering algorithm chosen for implementation is hierarchical clustering.

*F. Algorithms for Cluster Analysis*

There are two basic types of algorithms

*1) Hierarchical:*

- Don't specify the number of clusters
- Hierarchy of clusters
- Dendrogram
- Specify minimum similarity

*2) Partitional:*

- Do specify the number of clusters

We need a method to compute similarity or dissimilarity

Common Measures:

- Euclidean Distance

$$d(x_i, x_j) = (\sum_{k=1}^{m} (x_{ik} - x_{jk})^2)^{\frac{1}{2}}$$

- City Block Distance

$$d(x_i, x_j) = \sum_{k=1}^{m} |x_{ik} - x_{jk}|$$

- Cosine Correlation

$$S_{\cos}(x_i, x_j) = \frac{\sum_{k=1}^{m} (x_{ik} \bullet x_{jk})}{(\sum_{k=1}^{m} x_{ik}^2 * \sum_{k=1}^{m} x_{jk}^2)^{\frac{1}{2}}}$$

- Binary Similarity

Proportion of bits that are the same

## IV. PARTITIONAL ALGORITHMS

Centroid – The mean vector of a cluster

$$M_k = \frac{1}{n_k} \sum_{i=1}^{n_k} x_{ik}$$

Square Error for a Cluster

$$e_k^2 = \sum_{i=1}^{n_k} (x_{ik} - M_k)^2$$

Square Error for the Entire Clustering Space

$$E_k^2 = \sum_{k=1}^{K} e_k^2$$

The objective for partitional Algorithms is to minimize the Square Error for a given number of clusters k

*A. K-Means Partitional Clustering Algorithm*

- Select initial partition with k clusters containing randomly chosen samples and compute the centroids
- Generate a new partition by assigning each sample to the closest cluster center
- Compute new centroids
- Repeat steps 2 and 3 until a specified criterion is met or cluster membership stabilizes

*1) Pseudocode*

```
Pre
D:= {t1,t2,….,tn}  //Set of Elements.
c:    //Number of  desired clusters.
Post
K: //Set of Clusters.
   // Main procedure  for finding cluster set
Procedure Squared_error(D,c)
Begin
1)  Assign each element to some cluster arbitrarily.
K={k1,k2,….,kc}
SE = 0   // SE : Squared error of cluster set
Pre_SE = 1 // Pre_SE: Squared error of privious cluster set
2)  Calculate initial centroid for each cluster
for i = 1 to c
cti =  å(tm)/N    //  tm is one of the N elements of cluster ki
// cti is centroid of cluster Ki
end for
CT={ct1,ct2,…,ctc}   // CT : Array which store the Centroid of the each cluster
While (diff != 0)   // diff: difference between current and previous cluster set squared error
Begin
Pre_SE = SE    // storing the previous squared error
3)  Assign element to appropriate cluster based on  closest centroid.
for i = 1 to n
index = minimum (ti,CT )
// find out the cluster index  with minimum distance  from data element ti  to it's centroid
kindex  = kiindex  + ti    // assign the ti to cluster Kindex
end for
K = {k1, k2, … ,km}    //new cluster set generated after above  reassignment
4)  Calculate initial centroid for each cluster
for i = 1 to c
ctk =  å(tm)/N    //  tm is one of the N elements of cluster ki
// cti is centroid of cluster Ki
end for
```

CT={ct1,ct2,…,ctc} // CT : Array which store the centroid of the each cluster
5) Calculate squared error of each cluster.
Pre
D:= {t1,t2,….,tn} //Set of Elements.
c: //Number of desired clusters.
Post
K: //Set of Clusters.
// Main procedure for finding cluster set
Procedure Squared_error(D,c)
Begin
1) Assign each element to some cluster arbitrarily.
K={k1,k2,….,kc}
SE = 0 // SE : Squared error of cluster set
Pre_SE = 1 // Pre_SE: Squared error of privious cluster set
2) Calculate initial centroid for each cluster
for i = 1 to c
cti = å(tm)/N // tm is one of the N elements of cluster ki
// cti is centroid of cluster Ki
end for
CT={ct1,ct2,…,ctc} // CT : Array which store the Centroid of the each cluster
While (diff != 0) // diff: difference between current and previous cluster set squared error
Begin
Pre_SE = SE // storing the previous squared error
3) Assign element to appropriate cluster based on closest centroid.
for i = 1 to n
index = minimum (ti,CT )
// find out the cluster index with minimum distance from data element ti to it's centroid
kindex = kiindex + ti // assign the ti to cluster Kindex
end for
K = {k1, k2, … ,km} //new cluster set generated after above reassignment
4) Calculate initial centroid for each cluster
for i = 1 to c
ctk = å(tm)/N // tm is one of the N elements of cluster ki
// cti is centroid of cluster Ki
end for
CT={ct1,ct2,…,ctc} // CT : Array which store the centroid of the each cluster
5) Calculate squared error of each cluster.
for i = 1 to c
for j = 1 to m
sei = å ½tj – cti ½2 // sei Squared error for cluster ki
end for
end for
se={se1,se2,…,sec} // se = Array which store the squared error of the each cluster
6) Calculate squared error of cluster set.
SE = åsei // SE : Squared error of cluster set
7) Calculate the difference between current and previous squared_error of cluster set.
diff=SE – Pre_SE // Pre_SE: Squared error of previous cluster set
// if diff = 0 then terminate main while loop
// else continue the next iteration
End // end of main while loop
End// end of procedure

*B. Example*

In our web personalization project clustering is used .The database of customer, some of the tuples, is shown below

| User ID | Name | Age | Profession | Salary(k) |
|---|---|---|---|---|
| 10001 | Poonam | 17 | Student | 2 |
| 10002 | Shrikant | 20 | Student | 4 |
| 10240 | Chandrakant | 35 | Doctor | 10 |
| 10241 | Indira | 50 | Lawyer | 12 |
| 10242 | Dipali | 19 | Student | 3 |
| 20345 | Andrew | 47 | Doctor | 20 |
| 23456 | Navneet | 60 | Engineer | 30 |
| 25689 | Roger | 49 | Doctor | 11 |
| 23778 | Hari | 40 | Pilot | 25 |

Now superstore wishes to group its customer based on common features, which is not having any predefined labels. Based on the outcome of the grouping , they will target marketing and advertising to the different groups. Suppose company wishes to advertise a costly product and for that it required clustering based on income attribute.
Apply squared error clustering algorithm
Input parameters
– Number of clusters = 3
– Database values for salary={2,4,10,12,3,20,30,11,25}
Initial arbiter Assignment
– Cluster 1:={2,12,30}
– Cluster 2:={4,3,11}
– Cluster 3:={10,20,25}
Calculating centroid and squared for each cluster : centroid for ith can be calculated using equation
Cti = ∑(tm)/N where, N = no of element in cluster I
for e.g. centroid for cluster 1:{2,12,30}
ct1 = (2+12+30)/3 = 14.666667
same way we can calculate centroid for others cluster.
Squared error for cluster i can be calculated using equation
sei = ∑ |tj – cti |2 where, tj is one of the element in ith cluster .
for e.g.
squared error for ith cluster,
sei = ( |2-14.66667|2 + | 12-14.66667|2 + |30-14.66667|2 )
= (12.66667)2 + (2.66667)2 + (15.33333)2
= 402.6667
Same way we can find out the squared error for other cluster
Finally,

| Cluster | Centroid | Squared Error |
|---|---|---|
| Cluster 1:={2,12,30} | 14.666667 | 402.6667 |
| Cluster 2:={4,3,11} | 6.0 | 38.0 |
| Cluster 3:={10,20,25} | 18.333334 | 116.66667 |

Now, Squared error of cluster set,
SE = ∑ sei
= 402.6667 + 38.0 + 116.66667
= 557.3334
Previous squared error is 0 so difference between current and previous squared error is also 557.66667
Now, applying successive iterations and calculating above parameters until termination criteria meet (diff = 0).
*1) First iteration*
Now assigning each element to the clusters whose centroid is closest to element.
Foe e.g.
previously we get, ct1 = 14.66667, ct2 =6.0, ct3 =18.33334
So,

1st element 2 is closest to ct2 =6.0 - > 2 is assign to 2nd cluster
2nd element 4 is closest to ct2=6.0 - >2 is assign to 2nd cluster
3rd element 10 is closest to ct2=6.0-> 10 is assigned to cluster 2
4th element 12 is closest to ct1=14.66-> 12 is assigned to cluster 1
5th element 3 is closest to ct2=6.0 -> 3 is assigned to cluster 2
6th element 20 is closest to ct3=18.33-> 20 is assigned to cluster 3
7th element 30 is closest to ct1=18.33-> 30 is assigned to cluster 3
8th element 11 is closest to ct1=14.66-> 11 is assigned to cluster 1
9th element 25 is closest to ct1=18.33-> 25 is assigned to cluster 3
So after above reassignment cluster set look like,

Cluster 1:={12,11}
Cluster 2:={2,4,10,3}
Cluster 3:={20,30,25}

Now calculating centroid and squared error as done in previous iteration,

| Cluster | Centroid | Squared Error |
|---|---|---|
| Cluster 1:={12,11} | 11.5 | 0.5 |
| Cluster 2:={2,4,10,3} | 4.75 | 38.75 |
| Cluster 3:={20,30,25} | 25.0 | 50.0 |

Now, Squared error of cluster set,
SE = ∑ sei
= 0.5 + 38.75 + 50.0
= 89.25
Previous squared error is 557.3334, so difference between current and previous squared error is
(diff > 0) - > so continue for next iteration
Now assigning each element to the clusters whose centroid is closest to element.
Foe e.g.
previously we get, ct1 = 11.5, ct2 =4.75, ct3 =25.0
So,
1st element 2 is closest to ct2 =4.75 - > 2 is assign to 2nd cluster
2nd element 4 is closest to ct2=4.75 - >2 is assign to 2nd cluster
3rd element 10 is closest to ct1=11.5-> 10 is assigned to cluster 1
4th element 12 is closest to ct1=11.5-> 12 is assigned to cluster 1
5th element 3 is closest to ct2=4.75 -> 3 is assigned to cluster 2
6th element 20 is closest to ct3=25.0-> 20 is assigned to cluster 3
7th element 30 is closest to ct1=25.0-> 30 is assigned to cluster 3
8th element 11 is closest to ct1=11.5-> 11 is assigned to cluster 1
9th element 25 is closest to ct1=25.0-> 25 is assigned to cluster 3
So after above reassignment cluster set look like,

Cluster 1:={10,12,11}
Cluster 2:={2,4,3}
Cluster 3:={20,30,25}

Now calculating centroid and squared error as done in previous iteration,

| Cluster | Centroid | Squared Error |
|---|---|---|
| Cluster 1:={10,12,11} | 11 | 2.0 |
| Cluster 2:={2,4,3} | 3.0 | 2.0 |
| Cluster 3:={20,30,25} | 25.0 | 50.0 |

Now, Squared error of cluster set,
SE = ∑ sei
= 2.0 + 2.0 + 50.0
= 54.0
Previous squared error is 89.25, so difference between current and previous squared error diff = 35.25 (diff > 0) - > so continue for next iteration
Now assigning each element to the clusters whose centroid is closest to element.
Foe e.g.
previously we get, ct1 = 11.0, ct2 =3.0, ct3 =25.0
So,
1st element 2 is closest to ct2 =3.0 - > 2 is assign to 2nd cluster
2nd element 4 is closest to ct2=3.0 - >2 is assign to 2nd cluster
3rd element 10 is closest to ct1=11.0-> 10 is assigned to cluster 1
4th element 12 is closest to ct1=11.0-> 12 is assigned to cluster 1
5th element 3 is closest to ct2=3.0 -> 3 is assigned to cluster 2
6th element 20 is closest to ct3=25.0-> 20 is assigned to cluster 3
7th element 30 is closest to ct1=25.0-> 30 is assigned to cluster 3
8th element 11 is closest to ct1=110-> 11 is assigned to cluster 1
9th element 25 is closest to ct1=25.0-> 25 is assigned to cluster 3
So after above reassignment cluster set look like,

Cluster 1:={10,12,11}
Cluster 2:={2,4,3}
Cluster 3:={20,30,25

Now calculating centroid and squared error as done in previous iteration,

| Cluster | Centroid | Squared Error |
|---|---|---|
| Cluster 1:={10,12,11} | 11.0 | 2.0 |
| Cluster 2:={2,4,3} | 3.0 | 2.0 |
| Cluster 3:={20,30,25} | 25.0 | 50.0 |

Now, Squared error of cluster set,
SE = ∑ sei
= 2.0 + 2.0 + 50.0
= 54.0
Previous squared error is 54.0, so difference between current and previous squared error diff = 0.0
(diff = 0) - > so terminate the loop, as we got the desired cluster set.
cluster 0 : {10,12,11}
cluster 1 : {2,4,3}
cluster 2 : {20,30,25}
Means,
cluster 1 : {Chandrakant,Indira.Roger }
cluster 2 : {Poonam, Shrikant, Dipali}
cluster 3 : {Andrew, Navneet,Hari}

### *C. Market Basket Analysis*

Market basket analysis originated from looking at data from supermarkets. The context is as follows. Each customer has a basket of goods. The question addressed is "Which items, when purchased, indicate that another item will be purchased as well?" This is commonly referred to as an association rule. For example, early studies found that when diapers are purchased beer is frequently purchased as well. Association rules indicate a one-way dependency[MBA]. For example, it turns out that purchasers of beer are, in general, not particularly inclined to buy diapers.

To proceed, some notation is introduced. Let $I_1,...I_M$ be items that can be purchased. Thus, each market basket contains a subset of these items. We use $B_1,...B_N$ to denote the set of market baskets, where there is one basket per transaction. Thus, for $1 \le n \le N$, $B_n \subseteq \{I_1,...I_M\}$.

A key data mining problem is to find sets of items, typically referred to as itemsets, that occur in a large number of market baskets. This is captured in a metric called support. Support is computed by counting the number of baskets in which the itemset occurs and then dividing by $N$, the number of baskets. A second and closely related problem addresses prediction. Here, we are looking for $I_{j_1}...I_{j_K}$ that have a high probability of predicting that $I_{j_{K+1}}$ will be in the same basket. The metric used here is confidence. Confidence is computed by counting the baskets in which $I_{j_1}...I_{j_{K+1}}$ occur (which we denote by $Count(I_{j_1}...I_{j_{K+1}})$) and then dividing by $Count(I_{j_1}...I_{j_K})$.

Typically, mining involves finding all patterns whose support is larger than a minimum value of support, which we denote by MinSupp. A naïve approach is displayed in Algorithm 1. QC = $\phi$ For each possible pattern P.

```
Count = 0
For each market basket B
For each item I in P
If I is not in B, advance to next market basket
End
Count++
End
If Count > MinSupp, add P to QC
End
```

Algorithm 1: Naive Approach to Finding Frequent Patterns

In Algorithm 1, QC is the set of qualified candidates, those patterns that have the minimum support level. The algorithm considers all possible patterns, scans through all market baskets for each pattern, and for each market basket, counts the number tests if each item of the pattern is present in the market basket.

Considerable computation time is required to perform this algorithm, even on modest-sized data sets. In particular, observe that the number of iterations in the outer loop is exponential in the number of patterns since there are $2^M - 1$ possible patterns (where *M* is the number of items).

Clearly, Algorithm 1 scales poorly.

Fortunately, the search for frequent patterns can be made more efficient. Doing so rests on the following observation:

The support for $I_{j_1}...I_{j_{K+1}}$ can be no greater than the support for $I_{j_1}...I_{j_K}$.

This means that if we find a pattern with low support, there is no need to consider any pattern that contains that pattern. This is an example of the downward closure property.

With the downward closure property, we can improve the efficiency of Algorithm 1. This is shown below in Algorithm 2 that considers patterns of increasing length. Such a strategy is referred to as level-wise search.

```
FI = {I such that Count(I)> MinSupp} /* Frequent Items */
QC(1) = FIN = 1 While QC(N)
For
 Count = 0
For each market basket B
      For each item I in P
         If I is not in B, advance to next market basket
 End
Count++
 End
 If Count > MinSupp, add P to QC(N+1)
 End
 N++
End
```

Algorithm 2: Using Downward Closure to Find Frequent Patterns

The algorithm first finds frequent items, since by downward closure infrequent items cannot be in frequent patterns. QC(N) contains the qualified contains with N items. The potential patterns with N+1 items are those that have N items in combination with one of the frequent items not already in the N item pattern. Even though Algorithm 2 has four loops instead of the three in Algorithm 1, Algorithm 2 avoids looking through an exponential number of patterns and so is considerably more efficient. The downward closure property holds for some patterns and not for others. In particular, downward closure does not hold for the confidence of association rules. To see this, recall that confidence is computed as $Count(I_{j_1}...I_{j_{K+1}}) / Count(I_{j_1}...I_{j_K})$. Now consider the confidence with which $I_{j_1}...I_{j_K}, I_*$ predicts $I_{j_{K+1}}$. Observe that by including a new item, we decrease (or at least do not increase) the numerator of $Count(I_{j_1}...I_{j_{K+1}}, I_*) / Count(I_{j_1}...I_{j_K}, I_*)$. However, including $I_*$ in the pattern affects the denominator as well. Thus, it is unclear if the resulting ratio will be smaller or larger than the original ratio. Hence, downward closure does not hold.

Now, we return to the problem of mining event data. Here, the context changes in a couple of ways. First, there is no concept of a market basket. However, events have a timestamp and so looking for patterns of events means looking at events that co-occur within a time range. These ranges may be windows (either of fixed or variable size) or they may be contiguous segments of the data that are designated in some other way. In the data mining literature, this is referred to as temporal mining or temporal association.

A second consideration needed in event mining relates to the attributes used to characterize membership in itemsets. Several attributes are common to event data. Event type describes the nature of the event. Event origin specifies the source of the event, which is a combination of the host from which the event originated and the process and/or application that generated the event. (Due to the limited granularity of the data used in our running examples, we simplify matters in the sequel by just referring to the host

from which the event originated.) In addition to type and origin, there is a plethora of other attributes that depend on these two, such as the port associated with a "port down" event and the threshold value and metric in a "threshold violated" event.

The next three sections address patterns we have discovered in the course of analyzing event data: event bursts, periodicities, and mutually dependent events. Each is illustrated using the corporate intranet data. Then we discuss issues related to the efficient discovery of these patterns.

### *D. Event bursts*

This section describes a commonly occurring pattern in problem situations—event bursts. We begin by motivating this pattern and providing an example. Next, we outline our approach to discovering these patterns.

Event bursts (or event storms) arise under several circumstances. For example, when a critical element fails in a network that lacks sufficient redundancy (e.g., the only name server fails), communications are impaired thereby causing numerous "cannot reach destination" events to be generated in a short time period. Another situation relates to cascading problems, such as those introduced by a virus or, more subtly, by switching loads after a failure, a change that can result in additional failures due to heavier loads.

## V. ASSOCIATION RULE MINING

Association rule mining searches for interesting relationship among the items in a given data set. Market basket analysis is very popular example of association rule mining.

Basic Concept- Let I={$I_1$ , $I_2$ , $I_3$ … $I_m$} be the set of items. Let $D_i$ be the task relevant data, be a set of database transactions where each transaction is associated with an identifier, called TID. Let A be a set of items. A transaction T is said to contain A if and only if A is subset of T. Support of an itemset is percentage of transactions which contain that itemset. Large (Frequent) itemset are itemset whose number of occurrences is above a threshold.

### *A. Apriori Properties*

- Large Itemset Property:Any subset of a large itemset is always large.If an itemset is not large, none of its supersets are large.
- Implication:If an itemset is not large, none of its supersets are large.

#### *1) Apriori Pseudo code:*

Find frequent itemset using an iterative level-wise approach based on candidate generation.
Input: Database D of transactions, Minimum support threshold, min support.
Output: Frequent Itemset in D.
Method:
L1= Find Frequent_1_itemset(D);
for (k=2;Fk-1¹ f;k++) {
Ck=apriori_gen(Lk-1, Min_sup)
for all transactions t Î D {
//Scan D for counts
C1=subset(Ck, t) //get the subset of t that are candidates
For each candidate c Î Ct
c.count++
}
Lk={c Î Ck | c.count ³ min_sup};
}
Return L= Uk Lk;
Procedure apriori_gen(Lk-1, Min_sup,frequent(k-1)itemsets,minimum support threshold)
For each itemset I1 Î Lk-1
For each itemset I2 Î Lk-1
If(I1 [1]= I2 [1] ^( I1 [2]= I2 [2] ) ^…… ^(I1 [k-2] =I2 [k-2] ^( I1 [k-1]< I2 [k-1])
Then
C=I1 ►◄I2 // join step:generate candidates
If has_infrequent_subset(C, Lk-1 ) then
Delete c: //prun step remove unfruitful data
Else add c Ck; }
Return Ck;
Procedure has_frequent_itemset(C:candidate k itemset Lk-1 Frequent(k-1)itemset) //uses prior knowledge
For each(k-1)
if (s notÎ Lk-1)
then return true;
else return false;

### *B. Example: Input to Apriori Algorithm*

#### *1) Result*

| Transaction | Items |
|---|---|
| 1 | 1 2 4 5 |
| 2 | 2 3 5 |
| 3 | 1 2 4 5 |
| 4 | 1 2 3 5 |
| 5 | 1 2 3 4 5 |
| 6 | 2 3 4 |

Where
1→ TV
2→ Fridge
3→ Computer
4→ Laptop
5→ Living Room Set
Pass 1:Finding frequent 1-itemsets {1, 2 ,3, 4, 5}
Pass 2:Finding frequent 2-itemsets
{1 2, 1 4, 1 5, 2 3, 2 4, 2 5, 3 5, 4 5}
Pass 3:Finding frequent 3-itemsets
{1 2 4, 1 2 5, 2 3 5,2 4 5}
Pass 4:Finding frequent 4-itemsets {1 2 4 5 }

#### *2) Frequent Itemsets(min_sup = 50%)*

| Support | Itemset |
|---|---|
| 100%(6) | 2 |
| 43%(5) | 5,(2 5) |
| 67 %(4) | 1,3,4,(1 2), (1 5), (2 3), (2 4), (1 2 5) |
| 50%(3) | (1 4).(3 5),(4 5),(1 2 4),(1 4 5), (2 3 5),(2 4 5), (1 2 4 5) |

## VI. CONCLUSION

Web personalization is the process of customizing and displaying the content to Web site visitor according to individual needs. Showing personalized content can be achieved by taking advantage of the user's navigational behavior, as it can be revealed through the processing of the Web usage logs, as well as the user's characteristics and interests.

Many of the methods used in user profiling rise some privacy issues concerning the disclosure of the user's personal data, therefore they are not recommended. Since

user profiling seems essential in the process of Web personalization, a legal and more accurate way of acquiring such information is needed. P3P (Platform for Privacy Preferences Project) is an emerging standard recommended by W3C ( World Wide Web Consortium) that provides a technical mechanism that enables users to be informed about privacy policies before they release personal information and gives them control over the disclosure of their personal data.

The main component of a Web personalization system is the usage miner. Log analysis and Web usage mining is the procedure where the information stored in the Web server logs is processed by applying statistical and data mining techniques, such as clustering, association rules discovery, classification and sequential pattern discovery, in order to reveal useful patterns that can be further analyzed. There are various Data mining algorithms available but we have used ID 3 Algorithm in this project to get the personalized recommendations.

## REFERENCES

[1] Zitao Liu, Gyanit Singh, Nish Parikh, Neel Sundaresan, "A Large Scale Query Logs Analysis for Assessing Personalization Opportunities in E-commerce Sites", WSCD '2014 New York, New York USA, ACM-2014
[2] D.R. Ingle, P.S. Lokhande, B.B. Meshram, "Web Engineering Model and Quality Issues for Building e-Commerce Applications", International Association of Online Engineering( IAOE), university of Bridgeport, Bridgeport, CT, USA, 2009.
[3] Pang,Vipin Kumar,Micheal, "Introduction to Data Mining", Pearson India, ISBN: 9788131714720, 2007

*Web references*

[4] [APR] Aprori Algorithm http://www3.cs.stonybrook.edu/~cse634/lecture_notes/07apriori.pdf
[5] [MBA] Market Basket Analysis http://www.albionresearch.com/data_mining/market_basket.php
[6] [KDD] Knowledge Discovery http://researcher.watson.ibm.com/researcher/view_group.php?id=144