CHAPTER 5

POTENTIAL USER IDENTIFICATION

The second phase of next web page prediction system focuses on separating the potential users from non-potential users from the preprocessed web log data using machine learning classification algorithm. The growth of Internet has created an environment of abundant consumer choices, where companies must work to increase customer satisfaction. Recent study suggests that increasing retention by as little as 5 percent can mean as much as a 95 percent boost in profit, because return customers generate over twice as much gross income as new customers. Thus, business organizations must be able to predict their customers' behavior, preferences and future needs. The ultimate goal of Web mining is to identify useful information that can help, for example, acquire new customers, retain old customers and grow customers' profitability (Resnick and Varian, 1997).

However, there is no direct relationship between Web log data and the visitors' purchase patterns. The task is even more difficult, when the training data set is small and possibly inaccurate, because a small set of training data is insufficient. To solve this issue, this research has developed a subjective classification approach that provides companies with visitors purchase patterns, which is then used to predict their future browsing habits. The first step to develop such a system is to identify interesting or potential users and promotional plans focused on these users will gain high customer retention. This chapter presents details regarding the algorithm used for this purpose.

5.1. CATEGORIZATION OF WEB USERS

Customers or e-commerce site users can be grouped into three categories. They are, those

(i) Who click and purchase products (Best Potential Customers)

(ii) Who browse for information and familiarize themselves with various products offered by the vendor (Prospective Potential Customers)

(iii) Who browse but never purchase anything (Non-Potential Customers)

In this research, the preprocessed web log data file is analyzed and classified into two groups of users, namely, potential users (including both best and prospective) and non-potential users. Unique characteristics exhibited by customers can be analyzed to divide customers with and without purchase intent. In general, according to Yu *et al.* (2005), various characteristics as listed in Table 5.1 are exhibited by the potential and non-potential users.

TABLE 5.1

CHARACTERISTICS EXHIBITED BY POTENTIAL AND NON-POTENTIAL USERS

Potential Users	Non-Potential Users
• Potential users access certain pages for a long time to read the contents of the page. So, the ratio between the amount of time they need to read contents and the amount of time they navigate from one page to another is large. • They browse down to low-level pages because they need to access specific topics. • They use the HTTP POST mode, because they are interested in recording their information with the web sites by filling out forms from the site. • They access to images and graphic files are more.	• Non-Potential users access more number of pages quickly. They will not much time in reading the contents. The ratio between the time they need to read contents and the time they navigate from one page to another is almost 1. • They don't browse down to low-level pages but rather access a large number of high-level child pages, because they are not interested in any specific topics. • They will not reveal their information. So don't use POST mode. • Less access to images and graphic files.

5.2. CLASSIFICATION

The classification of users into the selected two groups can be performed using any of the available machine learning algorithms. Decision Tree (DT) classification using C4.5 algorithm (Suneetha and Krishnamoorthy, 2010) and Naïve Bayes classification algorithm (Santra and Jayasudha, 2012) are the most frequently used classifiers for this purpose. **DTs** are a non-parametric supervised learning method used for *classification* and *regression*. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. However, while using web log data, the following disadvantages were encountered (http://scikit-learn.org/stable/modules/tree.html, Nayab, 2011).

- Decision-tree learners can create over-complex trees that do not generalize the data well. This is called over fitting,
- Decision trees can be unstable because small variations in the data might result in a completely different tree being generated,
- Decision tree learners create biased trees if some classes dominate,
- Another fundamental flaw of the decision tree analysis is that the decisions contained in the decision tree are based on expectations and irrational expectations can lead to flaws and errors in the decision tree,
- Decision trees are also prone to errors in classification, owing to differences in perceptions and the limitations of applying statistical tools, and
- Exhibits high complexity and the tree might get too large even after applying pruning techniques.

On the other hand, the naïve bayes classifier also faces the disadvantages like over fitting, lowered accuracy when supplied with large datasets and difficulty

to model accurately the dependency characteristics of the potential and non-potential users.

Due to these disadvantages, the research work proposes the use of TSVM (Transductive Support Vector Machine) semi-supervised classifier for classifying the users into the selected two groups. The proposed classifier using TSVM consists of two major tasks, after preprocessing. They are,

(i) Attribute (user characteristics) Identification, and
(ii) Building the Classifier.

5.2.1. Attribute Identification

This task performs attribute identification in two steps. They are

(i) Attribute selection and feature vector creation, and
(ii) Discretization.

The first step of the classification algorithm is identifying a set of attributes that will have high discriminating capacity for differentiating potential and non-potential users. Three types of access attributes are identified for this purpose, from which a set of seven attributes, are identified. This set is used as feature set for training and testing the classifier. The three type of access attributes considered are

(i) Temporal Attributes,
(ii) Page Attributes, and
(iii) Communication Attributes.

Table 5.2 presents the attributes selected from each of the above three types. In the second step, the attribute values thus selected are discretized. Table 5.3 shows the discretized values of the selected attributes.

TABLE 5.2

ATTRIBUTE IDENTIFICATION

Temporal Attributes	A1	Accessing between midnight and 7 a.m.
	A2	The total session time
	A3	Statistics such as the time a visitor accesses the site, the total time a visitor stays at the site and the different amount of time a visitor stays on various pages
Page Attributes	A4	The total number of accessed pages during the whole session
	A5	The accessing width (the number of child pages accessed from a single page)
	A6	The accessing depth (the depth of the pages accessed from a single page)
Communication Attributes	A7	Access Request (such as GET and POST that visitors use to interact with the site

TABLE 5.3

DISCRETIZED VALUES OF THE ATTRIBUTES

Attribute	**Value 0**	**Value 1**	**Value 2**	**Value 3**
A1	No	Yes	-	-
A2	$\leq$ 2min	2-5min	5-15 min	15-30 min
A3	$\leq$ 3 sec	3-30 sec	$\geq$ 30 sec	-
A4	$\leq$ 2 pages	2-5 pages	$\geq$ 5 pages	-
A5	$\leq$ 2 pages	2-5 pages	$\geq$ 5 pages	-
A6	1 hierarchy	2-3 hierarchy	$\geq$ 5 hierarchies	-
A7	Use Get	Use POST	Use Head	-

Using the discretized values as guidelines, a small training set is created manually and is given in Table 5.4. Creation of a large dataset is difficult due to the fact that web logs do not contain much information about whether the user has purchased any product or not. Applying conventional classifiers in these situations is difficult because all these classifiers require large data set with both positive (identifying potential users) and negative (identifying non-potential users) samples. To solve this problem, the argument of Oates and Jensens (1997) is used. According to them, increasing the training set size might not increase accuracy of classifiers. Hence, a small training set is used which pays special attention to potential users (loyal customers), as the aim of this phase is to obtain high accuracy during correct potential customer Identification.

TABLE 5.4

TRAINING DATA

Potential Users	Non-Potential Users
0, 0, 1, 2, 2, 1,0	0,0,0,0,0,0,0
0, 0, 0, 2, 2,2,0	0,2,1,0,0,1,0
0,0,2,1,1,1,1	0,0,1,1,1,1,0
0,3,2,2,2,0,0	0,0,0,1,1,1,0
1,0,0,0,0,0,1	0,3,1,0,0,0,0
0,2,1,2,2,1,0	
1,2,2,1,1,0,0	
1,2,2,2,2,2,1	
1,1,2,2,2,1,0	
1,1,2,1,1,1,1	

5.2.2. Building the Classifier

As mentioned earlier, a TSVM classifier is used to identify the potential and non-potential users in the web log data. The proposed TSVM is designed as a binary classifier. This section first introduces the concept of classification, followed by the steps involved during the design of TSVM.

- **Overview to Classification**

Together with feature extraction, the most crucial step in the process of user identification is classification. Classification, also known as pattern recognition, discrimination, supervised learning or prediction, is a task that involves construction of a procedure that maps data into one of several predefined classes (Montejo-Raez, 2005). It applies a rule, a boundary or a function to the sample's attributes, in order to identify the classes. A classifier works to partition the feature space into decision regions that are identified using pre-defined labels. An efficient classifier should be able to differentiate these partitions with precise decision boundaries (borders between decision regions) (Figure 5.1).

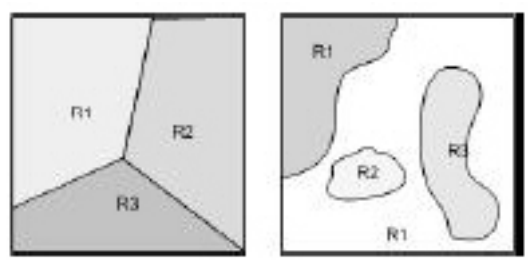


Figure 5.1 : Classifier and Decision Boundaries

The efficiency of a classification technique depends on various factors as follows:-

(i) Whether learning method is a supervised or unsupervised method

(ii) Type of label output (binary or multiple)

(iii) Whether they are statistical or non-statistical in nature.

Examples include Artificial Neural Network (Basheera and Haimeer, 2000), Decision Tree Classifiers (Jenhani *et al.*, 2008), Support Vector Machines (Steinwart, 2002), Naïve Bayes Classifiers (Lu *et al.*, 2010) and Rule-Based Classifiers (Mencar *et al.*, 2011). Each technique employs a learning algorithm to identify a model that best fits the relationship between the attribute set and class label of the input data. The model generated by a learning algorithm should satisfy two conditions as follows:-

(i) It should fit the input data well, and

(ii) It should correctly predict the class labels of records it has never seen before.

For that reason, the primary goal of the learning algorithm is to build models with good generalization capability so as to accurately predict the class labels of previously unknown records. A basic classification model is shown in Figure 5.2.

Figure 5.2 : Basic Classification Model

The input data for a classification task is a collection of features arranged as in row-wise fashion (records). Each record, also known as an instance or example, is characterized by a tuple (X, y) where X is the attribute set and y is a special attribute, designated as the class label (also known as category or target attribute).

Classification is the task of learning a target function f that maps each attribute set X to one of the predefined class labels y. The target function is also known informally as a classification model. Here, a training set consisting of records whose class labels are known must be provided. The training set is used to build a classification model, which is subsequently applied to the test set, which consists of records with unknown class labels.

Classification is the process of finding a set of models (or functions) that describe and distinguish data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown. The derived model is based on the analysis of a set of training data (i.e., data objects whose class label is known). The derived model may be represented in various forms, such as classification (IF-THEN) rules, decision trees, mathematical formulae or neural networks. Out of these the use decision tress representation is more popular as they can be easily converted to classification rules.

Classification can be used for predicting the class label of data objects. However, in many applications, users may wish to predict some missing or unavailable data values rather than class labels. This is usually the case when the predicted values are numerical data and is often specifically referred to as prediction. Although prediction may refer to both data value prediction and class label prediction, it is usually confine to data value prediction and thus is distinct from classification. Prediction also encompasses the identification of distribution trends based on the available data.

A classification model can be unsupervised, supervised or semi-supervised. Unsupervised learning is normally used to locate patterns in the input data. No information is given to the system, which finds the patterns as to the correctness or incorrectness of the patterns. The patterns it finds may therefore be arbitrary or

they may actually be representative of some real underlying process which caused them to appear gives for details on unsupervised classification problem.

Supervised learning or Classification takes a set of known input data and known responses to the data and builds a model that generates predictions for the response to new data. During the process of recognition, the known data (training features) has to be first collected. As mentioned previously, by selecting the appropriate set of features, the performance of the classifier can be improved. These training feature data are then used by the learning algorithm, to mimic the operation of the human brain. The learned knowledge is then applied to a new data (test features) for identification and recognition.

Semi-supervised learning is a combination of supervised learning tasks and techniques that also make use of unlabeled data for training - typically a small amount of labeled data with a large amount of unlabeled data. Semi-supervised learning falls between clustering (without any labeled training data) and classification (with completely labeled training data). The learning accuracy can be improved by many machine-learning algorithms, when using unlabelled data in conjunction with a small amount of labeled data,

Three paradigms can be identified during the classification (Figure 5.3) and are listed below:-

(i) Binary case,
(ii) Multi-class case, and
(iii) Multi-label case.

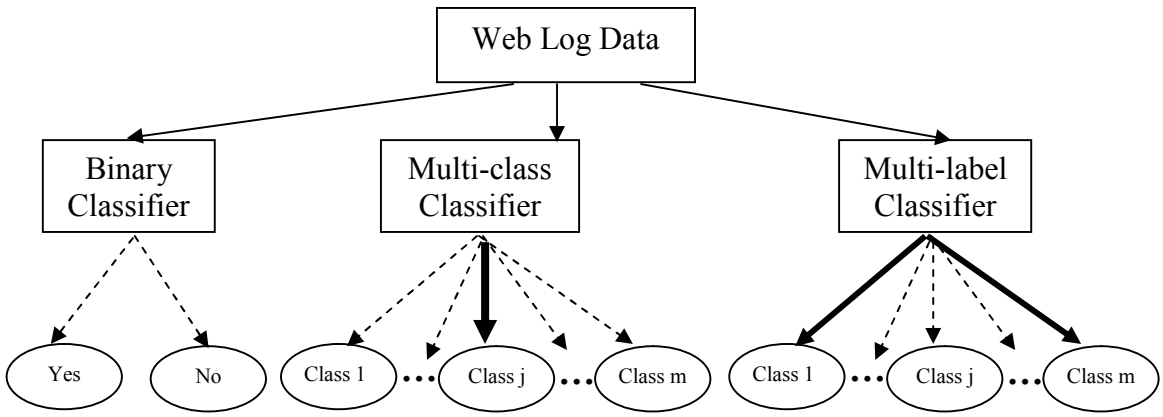


Figure 5.3 : Paradigms in Web Log Data Classification

The binary case classification classifies data into exactly two predefined classes. Here, a sample data belongs exactly to one of the two given classes. The classifier has to determine to which of the two sets the new web log goes (Mehta *et al.*, 2008). In mutli-class case, an web log belongs exactly to just one class of a set of 'm' classes (Foody and Mathur, 2004; Joshi *et al.,* 2009). Finally, in the multi-label case, an data may belong to several classes at the same time, that is, classes may overlap (Li *et al.*, 2004).

In binary classification a classifier is trained, to assign a sample document to one of the two possible sets. These two sets are usually referred to as belonging samples (positive) and not belonging samples (negative) respectively. This method is otherwise termed as the one-against all approach or one-against one approach. This approach is used to group users into potential or non-potential categories.

A classification technique, or a classifier, is a systematic approach to building classification models from an input data set. Examples include, Decision Tree Classifiers, Rule-Based Classifiers, Neural Networks, Support Vector Machines and Naïve Bayes Classifiers.

Each technique employs a learning algorithm to identify a model that best fits the relationship between the attribute set and class label of the input data. The

model generated by a learning algorithm should both fit the input data well and correctly predict the class labels of records it has never seen before. Therefore, a key objective of the learning algorithm is to build models with good generalization capability, i.e., models that accurately predict the class labels of previously unknown records. Figure 5.4 shows a general approach for solving classification problems.

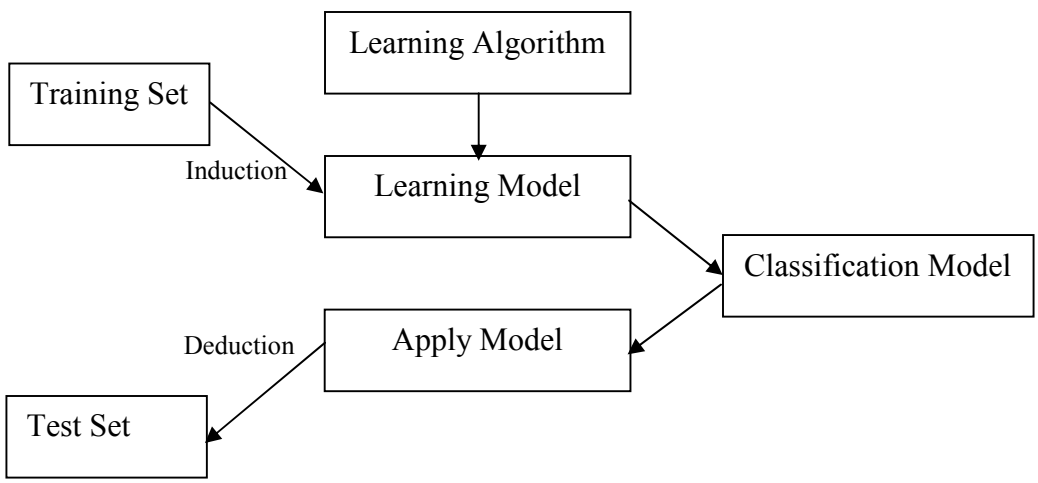


Figure 5.4 : Process of Classification

First, a training set consisting of records whose class labels are known must be provided. The training set is used to build a classification model, which is subsequently applied to the test set, which consists of records with unknown class labels.

The research work proposes the use of TSVM binary semi-supervised classifier to identify the potential users in a web log data file. As TSVM classifier is an extended version of SVM classifier, the following section describes the working of SVM classifier followed by the description of TSVM.

- **Support Vector Machine**

Support Vector Machines (SVMs) were introduced by BenHamza *et al.* (2005) and Ranganath (2005) and have proved to be fast effective classifiers and

works effectively with high dimensional datasets also. A Support Vector Machine (SVM) is a concept in computer science for a set of related supervised learning methods that analyze data and recognize patterns that are mainly used for classification and regression analysis. The standard SVM takes a set of input data and predicts, for each given input, which of two possible classes the input is a member of, which makes the SVM a non-probabilistic binary linear classifier.

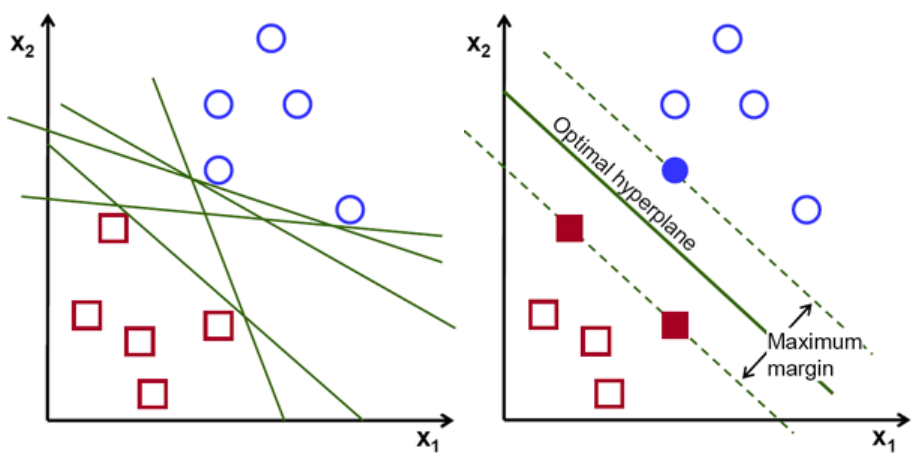
Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

Although SVMs were originally designed as binary classifiers, approaches that address a multi-class problem as a single “all-together” optimization problem exists (Robb, 1999). A multi-class classification task usually involves separating data into training and testing sets. Each instance in the training set contains one ‘target value" (i.e. class labels) and several “attributes" (i.e. features). The goal of SVM is to produce a model (based on the training data) which predicts the target values of the test data given only the test data attributes.

Considering the binary classification case, let $((x_1, y_1) \ldots (x_n, y_n))$ be the training dataset where x_i are the feature vectors that represent the observations and $y_i \in (-1, +1)$ be the two labels that each observation can be assigned to. From these observations, SVM builds an optimum hyperplane (a linear discriminant in the kernel transformed higher dimensional feature space) that maximally separates the two classes by the widest margin by minimizing the objective function. For a linearly separable set of 2D-points which belong to one of two classes, find a

separating straight line is shown in Figure 5.5a. In this example, there exist multiple straight lines that separate the data points into two groups. Deciding the optimal divider is an intuitive criterion.

In general, a line is considered bad if it passes too close to the points because it will be noise sensitive and it will not generalize correctly. Therefore, the goal here is to find the line passing as far as possible from all points. Thus, the operation of the SVM algorithm is based on finding the hyperplane that gives the largest minimum distance to the training examples. Twice, this distance receives the important name of **margin** within SVM's theory. Therefore, the optimal separating hyperplane *maximizes* the margin of the training data. An Example of an optimal hyperplane is shown in Figure 5.5b.



(a) **(b)**

Figure 5.5 : Support Vector Machine Hyperplane

- **Hyperplane Computation**

Let the hyperplane be defined as

$$f(x) = \beta_0 + \beta^T x \qquad (5.1)$$

where β is known as the weight factor and β_0 is the bias. The optimal hyperplane is represented in an infinite number of different ways by scaling of β and β_0.

As a matter of convention, among all the possible representations of the hyperplane, the one chosen is

$$| \beta_0 + \beta^T x| = 1 \tag{5.2}$$

where x symbolizes the training examples closest to the hyperplane. In general, the training examples that are closest to the hyperplane are called **support vectors** and this representation is known as the **canonical hyperplane**. Now, the result of geometry that gives the distance between a point x and a hyperplane $\{\beta, \beta_0\}$ is estimated using Equation (5.3).

$$\text{distance} = \frac{|\beta_0 + \beta^T x|}{||\beta||} \tag{5.3}$$

In particular, for the canonical hyperplane, the numerator is equal to one and the distance to the support vectors is Equation (5.4).

$$\text{distance}_{\text{support\_vectors}} = \frac{|\beta_0 + \beta^T x|}{||\beta||} = \frac{1}{||\beta||} \tag{5.4}$$

Let M denote the margin which is twice the distance to the closest examples (Equation 5.5).

$$M = \frac{2}{||\beta||} \tag{5.5}$$

Now, the problem of maximizing M is equivalent to the problem of minimizing a function $L(\beta)$ subject to some constraints. The constraints model the requirement for the hyperplane to classify correctly all the training examples x_i Formally, it can be defined as Equation (5.6).

$$\min_{\beta,\beta_0} L(\beta) = \frac{1}{2}||\beta||^2 \qquad \text{subject to } y_i(\beta^T x_i + \beta_0) \geq 1 \ \forall i \tag{5.6}$$

where y_i represents each of the labels of the training examples. The training vectors xi are mapped into a higher dimensional space by the function ϕ. SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space. C > 0 is the penalty parameter of the error term. Furthermore, $K(x_i, x_j) = (x_i)^T \phi(x_j)$ is called the kernel function. There are four basic kernels as listed below. Here, γ, r and d are kernel parameters.

1. Linear Kernel : $K(x, x_j) = x_i^T x_j$
2. Polynomial Kernel : $K(x_i, x_j) = (\gamma x_i^T x_j + r)d, \gamma > 0.$
3. Radial Basis Function (RBF) : $K(xi, xj) = \exp(-\gamma||x_i - x_j||^2), \gamma > 0.$
4. Sigmoid Kernel : $K(xi , xj) = \tanh(x_i^T x_j + r).$

- **Transductive Support Vector Machine (TSVM)**

TSVM has been widely used as a means of treating partially labeled data in semi supervised learning. Semi-supervised learning with label-error model was discussed by Amini and Gallinari (2003). Transductive SVM was introduced by Cortes and Vapnik (1995). In semi-supervised learning, sample $X^l, Y^l = \{(X_i, Y_i)\}_{i=1}^{n_l}$ is observed with an independent unlabeled sample $X^u = \{X_j\}_{j=n_l+1}^{n}$ and $n_l + n_u$. Here $X_i = (X_{i1}, \ldots, X_{ip})$ is an p-dimensional input and $Y_i \in \{-1,1\}$, independently and identically $P(x, y)$ and X^u is distributed according $P(x)$.

Transductive support vector machines extend SVMs in that they could also treat partially labeled data in semi-supervised learning by following the principles of transduction. TSVM uses an idea of maximizing separation between labeled and unlabeled data. It solves Equation (5.7).

$$\min_{y_j, f \in F} C_1 \sum_{i=1}^{n_l} L(y_i f(x_i)) + C_2 \sum_{j=n_l+1}^{n} L(y_j f(x_j)) + J(f) \qquad (5.7)$$

where f is a decision function in F, a candidate function class, $L(z) = (1 - z)_+$ is

the hinge loss and J(f) is the inverse of the geometric separation margin. In the linear case, $f(x) = w^T x + b$ and $J(f) = \frac{1}{2} ||w||^2$. In the nonlinear kernel case, $f(x) = (K(x, x_{1+}, \ldots, K(x, x_n))\ w^T + b$, $J(f) = \frac{1}{2}\ w^T K w$, where K is a kernel satisfying Mercer's condition to assure $w^T K w$ with $K = \left(K(x_i, x_j)\right)_{i,j=1}^{n}$ being a proper norm. The algorithm used is presented in Figure 5.6.

Input: training samples $(X_1, Y_1),\ldots,(X_n,Y_n)$ and test samples $X_i \ldots X_j$

Output: predicted labels of the test samples $X_i \ldots X$

Step 1 : Initialization

Set initial value $f^{(0)}$ as the solution of SVM with labeled data alone and an precision tolerance level > 0.

Step 2 : Iteration

At iteration k+1, solve yielding solution $f^{(k+1).}$

Step 3 : Stopping rule

Terminate when $|s(f^{(k+1)}) - s(f^{(k)})| \leq \varepsilon$ Then the estimate $\hat{f}$ is the best solution among $f^{(k)}$ where k=0, .., 1.

Figure 5.6 : The TSVM Algorithm

The classification of Web logs using TSVM can be summarized as follows.

- Identify the set of attributes
- Discretize the attribute values
- Identify the training data set with the help of customers.
- Build a classifier using training samples (with labels).
- Predict the labels of the test records by using the tranductive procedure.
- Calculate precision, recall and accuracy.
- Rerun the classifier by including the test records also.
-

5.3. CHAPTER SUMMARY

In this chapter, a TSVM based classification algorithm was proposed to group the web log data into two groups of customers. The first group is potential buyers and the second group is non-potential buyers. After grouping the customer log data, the non-potential buyers are removed, thus reducing the size of web log data file. The reduced web log data file is then used to predict the respective users' future web requests. The method used for prediction is described in the next chapter, Chapter 6, **Web Log Associative Classification**.

CHAPTER 6

WEB LOG ASSOCIATIVE CLASSIFICATION

With the proliferation and growth of e-commerce, Personalized recommendation systems, web-based information systems and web services have evolved as a vital application which is pre-requisite to the success of a website. It is nowadays common for Web browsers to face web sites that provide online recommendations of products and services, personalized link selections and banner advertising. This phenomenon is nowhere relevant to "business-to-consumer" e-commerce area. The reason for this is that, in today's very competitive e-commerce arena, the success of a web site is highly dependent on the site's ability to preserve visitors and turn browsers into reliable customers. To bring more users towards a web site, critical tools such as Automatic personalization and recommender system technologies need to be invoked in the site.

Traditional approaches used for these purpose are of two types. They are content-based and user-based techniques. Content-based approach use personal profiles of users and recommend other information based on their content similarity to the information in the user's profile. These systems perform well from the user's perspective, where the user is searching the Web for information. But they are less useful in e-commerce applications, due to the lack of server-side control by site owners and techniques based on content similarity alone may leave out other types of semantic relationships among objects.

User-based techniques, on the other side, mainly focus on the similarities among users rather than item-based similarities. User-based techniques compare the record with historical data of other users with similar interest. This technique is also called 'neighborhood of the current user'. The neighborhood mapping is based on access to identical content and pages, similarity in ratings of items, or

purchase/sale of similar items. The neighborhood identified is used to suggest items / pages not already purchased or sold / accessed by the active user. The advantage of user-based method over content-based approaches is that it can confine "pragmatic" relationships between items based on their proposed use or based on similar interest of the users.

However, they also suffer from a few well-known limitations (Sarwar *et al.*, 2000). Recommender systems and Neighborhood formation need to be developed to resolve the limitations with respect to scalability and efficiency. Web usage mining algorithms enhances this issue by using collaborative filtering.

The aim of web mining is to discover behavioral models and patterns, not only within the web pages of the site but also between pages of other Web resources. Thus, the goal is to depict and model behavioral patterns of the web user.

However, these pure usage-based approaches have an important demerit, that is, the recommendation process relies on the existing user transaction. Hence the items or pages added to a site recently cannot be suggested and this is commonly referred to as the "new item problem". A general approach to resolve this problem is to integrate content characteristics of pages with the user ratings or judgments (Claypool *et al.*, 1999; Pazzani, 1999). For this purpose, keyword-based approaches are used. These approaches extract keywords from the contents on the website and use them to either index or classify web pages. This approach would allow these systems to recommend pages to a user, not only based on similar users, but also based on the content similarity of these pages to the pages, the user has already visited.

Keyword-based approaches, however, are incapable of capturing more complex relationships among objects at a deeper semantic level based on the inherent properties associated with these objects. To be able to recommend

different types of complex objects using their underlying properties and attributes, the system must be able to rely on the characterization of user segments and objects, not just based on keywords, but at a deeper semantic level using the domain ontologies for the objects. An ontology defines a set of well-founded constructs that recommands significant concepts and their semantic relationships.

The ontology of a site can be built by extracting relevant concepts and relations from the content and structure of the web site, by using machine learning and Web mining techniques. Web usage related concepts and relations are also needed, in addition to concepts and relations that can be acquired from Web content and structure information for effective construction of ontology. In an E-commerce Web site, the relations between users and objects which define various types of online activity, such as e-business, searching for items, user registration, buying and selling may be of interest. The combination of such usage-based relations with ontological information, allows for more efficient knowledge discovery and pattern extraction algorithms.

In Dynamic recommender systems, the use of semantic knowledge can enhance the interaction of the browsers with the web site. Domain knowledge integrated with the web recommender systems allow the user to navigate in the proper direction.

Human emotions are a significant factor of human behavior in web mining analysis (Dolan, 2002). This fact is well-supported by Weinberg and Gottwald (1982) and Piron (1993), who studied the relationships between consumer emotions and their buying behaviors. Study of user emotions and behavior has been found to be more important in many web-based applications like personalization and recommendation.

A backbone of semantic web is ontologies which at present are often hand-crafted. The challenge is to learn ontologies and/or instance of their concepts, in a

semi-automatic or automatic way by using web mining techniques such as association rules. On the other hand, background knowledge ontologies, can be used to enhance the results of effective web mining.

Phase III of next web page prediction system focuses on the actual prediction of future page requests of users who access a website. For this purpose, this research work proposes an associative rule based semantic web usage mining algorithm that integrates human emotions and behaviors through self-reporting and behavioral tracking. The main advantage of such a system would be that it can improve the results of web usage mining by exploiting semantic structure in the web and to make use of web mining for building a semantic web. The proposed method aims at mining semantics from semantically enriched web usage logs and to create personalized web usage ontology for the Semantic Web.

This method mines periodic access patterns, which occur frequently in a particular period, e.g., every morning, directly from web usage logs that have been semantically enriched with information on topics and emotional influence. Such periodic access patterns are very useful for mid to long-term behavioral tracking. With the periodic web access patterns of a user, the resources that the user is most probably interested in during a specific time period can be deduced and prepared without knowing the user's current web access information for web personalization services.

Over time, the ontology will accumulate personal information on web access behavior and habits, as well as the emotional influence of the accessed resources. So, those who have concerns about privacy may choose not to use our system. Many ontology generation techniques, investigate and focus focus mainly on generating concept hierarchy for building ontology from free text documents or relational databases (De Maio *et al.*, 2009a; 2009b; Xu *et al.*, 2010).

The proposed approach uses methods proposed by Fong *et al.* (2012) to extract semantics from semantically enriched web usage logs automatically. The approach enhances this method by introducing Particle Swarm Optimization technique to identify the optimum session interval. This chapter presents the related materials and the working of the proposed method.

6.1. SEMANTIC WEB MINING

"Web mining is the process of discovering and extracting useful knowledge from the content, usage and structure of one or more Web sites". Semantic Web mining (Berendt *et al.,* 2002) includes the integration of domain knowledge into the Web mining process. This technology is now applied to adopt a different method of how data is collected, deposited and analyzed (Lavrač *et al.*, 2011). Semantic web mining is a field of research that "combines semantic and ontological knowledge into the process of Web usage mining". One of the most important application area is Web personalization and recommender systems.

Since traditional web usage logs only record requested URLs but not the semantics of contents requested by the users, it is difficult to use such logs for tracking the users' actual web access behaviors, emotions and interests. In response, a number of semantic web usage mining techniques (Stumme *et al.*, 2002) have been proposed. Dai and Mobasher (2002) used domain ontology to enhance web usage mining for traditional web usage logs, but the mapping from requested URLs to ontological entities lacks reliability, especially for dynamic websites. Oberle *et al.* (2003) proposed another framework for semantic enrichment of web usage logs by mapping each requested URL to one or more concepts from the ontology of the underlying website. It clusters groups of sessions with specific user interests from the semantically enhanced weblogs and applies association rule mining to the semantically enhanced weblogs. Eirinaki *et al.* (2003) obtained concept-logs (C-logs) by enriching each webserver log record

with keywords from a taxonomy representing the semantics of the requested URLs. C-logs were analyzed by Meo *et al.* (2004) with MINE RULE (a query language for association rule mining) for discovering access patterns. Also, Fraternali *et al.* (2003) created conceptual logs by combining the server log data with the conceptual schema of the web application.

Most semantic web usage mining techniques focus only on discovering simple usage statistics and common access patterns of user groups. Further, the discovered knowledge should be represented as ontology to enable Semantic Web applications. The two main tasks of semantic web usage mining are given below.

- Apply Web Mining techniques to understand ontologies from vast source if unstructured documents in the web.
- Define ontologies for existing and future data to make search more precise.

An ontology is a formal explicit description of concepts in a domain of discourse, properties of each concept describing various features and instances of the concept. An ontology together with a set of individual instances of classes constitutes a knowledge base.

"An ontology consists of a finite list of terms and the relationships between these terms. The terms denote important concepts (classes of objects) of the domain, while the relationships include hierarchies of classes". Ontologies may also include other information, such as properties, value restrictions, disjointedness statements and specifications of logical relationships between objects. Ontology languages are semantic markup languages for defining ontologies. This research work uses Web Ontology Language (OWL) (McGuinness *et al.*, 2004), which was proposed as W3C Recommendation, for ontology specification. OWL facilitates greater machine interpretability of web

content than XML (Yergeau *et al.*, 2004), RDF and RDF schema (Klyne and Carroll, 2004).

Ontologies can be constructed manually using an ontology editor, e.g., Protege (Noy and Mcguinness, 2001) and OntoEdit (Sure *et al.*, 2002), but the process can be tedious. The integration of knowledge acquisition with machine learning facilitates research toward automating the ontology generation process. Many approaches have been investigated for generating ontology (Maedche and Staab, 2001). These include Natural Language Processing (NLP) techniques (Todirascu *et al.*, 2000), association rule mining (Maedche and Staab, 2000), hierarchical clustering (Clerkin *et al.*, 2001), translation from relational databases (Ma *et al.*, 2008) and Formal Concept Analysis (FCA) (Cimiano *et al.*, 2003; Quan *et al.*, 2006). However, these techniques focus mainly on constructing concept hierarchies from text documents or relational databases.

Protege OWL is an open source tool created to support ontology development for the Semantic Web. It is a plug-in extension to the Proteg ontology development platform. Protege OWL allows users to edit ontologies in the Web Ontology Language (OWL) and to use description logic classifiers to maintain consistency of their ontologies. Protege OWL can also assist developers of intelligent applications in biomedicine, because many of the problem-solving tasks they seek to automate can be construed as classification tasks.

Protege OWL provides access to emerging knowledge representation standards such as OWL and high-performance classifiers. Being integrated with Protege, the OWL Plug-in allows users to exploit Protege's core features and services such as graphical user interfaces, a variety of storage formats and data acquisition and visualization tools. Finally, Protege OWL provides an API allowing it to be integrated into applications.

OWL has emerged as a standard language for representing knowledge in the Semantic Web. OWL is also based on description logics and it supports automated reasoning. If an intelligent application is amenable to being posed as a classification task, then OWL provides the advantage of a standard knowledge representation language that can encode both the domain knowledge as well as reasoning knowledge in the form of axioms and class definitions. Protege OWL has been successfully deployed for the last two years. It is implemented in Java and it runs on a broad range of hardware platforms. It has an extremely active community of hundreds of users and it is becoming the de-facto standard OWL editor.

6.2. RELATED TECHNIQUES

The proposed method uses fuzzy rules and periodic pattern mining during the design of prediction system. This section introduces the concept behind these topics.

6.2.1. Fuzzy Association Rule Mining

Association rules (Lin *et al.*, 2002; Moreno *et al.,* 2004), can help discover relationships between web resources accessed by a user that would otherwise be missed, especially if the resources are disjoint. They can also be used to find groups of people with similar interests. A major problem of traditional association rule mining techniques is that each item in a transaction is considered only to either exist or not. Thus, the user's preference and interest in each transaction item cannot be precisely represented. Since the concepts of preference and interest are fuzzy data, fuzzy logic (Zadeh, 1975) can be applied. For example, Wong *et al.* (2001) combine fuzzy association rule mining and case-based reasoning (CBR) (Shiu and Pal, 2004) to improve the quality of web access pattern prediction. The fuzzy rule set was found to perform better in prediction accuracy and rule coverage than traditional rule set.

6.2.2. Periodic Pattern Mining

Discovering periodic patterns from time series databases is an important data mining task for many applications, such as behavioral tracking. According to the type of patterns, periodic patterns can be divided into periodic association rules and periodic sequential patterns. Periodic association rules are rules that associate with a set of events that occur periodically; such association rules hold only during certain time intervals but not others. Periodic sequential pattern mining can be viewed as an extension of sequential pattern mining (Agrawal and Srikant, 1995) by taking into account the periodic characteristics in the time series data.

6.2.3. Lattice

As lattices form a vital role during the design of OSIPSO algorithm, this section gives an introduction to that topic. In lattice theory (Wille, 1982), a formal context is defined as a structure that portrays the relationship between structure and attributes in a domain. Using this, it is possible to general formal concepts and interprets them into its corresponding concept lattice. The lattice is used represent the concept hierarchy. The following definitions are used (Vasumathi and Govardha, 2009).

A formal context is a triple K = (G, Mp, Mr I) where G is a set of objects, Mp is a set of period attributes, Mr is the set of resource attributes. I is a fuzzy set of the domain to denote the associations between user access sessions and attributes. Each fuzzy relation $R(g, m) \in I$ is represented by a membership value $\mu(g, m) \in [0,1]$. Each user access session $g \in G$ can also be represented as a fuzzy set on the domain. For a periodic attribute $m_p \in M_p$, the membership value $\mu_p(g, m_p)$ in a user access session $g \in G$ can be computed using the period of g. Also, for a resource attribute, the membership value $\mu_r(g, m_r)$ in a user access session $g \in G$ can be computed using the total duration of m_r. If $Z(g, m_r)$ is less than the $Z(m_r)$ of the user the membership of the resource attribute is zero. If

$Z(g, m_r)$ is greater than $Z(m_r)$ membership of the resource attribute is one. $Z(m_r)$ is nothing but the proportion of the total duration of accessing the resource in all web access sessions of the user, which denotes the user's global interest of the resource. $Z(g, m_r)$ is the proportion of the duration of accessing the resource within the user access session g, weighted by an emotional influence factor e_r derived from the consequent ΔE using one of the following rules:

Rule 1 : -E: This is the baseline situation where $e_r = 1$; ignoring emotional influence.

Rule 2 : $+E_1$. If $\Delta E < 3$ then $e_r = 0.5$; otherwise $e_r = 1$; we only repress resources with negative emotional influence.

Rule 3 : $+E_2$. e_r is derived by using the formula, $e_r = 0.1\Delta E + 0.7$, thus the larger value is assigned to e_r with increasing ΔE, i.e., e_r =0.8,0.96,1,1.1,1.2

$z(g, m_r)$ denotes the user local interest and emotional influence of the resource. For a given web usage context $K = (G, M_p, M_r, I)$ the set of common to user access sessions are defined as $A \subseteq G$ and the set of user access sessions are defined as $B \subseteq M_p$. The fuzzy support of set of attributes is defined as how frequently the set of attributes common to user access sessions and set of user access sessions which have all the same attributes. The fuzzy confidence is defined as number of times the user access the similar resource attributes in the set of user access sessions and the set of resource attributes frequent to user access sessions.

$I \subseteq G \times M$ is a binary relation between G and M. An object g in a relation I with attribute m is denoted as (g, m) $\in$ I and read as "the object g has the attribute m".

A formal context is generally represented using a 2-dimensional table format, where the rows represent the object name and columns represent the attributes. A cell in row ‘g’ and column ‘m’ indicates that the object ‘g’ has attribute ‘m’.

Consider the example shown in Figure 6.1, which consists of 2-dimensional table having 3 web pages (W1, W2 and W3) and three attributes (A1 = ‘Equipment’, A2 = ‘Doctor’ and A3= ‘Thermometer’). As mentioned previously, the symbol ‘X’ indicates that a web page (w_i) has the corresponding attribute (a_i).

	A1	A2	A3
W1	X		
W2	X	X	
W3			X

Figure 6.1 : 2-Dimensional Representation of Concept Relationship

Given a formal context (G, M, I), A' = {m ∈ M | ∀ g ∈ A: (g, m) ∈ I } for a set A⊆ ⋅G (the set of attributes common to the objects in A) and B' = {g ∈G | ∀ m ∈ B: (g, m) ∈ I} for a set B ⊆ M (the set of objects which have all attributes in B).

A formal concept of a formal context (G, M, I) is a pair (A, B) with A ⊆ G, B ⊆ M, A’ = B and B’ =A. The sets A and B are called the extent and intent of the formal concept (A, B) respectively.

Let (A1, B1) and (A2, B2) be two formal concepts of a context (G, M, I). Then, (A1, B1) is called the sub concept of (A2, B2), denoted as (A1, B1) £ (A2, B2), if the relationship, A1 ⊆ A2 (⇔ B2 ⊆ B1) exist, else (A2, B2) is called the super concept of (A1, B1). The relation £ is called the hierarchical order

of the formal concepts. The set of all formal concepts of (G, M, I) ordered in this way is called the concept lattice of the formal context (G, M, I) which is denoted as ℜ(G,M,I). Figure 6.2 shows the lattice example -1.

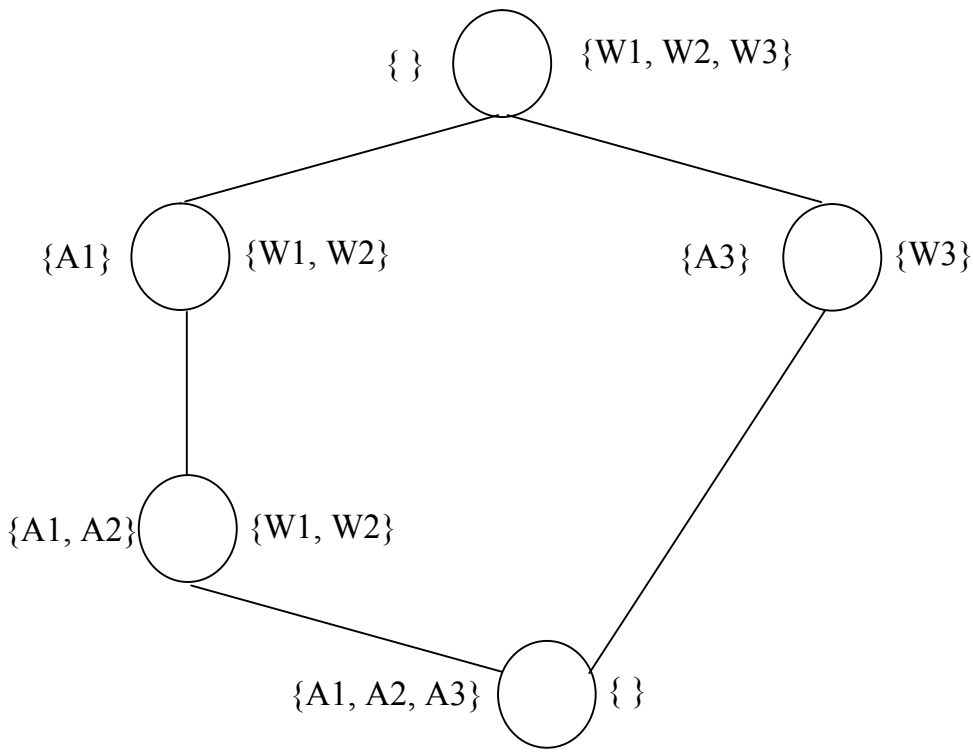


Figure 6.2 : Lattice Example

The concept lattice thus generated is a complete lattice, with one the concept at lower bound called minimum and one concept at upper bound called super mum. Figure 6.2 also shows the intent (given at the left of each node) and the extent (given at the right of each node) of every concept. For example, the intent and extent of the concept at the top left node are {Equipment} and {W1, W2} respectively.

6.3. OSIPSO ALGORITHM

The steps involved in the OSIPSO are shown in Figure 6.3. The algorithms used in the initial steps, that is, preprocessing and potential user identification, are explained in the previous two chapters. As the proposed algorithm converts the

web log data into a semantic data, the first step is the ontology generation, which will then be used during lattice creation.

Web logs are semantically enriched by associating each requested URL with one or more ontological entities, such as concepts, attributes and relations, to better describe the patterns of web navigation. The framework proposed by Fong *et al.* (2012) is adopted by the proposed method and it is assumed that each requested URL can be annotated with pertinent semantic information (topics, concepts, etc.) manually or semi-automatically.

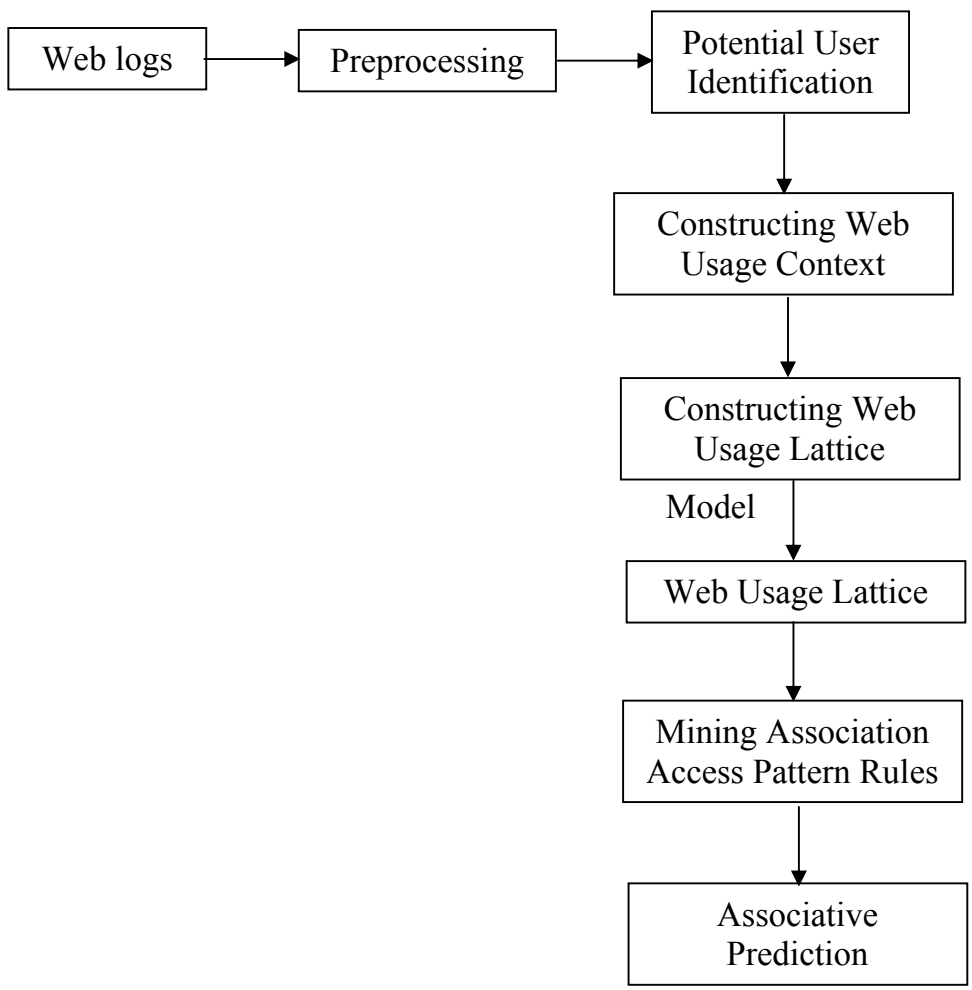


Figure 6.3 : OSIPSO Procedure

Each log entry is annotated with one or more predefined topics (e.g., News, Sports, etc.) and an emotional influence score ΔE reported by the user. Specifically, at the end of each web access activity (just before the next request) the user is asked to score how the web content has influenced their emotions on a scale of 1 to 5 (highly negative, negative, neutral, positive, or highly positive). Set at 3 (neutral) by default, if no new ΔE is recorded, it is assumed to be the same as the most recent score. Table 6.1 illustrates an example of a semantically enriched weblog.

TABLE 6.1

SEMANTICALLY OPTIMZIED WEB USAGE LOG

UserID	Timestamp	URL	ΔE	Topics
User1	21/May/2010 08:20:01	URL1	3	#Topic2, #Topic3,...
User1	21/May/2010 08:22:32	URL2	3	#Topic7, #Topic5,...
User2	21/May/2010 08:22:50	URL7	4	#Topic1, #Topic8,...
User1	21/May/2010 08:27:30	URL3	2	#Topic3, #Topic1,...
User3	21/May/2010 08:33:10	URL3	1	#Topic6, #Topic2,...
User1	21/May/2010 09:10:02	URL5	4	#Topic7, #Topic3,...
User3	21/May/2010 09:17:32	URL6	3	#Topic2, #Topic1,...
User2	21/May/2010 09:26:17	URL4	5	#Topic3, #Topic7,...

Each entry in Table 6.1 can be interpreted as "User N accessed specific resources at a specific time and was emotionally influenced by a specific amount." If the user has accessed specific resources periodically, then the user is said to have web access activity (i.e., periodic web access pattern). Therefore a set of periodic attributes and resource attributes are used to represent web access activities. Eight real-life temporal concepts are used. They are, Early Morning, Late Morning, Noon, Early Afternoon, Late Afternoon, Evening, Night and Late Night, as periodic attributes.

The next step is to construct the web usage context. This step will construct the web usage context (the formal context of web logs) based on all web access

sessions obtained from the preprocessing step. Let M be a set of unique access events, which represents web resources accessed by users, i.e. web pages, URLs, or topics. A web access session can be denoted as $S = \{(e_1, t_1), (e_2, t_2), \ldots, (e_n, t_n)\}$, Where $e \in M$ and t_i is the request time of e_i for $1 \leq i \leq n$. Here, repetition of items is allowed. The web usage context can then be constructed using the above information. Figure 6.4 shows an example of a web usage context. Each row represents a web access session (S) and each column represents a web page access (PA).

	PA1	PA2	PA3	PA4	PA5	PA6
S1	X	X			X	X
S2		X	X			X
S3			X	X		X
S4	X	X	X	X		X
S5			X	X		

Figure 6.4 : Example of Web Usage Context

The next step of OSIPSO is the web usage lattice creation. The semantic web log file created initially is used in this step. To build this optimized semantic web log file, the method first identifies a set of periodic attributes (i.e., temporal concepts such as morning) and a set of resource attributes (i.e., useful domain ontological concepts) enhanced with user-reported emotional influence to represent periodic pattern-based web access activities. With the user's web access activities, it constructs a Personal Web Usage Lattice (PWUL) from the access sessions of the user. The method of constructing lattices was explained before. The lattice constructed for the example shown in Figure 6.4 is given in Figure 6.5.

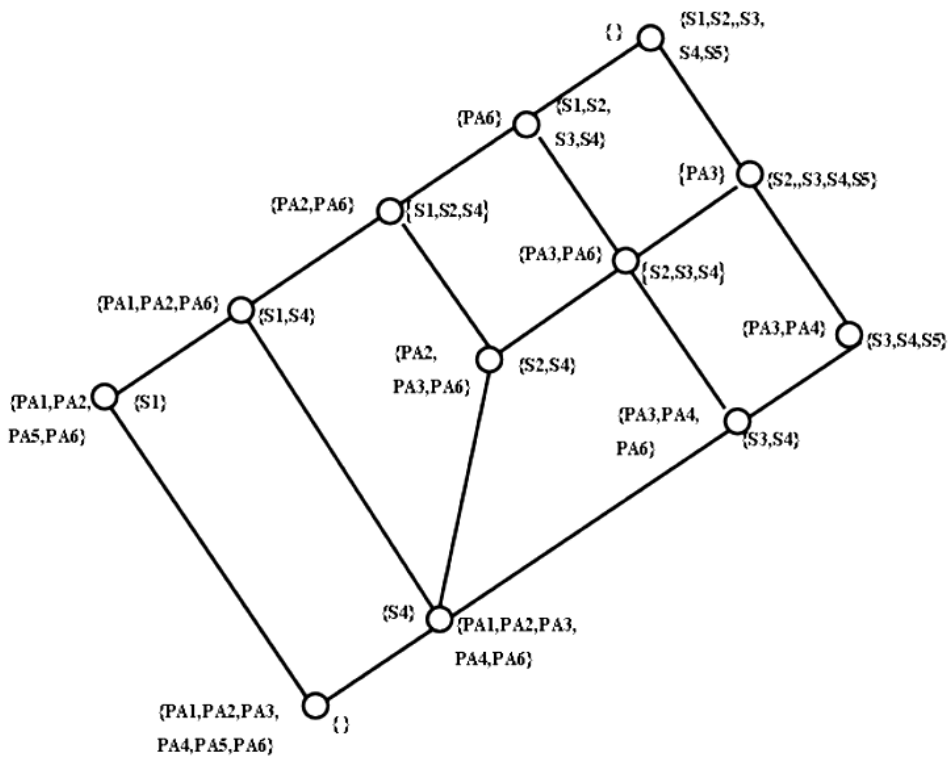


Figure 6.5 : Lattice construction

In a similar fashion, global web usage lattices are also constructed. To create the global web usage lattice, the set of selected periodic attributes M_p and resource attributes M_r is selected for all users. $W_G = \{B_k\}$ represents the set of all web access activities. $|W_G|$ is the total number of global web access activities.There should be a total of $\left(\frac{a}{i}\right) \times \left(\frac{b}{j}\right)$ global web access activities with i periodic attributes and j resource attributes, where $\frac{a}{i}$ and $\frac{b}{j}$ represent the number of combinations. Each web access activity has direct sub activities. The direct sub activity relationships are used to create a Global Web Usage Lattice.

The personal web usage ontology is generated by combining the personal web usage lattice of a user with the global web usage ontology by using the concept of instance mapping. Instance mapping creates a set of activity instances of the corresponding activity classes in the global web usage ontology and an

activity instance hierarchy. The algorithm for generating personalized ontology is shown in Figure 6.6.

Finally, the user satisfaction level is determined. The satisfaction for the overall web personalization is defined as,

$$\text{Satisfaction} = \frac{\sum_{PR_i \in PR_a} \text{satisfaction}(PR_i)}{|PR_a|}$$

where, PR_a is a subset of personalized resources. PR_i is set of personalized resources. The evaluation of satisfaction is used to measure how well the user is interested in the personalized resources.

In this stage, to improve the process of session interval based prediction system, this study introduces Particle Swarm Optimization to optimize the process of session intervals selected. The aim is to improve the satisfaction level of the user along with the accuracy of prediction.

Input: Set of weblogs Output: Satisfaction level of user

1. Take set of Web Usage Logs

   Pre-processing in weblogs

2. Sequence of web access session S = < (URL_1, t_1), (URL_2, t_2), …, (URL_n, t_n) >

3. Estimate duration $d_i = (t_{i+1} - t_i)$

4. Every URL_i has set resource attributes $M_{ri} \subseteq M_r$

5. $d(S, m_k) = \sum_{i=1}^{n} \alpha_{ki} d_i$, where // Estimation of level of interest

6. $\alpha_{ki} = \begin{cases} 1, & \text{if } m_k \in M_{ri} \\ 0, & \text{otherwise} \end{cases}$

7. $\mu(g, m) = \begin{cases} \mu_p(g, m), & \text{if } m \in M_p \\ \mu_r(g, m), & \text{if } m \in M_r \end{cases}$ // Membership function

   // where G =set of user access sessions, M_p = set of periodic attributes, M_r= set of resource attributes, I = fuzzy set of the domain

8. $\mu_p(g, m_p) = \max_{t \in p(g)}\{\mu_p(t, m_p)\}$, where $\mu_p(t, m_p)$ // Membership function of periodic attribute

9. $\mu_r(g, m_r) = \begin{cases} 0, & \text{if } z(g, m_r) < \frac{1}{2} Z(m_r) \\ \frac{2z(g,m_r)}{Z(m_r)} - 1, & \text{if } \frac{1}{2} Z(m_r) \le z(g, m_r) \le Z(m_r) \\ 1, & \text{if } z(g, m_r) > Z(m_r), \end{cases}$ // Membership function of resource attribute

10. $Z(m_r) = \frac{\sum_{g_k \in G} d(g_k, m_r)}{\sum_{g_k \in G} (te(g_k) - ts(g_k))}$ // Total duration of accessing resource

11. Generate Periodic association pattern

12. $Sup(B) = \frac{\sum_{g \in B} (\mu_p(g) \times \mu_r(g))}{|G|}$ // Fuzzy support value

13. $Conf(v(B)) = prob\left((B \cap M_r) \middle| (B \cap M_p)\right) = \frac{Sup(B)}{Sup(B \cap M_p)}$ // Fuzzy confidence value

14. // Creation of global web usage lattice

15. Global and personal web usage Ontology generation

Figure 6.6 : Generation of personalized Ontology

The optimum session interval is identified by using particle swarm optimization algorithm. The particle swarm optimization is a computational

method which optimizes a problem by continuously trying to enhance a candidate solution with regard to a given measure of quality. In every iteration process, each candidate solution is calculated by the objective function being optimized, deciding the fitness of that solution. Every particle preserves its position, composed of the candidate solution and its evaluated fitness and its velocity. Furthermore, it considers the best fitness value it has accomplished thus far during the process of the algorithm, referred to as the individual best fitness and the candidate solution that achieved this fitness, referred to as the individual best position. At last, the PSO algorithm maintains the best fitness value accomplished among all particles in the swarm, called the global best fitness and the candidate solution that achieved this fitness, called the global best position or global best candidate solution. The PSO algorithm includes three major steps as listed below.

1. Compute the fitness of every particle
2. Update individual and global best fitness and positions
3. For every particle update velocity and position

The session interval based algorithm optimized with the inclusion of PSO is presented in Figure 6.7.

The final step in the OSIPSO algorithm is the mining of associative access pattern rules. The first studied approach in the field of frequent pattern mining is based on association rules (Park *et al.*, 1995; Agrawal *et al.*, 1996). Association rules mining was first proposed to find all the rules in the transaction data to analyze the relationship between items purchased by customers in a shop. Association rule mining can be stated as follows.

Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of items. Let 'D' be a set of transactions, where each transaction 'T' is a set of items such that $T \subseteq I$. An association rule is an implication of the form, $X \rightarrow Y$, where $X \subset I$, $Y \subset I$ and $X \cap Y = \emptyset$. The rule $X \rightarrow Y$ holds in the transaction set T with confidence 'c', if c% of transactions in

T that support ‘X’ also support ‘Y’. The rule has support ‘s’ in T if s% of the transactions in T contains X ∪Y.

1. Initialize N number of particles in the swarm, each particle having a position X_i and velocity V_i. Let $pbest$ be the best known position of particle i and g_{best} is the best known position of the entire swarm
2. Initialize the particle's position X_i
3. For each particle i=1,2,…N
    a. Calculate fitness value for every particle

    // For fitness calculation satisfaction is taken from the previous algorithm

    $$Fitness = \frac{Satisfaction}{Number\ of\ user\ sessions \times Number\ of\ particles}$$

    b. If fitness value is better than the best fitness value (pBest)

    c. Set current value as the new pBest

    Until a termination criterion is met
4. Select the particle with best fitness value of all particles as the gbest
5. For every particle

    // Calculation of particle velocity

    a. $V_i(t+1) = wv_i(t) + c_1r_1[\hat{x}_i(t) - x_i(t)] + c_2r_2[g(t) - x_i(t)]$ //where, the index of the particle is represented by i, $v_i(t)$ is the velocity of particle i at time t, $x_i(t)$ is the position of particle i at time t, parameters w, c1 and c2 are coefficients

    b. Update particle position - $x_i(t+1) = x_i(t) + v_i(t+1)$

    Until some stopping condition is met

Figure 6.7 : OSIPSO Algorithm

Given a set of transactions D (the database), the problem of mining association rules is to discover all association rules that have support and

confidence greater than the user-specified minimum support (called minsup) and minimum confidence (called minconf). An association rule 'r' is a relation between itemsets of the form

$$r: X \Rightarrow (Y - X)$$

where X and Y are frequent itemsets and $X \subset Y$. The itemsets X and (Y – X) are called, respectively, antecedent and consequence of the rule 'r'. The valid association rules are those for which the measure of support and confidence, is greater than or equal to the minimal thresholds of support and confidence, called minsup and minconf (Webb, 2000). Support and confidence are calculated as in Equation (6.1) and (6.2).

$$\text{Support}(X) = \frac{|\{t \in D \mid X \subseteq t\}|}{|D|} \tag{6.1}$$

$$\text{Confidence}(r) = \frac{\text{Support}(Y - X)}{\text{Support}(X)} \tag{6.2}$$

A typical association rule is of the form

$$A \Rightarrow B, C \; [\text{ Support} = 60\%, \text{Confidence} = 80\%]$$

"50% of visitors who accessed URLs B and C also visited A"

The process of finding all the association rules with confidence and support above the respective thresholds is a two stage process. The first stage consists in finding all sets of items with support above the support threshold; these sets are called large item sets. The second step consists in computing for each large itemset the confidence for all its expressions with the form of a rule; the expressions whose confidence is above the confidence threshold are the rules. The second step of the process is trivial, therefore, the problem of mining association rules can be viewed as the problem of finding all the item sets with support above a given threshold.

Most traditional association rule mining algorithms employ a support-confidence framework. However, such approaches suffer from the same problem in which a large number of rules are usually returned. In fact, there are many redundancies among the returned association rules. In other words, despite using the minimum support and confidence thresholds to help weed out or exclude the exploration of uninteresting rules, many rules that are not interesting may still be produced. Mining association rules using Formal Concept Analysis can significantly reduce the number of rules without compromising on the quality. This approach extracts only a small subset from all association rules, called basis, from which all other rules can be derived. \

During discussion, the following terminologies are used. Given a set of items $I = \{I_1, I_2, \ldots, I_m\}$ and a database of transactions $D = \{t_1, t_2, \ldots, t_n\}$, where $t_i = \{I_{i1}, I_{i2}, \ldots, I_{ik}\}$ and $I_{ij} \in I$, an association rule is an implication of the form $X \Rightarrow Y$ where $X, Y \subset I$ are sets of items called itemsets and $X \cap Y = \varnothing$. X is called the antecedent and Y is called the consequent. Given a formal context K = (G, M, I), the association access pattern rules consist of two kinds of rules:

- Exact rules B1 => B2, where B1 and B2 are frequent non-empty concept intents and the concept (B1', B1) has concept (B2', B2) as its only immediate super concept
- Approximate rules B1 => B2, where B1 and B2 are frequent nonempty concept intents and the concept (B1', B1) is an immediate subconcept of (B2', B2).

From the concepts of association rule mining, given the minimum support (MinSup) and minimum confidence (MinConf), for all rules B1 => B2 (B1, B2 $\neq \varnothing$), sup(B1 => B2) ≥ MinSup and conf(B1 => B2) ≥ MinConf.

From the above definition, it is obvious that each exact rule corresponds exactly to one edge that connects the subconcept with its only super concept in a

concept lattice, and each approximate rule corresponds exactly to one edge that connects the super concept with one of its sub concepts in the concept lattice. For example, in the web usage lattice shown in Figure 6.5, the edge from the concept node {PA2, PA6} to {PA6} represents an exact rule PA2=>PA6 with support = 60% and confidence = 100%, and the edge from the concept node {PA6} to {PA2, PA6} represents an approximate rule PA6 =>PA2 with support= 60% and confidence = 75%. The computation for the support and confidence was described previously. The algorithm for mining association rules from lattices is given in Figure 6.8.

1. Construct local, global web usage lattices and global web usage ontology
2. Obtain the set of concept nodes, NL = {N1, N2, ..., Nm} where N_i = {A_i, B_i, P_i} where A_i is the extent of N_i, B_i is the intent of N_i and P_i is the immediate parent node of N_i.
3. Estimate minsupport and minconf.
4. Generate associative patterns of the form ARS= {AR1, AR2, ..., ARn} where ARi is the association rule of the form (Xi => Yi, support, confidence). This is performed using Steps a-.
    a. Let ARS= {}
    b. For each $N_i \in N_L$, if PAi= $\varnothing$ and supp >= minsupp do
        i. if |PAi|=1 and |Bi1= $\varnothing$ do
            1. Insert ((Bi-Bi1)=>Bi1, sup, 100%) into ARS as exact rule
        ii. For each $N_{IJ} \in$ PAi, if BIJ $\neq \varnothing$ and conf >= minimum confidence do
            1. Insert ((Bij=>(Bi-Bij), sup, conf) into ARS as approximate rule
    c. Return ARS

Figure 6.8 : Associative Access Pattern Rule Mining

6.4. CHAPTER SUMMARY

This chapter presented details regarding the method for automatic generation of the Personal Web Usage Ontology of periodic access patterns from web usage logs that have been semantically developed with information on emotional influence. From this the consumer web access behavior and emotional influence web resources are captured. This method finds the consumer emotions in every session interval. If the session interval changes the accuracy level also changes.

Thus, in order to find optimum session interval, Optimum Session Interval based on Particle Swarm Optimization is included. By inclusion of PSO, the best session interval was chosen. Lattices were constructed. Associative access patterns were extracted and formed as rules. Classification was done by using the rules with respect to periodic attribute. Associative classification rules with highest support and confidence value was used to predict future requests of the user. The performance of the proposed algorithms in each phase of the research work was analyzed extensively and the results obtained are presented in the next chapter, **Results and Discussion**.