Web Usage Mining using Statistical Classifiers and Fuzzy Artificial Neural Networks

Prakash S Raghavendra, Shreya Roy Chowdhury, Srilekha Vedula
Kameswari *Department of Information Technology*
National Institute of Technology Karnataka

Abstract

There are many models in literature and practice that analyse user behaviour based on user navigation data and use clustering algorithms to characterize their access patterns. The navigation patterns identified are expected to capture the user's interests. In this paper, we model user behaviour as a vector of the time the user spends at each URL, and further classify a given new user access pattern. The clustering and classification methods of k-means with non-Euclidean similarity measure, Bayesian classifiers and artificial neural networks, with standardised fuzzy inputs are implemented and compared. Apart from identifying user behaviour, the model can also be used as a prediction system where we can identify deviational behaviour.

Keywords:
K-means, MLP, clustering, classification, web usage mining

1. Introduction

Web usage mining is the type of web mining activity that involves the discovery of user access patterns from one or more web servers. As more organizations begin to rely on the Internet and the World Wide Web to carry out business, the traditional strategies and techniques using databases for market analysis need to be revisited in this context. Organizations often generate and collect large volumes of data in their daily operations. Most of this information is usually generated automatically by Web servers and collected in server access logs. Other sources of user information include *referrer logs* which contain information about the referring pages for each page reference, user registration and survey data gathered via tools such as CGI scripts. Analysis of server access logs and user registration data can also provide valuable information on how to better structure a Web site in order to create a more effective presence for the organization. . In organizations using intranet technologies, such analysis can shed light on more effective management of workgroup communication and organizational infrastructure. For organizations that sell advertising on the World Wide Web, analyzing user access patterns helps in targeting advertisements to specific groups of users. Web mining also enables Web based businesses to provide the best access routes to services or other advertisements. When a company advertises for services provided by other companies, the usage mining data allows for the most effective access paths to these portals. Usage mining is also valuable to e-businesses whose business is based solely on the traffic provided through search engines. The use of this type of web mining helps to gather important information from customers visiting the site. This enables an in-depth log to complete analysis of a company's productivity flow. E-businesses depend on this information to direct the company to the most effective Web server for promotion of their product or service. Therefore, usage mining has definite valuable utility to the marketing of businesses and a direct impact on the success of promotional strategies.

In web usage mining, pattern discovery is difficult because only bits of information like IP addresses and site clicks are available. But analysis of this usage data will yield the information needed for organisations to provide an effective presence to their customers. The most effective way to retrieve useful information from a database is application-dependent. In this paper, the parameters that were determined to be relevant to web usage mining are presented. After a thorough literature survey in Section 2, the problem model is discussed in Section 3. The Section 4 discusses the statistical classifiers and Section 5 discusses ANN based classification. The experiments conducted and results obtained are indicated in Section 6 of this paper. Finally we conclude in Section 7.

2. Related Work

A survey of unsupervised and semi-supervised clustering methods was presented by Grira, Crucianu and Boujemaa in [1]. Squared error algorithms rely on the possibility of representing each cluster by a prototype. In general, the prototypes are the cluster centroids, as in the K-means algorithm. Fuzzy versions of methods based on the squared error are also defined, such as the Fuzzy C-Means. When compared to their 'crisp' counterparts, fuzzy methods are more successful in avoiding local minima of the cost function and can model situations where clusters actually overlap. In Morzy et al [2], a bottom-up approach of clustering based on Web Access Sequences is given, where frequent sequence patterns among web user sessions are identified. The users are then clustered based on their access sequence similarity. Shi [3] has used the approach of fuzzy modelling taking into account the time duration that a user spends at a URL. Nasraoui et al [4] have used the Competitive Agglomeration algorithm for Relational Data which yielded optimal number of clusters with non-Euclidean measures. In [5], it is argued that web user session identification itself is a non-trivial issue and clustering techniques have been used to characterise a user session. [6] gives a basis of evaluating web usage mining approaches and for predicting the user's next request. A survey of classification in data mining is given in [7]. A sequence based clustering for web usage mining using K-means algorithm with artificial neural networks and Markov models is given in [8]. It also demonstrates how a fuzzy approach yields superior accuracy. Artificial neural networks have been proven to be effective in dealing with classification problems and other machine learning areas. [9] contains a brief tutorial of ANNs referred to in Section 6 and 7. Multilayered Perceptrons (MLP) were found to be appropriate for the dataset used. The applicability of MLPs is discussed in [10]. [14] talks about Naive Bayes classifier which assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature.

Access sequences as a criterion is not primary because these can be misleading in cases where the user does not know the ideal route to his destination. Also, considering sequences by themselves as a parameter has the risk of incorporating the undesirable step of giving equal importance to all sites, irrespective of the amount of time spent there, due to which the focus of the analysis is lost. In this paper, the time spent by a user at a URL is the criterion for analysing his degree of interest. The Naive Bayes Classifier is applied, following which, the K-means classification algorithm (statistical) is then compared with the Multilayer Perceptron (artificial neural networks) method using logged web usage data to analyse accuracy in classification.

3. Modelling User Sessions as Vectors

From the initial web usage log, pre-processing yields the required fields of <ip, date_time, URL>. This refined log is used to identify user sessions. A *session* is defined as the sequence of URLs visited by the user. The time spent at a URL is determined by the difference in the timestamps of that URL visit and the next. Some of these values were found to be too 'large' to depict the duration of a URL visit. This large duration can be explained by scenarios like idling of the user or ending of a session. The term 'large' can be defined by a maximum limiting variable that can be specified. The time spent at the last URL in a session is required to be estimated as there is no URL succeeding it. For this paper, it was approximated as the average of the time spent by that user in the previous visited URLs in the session. Since there is no way of estimating the time spent by a user at a URL if it is the only visited URL in the session, singleton sessions were eliminated from the purview. After identifying the number of unique URLs, each user session is modelled as a vector $V^{(A)}$, where $v_i^{(A)}$ denotes the time spent (in seconds) at url_i by the a^{th} user, where $0<i<N$ and N is the total number of unique URLs, the dimensionality of the vector.

4. Statistical Clustering

After all the identified user sessions are modelled as vectors, the entire data is separated into 2 sets – training and test. The K-means clustering algorithm is applied on the training data. Since the data being clustered is not in the form of data points but vectors, standard k-means algorithm is modified to suit the requirements of this paper as described below. [4] puts forth a vector similarity measure which has been modified in this paper. The cosine of the vectors is taken as the similarity/distance measure instead of some Euclidian distance. The similarity between 2 sessions A and B can be given as follows:

$$similarity = \frac{\sum_{i=1}^{N} v_i^{(A)} v_i^{(B)}}{\sqrt{\sum_{i=1}^{N} v_i^{(A)2}} \sqrt{\sum_{i=1}^{N} v_i^{(B)2}}}$$

where N is the length of the session vector, $v_i^{(A)}$ is the time spent at the i^{th} URL in user session A and $v_i^{(B_j)}$ is the time time spent at the i^{th} URL in user session B. The cluster head or centroid of each cluster, for every iteration is a vector computed in the following manner:

$$V[i] = \frac{\sum_{m=1}^{m=n} V_m[i]}{n}$$

where i lies in the range [1,N], N is the number of URLs, n is number of members in the cluster currently and V_m is a vector of the member belonging to the cluster under consideration. A new vector is added to a cluster based on best proximity to its centroid using the similarity measure described above.

4.1 Bayesian Classification

A Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model". In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. For experimentation, we consider time spent at each URL as a parameter. This yields 145 parameters per vector in our case. The vectors being dealt with are normalized. We then train the classifier by feeding it each cluster, followed by all the training sample vectors that belong to that cluster. For each cluster, we calculate the mean (of time spent) and variance (of time spent) associated with each of the 145 URLs. In case variance is found to be zero, we ignore that parameter, i.e, we ignore the contribution of the time spent at that particular URL to the cluster's property. When a test session vector is given to the classifier, the classifier calculates the posterior of that test vector with respect to each of the classes. The class which yields the highest posterior is the class to which the test vector gets classified. The results of application of this classifier have been tabulated in Table 1. As is seen in the table, the results of classification are not good enough and hence have not been used in comparison with the following methods. We believe the reason for these results is due to the fact that there were many patterns for which variance was zero and hence could not to correct classification.

4.2 K-means Classification

The test data is classified according to best fit into a cluster and is assigned to that cluster. Belongingness is determined by best similarity between each cluster head vector and the test case vector.

Table 1. Results for Bayesian Classifier

Number of classes	Train %	Test%	No. of test cases	Percentage of test cases correctly classified
10	40	60	255	20.39%
10	60	40	170	26.47%
10	70	30	128	33.59%
10	80	20	85	35.29%
20	40	60	255	11.75%
20	60	40	170	12.94%
20	70	30	128	12.5%
20	80	20	85	15.29%

4.3. Accuracy of Classification

In order to determine the accuracy of classification we consider the clusters formed from K-Means clustering over the entire data set as a standard. Then, the percentage match between the neighbours of the test vector in the new cluster and the neighbours of the test vector in the standard clusters determines the accuracy of the classification. An example is illustrated in Figure 1. To check if a test case has been classified correctly, the number of common neighbours between the cluster it got assigned to and the cluster it belongs to in the standard is obtained. The ratio of the above number to the number of total neighbours in the standard cluster, multiplied by hundred gives us a metric which is compared against a threshold. If this percentage is above the specified threshold, the test case is considered as correctly classified. This paper uses a threshold of 60%, i.e. if 60% or more of a test vector's neighbours are the same as in the standard clusters, it has been classified correctly.

5. Artificial Neural Networks

5.1 Motivation

The utility of artificial neural network models lies in the fact that they can be used to infer a function from observations. This is particularly useful in applications where the complexity of the data or task makes the design of such a function by

For Example Say,

K=2
Total number of user sessions=10

Say, K-means clustering applied on all 10 sessions (Standard) gives:

Cluster 1 : 2, 3, 6, 8, 9
Cluster 2: 1, 4, 5, 7,10

Train Percentage = 50%
K- means on 50%of 10 (training data)
Cluster 1 : 1, 2, 3,
Cluster 2: 4, 5

Say, testing for 6, 7, 8, 9, 10 gives :

Cluster 1 : 1, 2, 3,6, 8
Cluster 2: 4, 5, 7, 9, 10

Figure 1. Example for Section 4.3

hand impractical, such as in web usage based classification. We choose the Multilayer Perceptron because it is an ANN model that maps sets of input data onto a set of appropriate output. It uses three or more layers of neurons (one or more hidden layers) with nonlinear activation functions, and can distinguish data that is not linearly separable. The activation function used in this paper is the *tanh* function.

5.2 Employing artificial neural networks

In supervised learning, we are given a set of example pairs $(x, y), x \in X, y \in Y$ and the aim is to find a function $f : X \rightarrow Y$. In other words, we wish to *infer* the mapping implied by the data; the cost function (error) is related to the mismatch between our mapping and the data. The mean-squared error cost function is used which tries to minimize the average squared error between the network's output, f(x), and the target value y over all the example pairs. When one tries to minimize this cost using gradient descent for the class of neural networks called Multi-Layer Perceptrons, one obtains the common and well-known back propagation algorithm for training neural networks.

5.3 The Multilayer Perceptron

This class of networks consists of multiple layers of computational units, usually interconnected in a feed-forward way. Each neuron in one layer has directed connections to the neurons of the subsequent layer. Connection weights are modified based on propagation of error till the network converges to a minimum error state. The multilayer perceptron consists of three or more layers (an input and an output layer with one or more *hidden layers*) of nonlinearly-activating nodes. Each node in one layer connects with a certain weight to every node in the following layer, influencing the importance of the node in the final output. Learning occurs in the perceptron by changing connection weights after each piece of data is processed, based on the amount of error in the output compared to the expected result. This is an example of supervised learning, and is carried out through back propagation. The learning rate and momentum parameters are chosen based on trial and error for minimum cost. The Learning rate generally lies in the range 0.2 to 0.8. The learning rate, when varied dynamically until a threshold, has proven to yield a better trained network and an improved convergence rate [11]. In this work, initially the weight adjustments are drastic and the network changes to accommodate incoming training cases. Towards the end of the training phase, the weight adjustments become lesser to retain the properties of the network after the training thus far. The MLP neural network was trained with training data from the standard K-means clusters. This was done by specifying the number of inputs to the MLP as the length of the user session vector, which is the number of unique URLs, and the number of outputs as the number of clusters into which classification can be done. ‘K’ is the number of output neurons. After the neural network is trained, a test input is classified into a cluster by checking the output neuron which gave maximum value of predicted output. A test case is considered to be correctly classified if the output neuron which gives the highest value of predicted output, maps to the same class as the original set of clusters (where the test case actually belongs).

Table 2. Tabulated Results of Experiments using Different Classification Methods

Training:Test Ratio	Total Test Cases	No. of correctly classified cases	Percentage Accuracy	Percentage increase from k-means
Vector Based K-Means: With K=10 and minimum percentage match = 60%				
40:60	255	89	34.90	
60:40	170	110	64.71	
70:30	128	83	64.84	
80:20	85	57	67.06	
Vector Based K-Means: With K=20 and minimum percentage match = 60%				
40:60	255	175	68.63	
60:40	170	121	71.18	
70:30	128	104	81.25	
80:20	85	72	84.71	
Multilayer Perceptron with Normal Input: K=10 and Learning Rate = 0.5				
40:60	255	164	64.31	29.41
60:40	170	124	72.94	8.23
70:30	128	97	75.78	10.94
80:20	85	67	78.82	11.76
Multilayer Perceptron with Normal Input: K=10 and Learning Rate = 0.6				
40:60	255	174	68.24	33.34
60:40	170	123	72.35	7.64
70:30	128	94	73.43	8.59
80:20	85	63	74.12	7.06
Multilayer Perceptron with Normal Input: K=20 and Learning Rate= 0.5				
40:60	255	194	76.08	7.45
60:40	170	130	76.47	5.29
70:30	128	106	82.81	0.56
80:20	85	67	78.82	-5.89
Multilayer Perceptron with Normal Input: K=20 and Learning Rate= 0.6				
40:60	255	183	71.76	3.13
60:40	170	131	77.06	5.88
70:30	128	94	73.44	-7.81
80:20	85	70	82.35	-2.36
Multilayer Perceptron with Standardised Input: With K=10 and Learning Rate = 0.5				
40:60	255	183	71.76	36.86
60:40	170	131	77.08	12.37
70:30	128	101	78.90	14.06
80:20	85	62	72.94	5.88
Multilayer Perceptron with Standardised Input: With K=10 and Learning Rate = 0.6				
40:60	255	178	69.80	34.90
60:40	170	128	75.29	10.58
70:30	128	95	74.21	9.37
80:20	85	63	74.12	7.06
Multilayer Perceptron with Standardised Input: With K=20 and Learning Rate = 0.5				
40:60	255	180	70.59	1.96
60:40	170	122	71.76	0.58
70:30	128	87	67.97	-13.28
80:20	85	57	67.06	-17.65
Multilayer Perceptron with Standardised Input: With K=20 and Learning Rate = 0.6				
40:60	255	183	71.76	3.13
60:40	170	123	73.35	2.17
70:30	128	93	72.65	-8.6
80:20	85	65	76.47	-8.24

5.4 Normalized input

The input to the neural network has to be normalized to values in the range 0 to 1. This gives the normalized time spent at each URL. This was used as input for the initial experiments with the MLP.

5.5 Standardised (Fuzzy) input

The range 0.0 to 1.0 was divided into smaller intervals of width 0.2. Each interval denotes in increasing order the interest of the user in a particular URL.
[0.0, 0.2) → very low
[0.2, 0.4) → low-medium
[0.4, 0.6) → medium
[0.6, 0.8) → medium-high
[0.8, 1.0] → high

The weighted average of times spent by users which belong to a particular range is used as the representative of the range, except the first and last ranges where representative values are taken as 0.0 and 1.0 are used as representative values. These *fuzzified* values were used as input to the MLP in later experiments for comparison.

6. Experiments and Results

For experiments, data from logs of www.engineer.nitk.ac.in (Technical fest of NITK) was used, which spanned over 2 days, and had site clicks from all over the world. As mentioned in Section 3, 3600 seconds (1 hour) was fixed as the maximum time that could be spent by a user on a page. The pre-processing yielded 425 user sessions, after removing singleton sessions, and 145 unique URLs were obtained. Each session was modelled as a vector of dimension 145, and hence 425 such vectors representing 425 user sessions were present. First, the modified K-means clustering was performed to generate 10 and 20 clusters. The Bayesian classifiers were poor in classification.
classification and the MLP scenario. For MLP, the source code available in [12] was adapted to the present requirement and used. Using MLP, training and testing over 850 epochs with two different learning rates were performed. The number of epochs was set to 850 as it was observed that the network converged to minimum error at 850, after which it began to oscillate. Number of hidden neurons used was 30, since experimentation with other values yielded poorer results. This can be attributed to the fact that the variability in the sessions are higher and statistical classifiers are not able to capture and recognize the patterns. See the Table 2. for results. These were used as standard for calculating the accuracy of K-means The Momentum parameter was set to 0.03 for all cases. Results of all the experiments are tabulated in Table 2. showing the number of clusters being dealt with, ratio of training data to test data, total number of test cases, number of correctly classified cases, the percentage of correctly classified test cases and for MLP the learning rate and the difference in accuracy from the K-means method. The Figures 2 and 3 summarize the table graphically. Each graph has the methods employed on the horizontal axis and the percentage accuracy of classification on the vertical. The four curves indicate different train to test ratios.

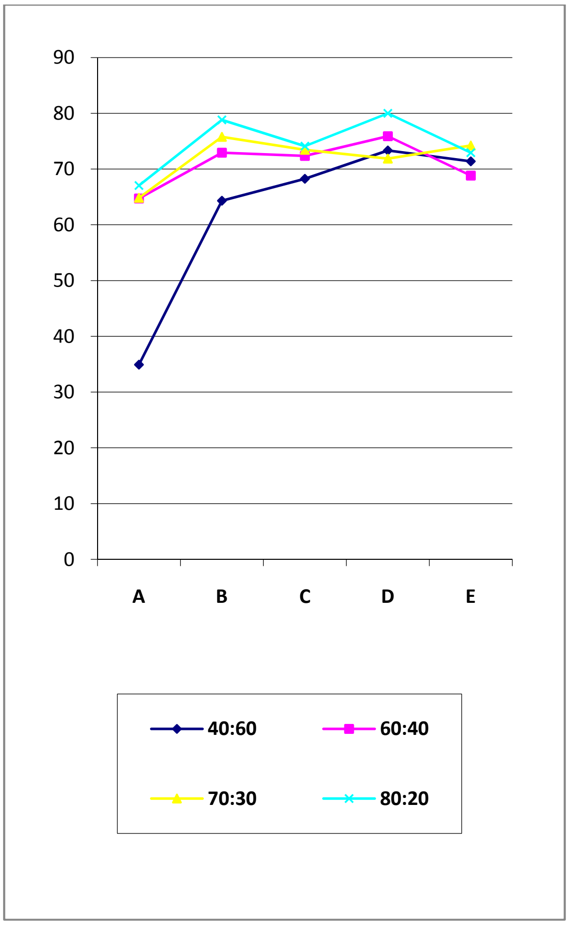

Figure 2. Comparison showing different classification results for k=10

In Figure 2, the graph shows a comparison of results for classification into 10 clusters (k=10) for different train to test ratios. As is evident, neural networks are far more accurate than statistical methods such as k-means. Observing the corresponding accuracy values for different train to test ratios when the number of clusters is increased to 20 as in Figure 3, MLP shows poorer accuracy. This can be due to insufficient training examples for each cluster and hence poor learning. This phenomenon is known as *underfitting*, where the network yields high prediction/classification error due to the model being too simple or untrained. However, the accuracy of k-means classification increases when the number of clusters is increased as it is a statistical approach, that is, the ability to arrive at the label for a cluster most closely representing its members is greater when the number of clusters is increased. Also in some cases, higher train:test ratio yields poor classification accuracy in MLPs. This can be due to the phenomenon called *overfitting*, where the network is too specific to train data**.**

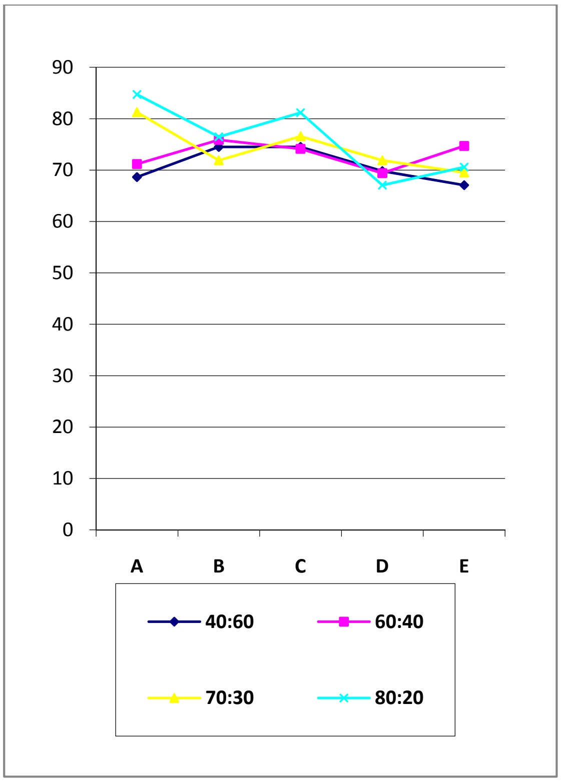

Figure 3. Comparison showing different classification results for k=20

Legend for Fig 2 and 3:
X-Axis
A – K-means
B – MLP with Normal Input with Dynamic Learning Rate decreasing from 0.5
C – MLP with Normal Input with Dynamic Learning Rate decreasing from 0.6
D – MLP with Weighted Standardised Input with Dynamic Learning Rate decreasing from 0.5
E – MLP with Weighted Standardised Input with Dynamic Learning Rate decreasing from 0.6
Y-Axis: Percentage Accuracy

7. Conclusion

Methods for web usage based classification were surveyed, and artificial neural networks, in particular, Multilayered Perceptrons, were found to be to be especially effective and relevant for the problem of classification. Introducing the MLP algorithm to standardised (fuzzified) input resulted in a marked improvement in classification accuracy over the modified k-means algorithm. With this achieved, prediction of user's belongingness to a cluster is also realised. Neural networks have thus been observed to be far more intuitive for a supervised learning approach than the statistical method of K-means. The ratio of train:test data most resembling real applications is 40:60. With optimum number of classes, we see that MLPs perform drastically better than K-means method for this scenario. A neuro-fuzzy network, that is, a fuzzy inference system (FIS) in the body of an artificial neural network may be used. Embedding an *FIS* in a general structure of an *ANN* has the benefit of using available *ANN* training methods to find the parameters of a fuzzy system. This is closer to the neurobiological processes that take place in our brain and may hence yield more accuracy.

8. References

[1] N. Grira, M. Crucianu, and N. Boujemaa, "Unsupervised and semi-supervised clustering: a brief survey." In A Review of Machine Learning Techniques for Processing Multimedia Content. Report of the MUSCLE European Network of Excellence, July 2004.

[2] T. Morzy, M. Wojciechowski, and M. Zakrzewicz. "Web users clustering." International Symposium on Computer and Information Sciences, 2000.

[3] Shi, P., "An Efficient Approach for Clustering Web Access Patterns from Web Logs", *International Journal ofAdvanced Science and Technology*, Vol. 5, April 2009.

[4] Nasraoui O., Frigui H., Joshi A., and Krishnapuram R., "Mining Web Access Logs Using Relational Competitive Fuzzy Clustering", Proceedings of the Eighth International Fuzzy Systems Association Congress, Hsinchu, Taiwan, August 1999.

[5] A. Bianco, G. Mardente, M. Mellia, M. Munafò, and L. Muscariello, "Web user session characterization via clustering techniques," in Proc. *IEEE GLOBECOM 2005*, St. Louis, MO, Nov. 2005, vol. 2, pp. 1102-1107.

[6] Gery M. and H. Haddad, “Evaluation of Web Usage Mining Approaches for User’s Next Request Prediction”,Proceedings of the fifth ACM international workshop on Web information and data management, 2003, pp. 74–81.

[7] T. Phyu, “Survey of Classification Techniques in Data Mining”, Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I, IMECS 2009, Hong Kong, March 18 - 20, 2009.

[8] Park, N. Suresh, Jeong, “Sequence-based clustering for Web usage mining: A new experimental framework and ANN-enhanced K-means algorithm”, Data & Knowledge Engineering 65 (2008) 512–543.

[9]http://richardbowles.tripod.com/neural/neural.htm (14, April 2010).

[10] S.K. Pal, S. Mitra, “Multilayer Perceptron, Fuzzy Sets and Classification”, IEEE Transactions on Neural Networks, Vol 3, No. 5, September 1992.

[11] R. Salomon, “Improved convergence rate of back-propagation with dynamic adaption of the learning rate ”, *Parallel Problem Solving from Nature*, Volume 496/1991, Springer Berlin, 1991

[12]‘neural network software’ http://www.philbrierley.com. (Access date: 3 March 2010).

[13]‘Web Data Mining.net’, http://www.web-datamining.net/usage/. (Access date: 5 May 2010).

[14]‘Naive Bayes classifier’ http://en.wikipedia.org/wiki/Naive_Bayes_classifier, (Access date: 10 Jan 2011).