APPLYING AND COMPARING HIDDEN MARKOV MODEL AND FUZZY CLUSTERING ALGORITHMS TO WEB USAGE DATA FOR RECOMMENDER SYSTEMS

Shaghayegh Sahebi
s.sahebi@ece.ut.ac.ir

Farhad Oroumchian
oroumchian@acm.org

Ramtin Khosravi
rkhosravi@ece.ut.ac.ir

ABSTRACT

As the information on the Web grows, the need of recommender systems to ease user navigations becomes evident. There exist many approaches of learning for Web usage based recommender systems. In this study, we apply and compare some of the methods of usage pattern discovery, like simple k-means clustering algorithm, fuzzy relational subtractive clustering algorithm, fuzzy mean field annealing clustering and hidden Markov model, for recommender systems. We use metrics like prediction strength, hit ratio, precision, prediction ability and F-Score to compare the applied methods on the usage data of the CTI Web site of DePaul University. Fuzzy mean field annealing clustering and hidden Markov model acted better than other methods due to fuzzy nation of human behavior in navigation and extra information utilized in sequence analysis.

KEYWORDS

Web mining, usage pattern discovery, recommendation system, fuzzy clustering, hidden Markov model.

1. INTRODUCTION

With the rapid growth of World Wide Web, Web mining has been the subject of many researches. Web mining is the use of data mining techniques to automatically discover and extract information from Web documents and services (Etzioni, 1996). Web mining problem is mainly divided into three parts which are Web content mining, Web structure mining and Web usage mining (Cooley et al, 1999). Web usage mining is the automatic discovery of user access patterns from Web servers and tries to discover valuable information from users' transactional data (Srivastava et al, 2000). As a result, prediction of users requested pages is an important issue for personalization applications.

A Web personalization system is defined as any system that tailors the Web experience for a particular user/a group of users (Mobasher et al, 2000). Many web mining techniques have been used in web personalization systems to discover usage patterns from Web data such as clustering techniques, association rule mining, and click pattern analysis. For example, Shahabi et al (1997) and Nasraoui et al (1999) have proposed clustering of user sessions to predict future user behavior. Cadez et al. (2003) used the Expectation-Maximization (EM) algorithm on a mixture of Markov models for clustering user sessions. Joshi and Krishnapuram (2000) have used a fuzzy clustering approach for clustering user sessions.

Since there exist uncertainties in the nature of human behavior, fuzzy models are better to handle such uncertainties in data comparing to conventional models (Abraham, 2003). In this paper, we apply and compare fuzzy clustering algorithms, hidden Markov model and simple k-means clustering algorithm to predict requested pages of users. We use Web usage data from the CTI Web pages of the DePaul University (http://cti.depaul.edu) in Section 3. The fuzzy clustering algorithms, which have been applied, are relational

fuzzy subtractive clustering (RFSC) algorithm (Suryavanshi et al, 2005) and fuzzy mean field annealing (MFA) clustering algorithm (Song et al, 2007). The applied methods are compared with a simple k-means clustering algorithm and hidden Markov model (HMM) (Bishop, 2006) which is an advanced sequence analysis algorithm in experimental results section. The algorithms are described in Sections 2.1 and 2.2.

2. METHODS

2.1 Clustering Algorithms

The goal of the simple clustering algorithms is to partition the state space into C clusters. In fuzzy clustering, the idea is to specify the degree of membership of a data point (between 0 and 1) to each cluster. If we consider the partitioning condition of the fuzzy clustering, the assigned weights for each data point should add up to 1. If we consider the set of N data points as $X = \{x_j \mid j = 1...N\}$ and the set of C clusters as $C = \{c_i \mid i = 1...C\}$, we can record the membership weight of the data point in a $C \times N$ matrix called U. In this case, the fuzzy partition condition can be regarded as:

$$\sum_{i=1}^{C} u_{ij} = 1 \quad \text{for all } j = 1...N \qquad (1)$$

2.1.1 Simple k-means

The simple k-means algorithm is a mass type clustering algorithm which starts with an initial cluster assignment to the data points and tries to minimize an objective function by moving the data points in different clusters. The objective function J is defined by Equation 2 in which i is the i^{th} cluster, x_j is the j^{th} data point and μ is the cluster centers matrix defined in Equation 3.

$$J = \sum_{i=1}^{C} \sum_{x^j \in \omega_i} \left\| x^j - \mu_i \right\|^2 \qquad (2)$$

$$\mu_i = \frac{1}{N_i} \sum_{x^j \in \omega_i} x^j \qquad (3)$$

We can obtain the best cluster number by repeating this algorithm when increasing the cluster number if there is no self evidence about the number of clusters. The best cluster number then is chosen when reduction rate of objective function were less than increase rate of cluster numbers.

Generally, the Euclidean distance is used for simple k-means algorithm. The results of using this distance are not satisfying for Web usage data. As a result, we have used the cosine distance instead of Euclidean distance.

2.1.2 Relational Fuzzy Subtractive Clustering (RFSC)

This algorithm works based on the distances between all data points which is stored in a distance matrix R. In RFSC algorithm (Suryavanshi et al, 2005), we consider every data point x_i as a potential cluster center with the following potential:

$$P_i = \sum_{j=1}^{N} e^{-\alpha \left\| x_i - x_j \right\|^2} \quad \text{where } \alpha = 4 / r_a^2 \qquad (4)$$

In Equation 4, r_a is the (Euclidean) radius defining a neighborhood. Data points outside this radius have less influence on the potential. In this algorithm, the goal is to reduce the number of the parameters which user specifies so we try to determine α from the given dissimilarity matrix R. For Web log data, a heuristic method has been defined in (Suryavanshi et al, 2005). If we consider dissimilarity (γ_i) of each object i (session) from every other object as the median of dissimilarity values of object i to all other objects, then we can define the dissimilarity (γ) for the dataset as the median of all γ_is, for $i \in [1...N]$. It can be seen that γ will

always be a value in the range [0, 1]. We choose $\alpha = 4/\gamma^2$. The RFSC algorithm is shown in Listing 1. It uses the following equation for updating the potentials:

$$P_i = P_i - P_k^* e^{-\alpha R_{ic_k}^2} \quad \text{for each } i = 1 \ldots N \qquad (5)$$

The accept ratio(ϵ) in the algorithm is defined to achieve a higher limit for the potential value so that if a potential was more than accept ratio, it is definitely chosen as a cluster center. On the other hand, if the potential of a data point is less than reject ratio (ϵ'), it will never be chosen as a cluster center. After finding C clusters, we can calculate the membership degree of x_j to each cluster c_i as follows:

$$u_{ij} = e^{-\alpha R_{cij}^2} \quad \text{for } i = 1 \ldots C \text{ and } j = 1 \ldots N \qquad (6)$$

Listing 1. Relational Fuzzy Subtractive Clustering (Suryavanshi et al, 2005)

1. Set ϵ (accept ratio) and ϵ' (reject ratio)
2. Calculate the potential of each session as equation (4). Select x_i with the maximum potential as the first cluster center. Set cluster counter $k = 1$, and let $c_k = c_1 = i$. Here x_{ck} is the cluster center. Let $P_k^* = P_1^* = P_i$
3. Revise the potential of each session using equation (5) (subtractive step). Now, select x_t with the current maximum potential P_t as the candidate for the next cluster center
   - (Accept) If $P_t > P_1^*$ then accept x_t as the new cluster center, increment the cluster counter, $k = k+1$, set $c_k = t$, and go to step 3
   - (Reject) else if $P_t < P_1^*$ then reject x_t and terminate.
     Else Let d_{min} = minimum of the dissimilarity values between x_t and all the previously found cluster centers
     i. (Accept) If $(d_{min}/\gamma) + (P_t/P_1^*) \geq 1$, then accept x_t as the new cluster center, $k = k+1$, $c_k = t$, go to step 3
     ii. (Reject) else reject x_t as a cluster center and set its potential P_t to 0. Select the data point with the next highest potential as the new candidate cluster center and go to step 3
     iii. endif
   - endif

For evaluation of this clustering (Suryavanshi et al, 2005), division of compactness of clusters to their separation, is used. To determine proper values for accept and reject ratios, we should repeat the algorithm with different values for them and choose the ratios in which the measure is smallest. The goodness measure of this algorithm is calculated like below:

$$\text{Compactness} = \frac{1}{C} \sum_{i=1}^{C} \frac{\sum_{j=1}^{N} u_{ij}^2 R_{c_{ij}}^2}{\sum_{j=1}^{N} u_{ij}^2} \qquad (7)$$

$$\text{Separation} = \min_{i \neq k} \; R_{c_i c_k}^2 \quad \text{for } i = 1 \ldots C \text{ and } k = 1 \ldots C \qquad (8)$$

$$\text{Index Of Goodness} = \frac{\text{Compactness}}{\text{Separation}} \qquad (9)$$

2.1.3 Fuzzy Mean Field Annealing Clustering

This algorithm (Song et al, 2007), unlike common fuzzy clustering algorithms which find the cluster centers at first step and then recalculate the fuzzy membership degrees, determines whether new membership degrees are acceptable or not by changing the membership degrees regularly and calculating the change of energy.

In (Song et al, 2007), the membership matrix numbers are updated in parallel to reach a global optimum. The simulated annealing (SA) algorithm guarantees to reach the global optimum in a time-consuming approach. This algorithm leverages a probabilistic approach to perturb the current state to reach the next state while the Mean Field Annealing (MFA) algorithm uses a deterministic perturbation method. The MFA algorithm combines many of SA features with neural networks to preserve the rapid convergence of neural networks while keeping the quality of results of SA. In extended MFA, the algorithm is also applicable on continuous values. In fuzzy clustering we should use extended (fuzzy) MFA. In fuzzy MFA clustering, we can use the below fuzzy clustering measure. The cluster center matrix μ is calculated by equation (10).

$$\mu_i = \frac{\sum_{j=1}^{N} u_{ij}^m \times x^j}{\sum_{j=1}^{N} u_{ij}^m} \quad (10)$$

$$S(C,m,U) = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^m \left(x^i - \mu_j\right)^2 \quad (11)$$

In Listing 2 you can see the fuzzy MFA clustering algorithm. The updates in algorithm are done based on following equations:

$$\mu_{jk}^{t+1} = \frac{\sum_{j=1}^{N} \left(u_{ij}^{t+1}\right)_{new}^m \cdot x_{ik}}{\sum_{j=1}^{N} \left(u_{ij}^{t+1}\right)_{new}^m} \quad (12)$$

$$S(C,m,U_{new})^{t+1} = \sum_{i=1}^{N}\sum_{j=1}^{C} \left(u_{ij}^m\right)_{new}^{t+1} \left(x^i - \mu_j^{t+1}\right)^2 \quad (13)$$

$$\Delta S = S(C,m,U_{new})^{t+1} - S(C,m,U_{old})^{t+1} \quad (14)$$

Listing 2. Fuzzy Mean Field Annealing Clustering (Song et al, 2007)

1. Initialize T, T_f, m and ε
2. While ($T > T_f$) { (do each of $u_{i,k}$ in parallel)
   a. Pick $u_{i,k} \in U^{t+1} | U \in R^{C \times N}$
      $U = \{ U_{ik} \in [0,1] \text{ for each } i,k \}$
      $\sum_{i=1}^{C} u_{ik} = 1$ for each i
   b. (Perturb to get a new $u_{i,k}$)
      $< u_{i,k} >_{new} =< u_{i,k} > + \varepsilon \quad (-0.2 < \varepsilon < 0.2)$
   c. Calculate μ_{jk}^{t+1}, $S(C,m,U_{new})^{t+1}$ and ΔS using equations (12) to (14)
   d. (Check acceptance)
      $< u_{ik} >=< u_{ik} >_{new}$
      if ($\Delta S < 0$) (then accept)
         $< u_{ik} >=< u_{ik} >_{new}$
      else if ($exp(-\Delta S / T) > random(0..1)$)
         $< u_{ik} >=< u_{ik} >_{new}$
      else (reject)
         $< u_{ik} >=< u_{ik} >$
      set $t = t+1$ and return to step (a)
   e. Repeat step(b) to step(d) sufficiently
   f. Calculate the average of accepted state from step(b) to step(e) and then set the average as $< u_{ik} >$ in T
      $T = \alpha * T$
} (end while)

2.2 Sequence Analysis Algorithms

2.2.1 HMM

Hidden Markov model (Bishop, 2006) is a statistical model which the system modeling in it, is considered as a Markov process and the challenge is to estimate the model parameters based on observable states. After estimating these parameters, model can be used in secondary analyses. This model is a simple model of Bayesian networks.

In the regular Markov state, all the states are observable and the parameters are only probabilities of state changes. In hidden Markov model, states are not observable but we can observe the parameters affected by

states. These types of models can be applied to temporal pattern recognition, bioinformatics and sound recognition. We can see a global architecture of Markov model when running, which is called Trellis, in Figure 1. Each oval shows a variable which can take some values; *x(t)* is the hidden state in time *t*, *y(t)* is the observation in time *t* and the arrows show the conditional dependence of variables. As we can see, state at time *t* just depends on the data at time *t -1* and the observed value depends on the state at the same time.

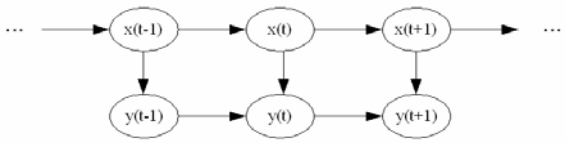

Figure 1. A global architecture of hidden Markov model while running

In this section, we have used MATLAB toolbox (http://www.mathworks.com) to build our model. We adopted Baum-Welch algorithm (Baum et al, 1970) which is a derivation of EM algorithm. This algorithm uses a predefined number of states so that the emission probabilities and state changes matrixes converge to their local optimum. We have used a heuristic based on the number of clusters which obtained in clustering algorithms for states number. Based on the obtained number of clusters, we have used 25, 30 and 35 states for our reduced dataset with 366 page groups. The best result was for 30 states.

2.3 Recommendation in Different Algorithms

To recommend items (pages) to the users in simple k-means algorithm, we first find the best cluster for each evaluation data point (session) by calculating the distance between these data points with cluster centers. After that, we sort the pages of the best cluster based on the sum of times of user views on those pages. The most important pages of each cluster are pages in which this sum is maximized. We recommend most important pages of the assigned cluster which user has not seen yet.

For recommendations in fuzzy clustering algorithms, we first calculate the distance between evaluation data points and cluster centers and then calculate fuzzy membership matrix (*U*) for the evaluation data based on the approach of calculating membership degrees in each algorithm. We will sort the clusters based on their degree of importance for each data point or the ascending order of membership degrees in each row. For each session, the number of pages recommended from each cluster is determined by the membership degrees of each session to each cluster. For each cluster, we recommend the most important pages of the cluster by calculating a weighted sum of the viewing time of users on each page and choosing the pages with higher weighted sum. This weighted sum is calculated by multiplication of membership matrix with the time of viewing of each page in each session.

In HMM, first we select sequences of the length greater than *L (L= 4, 5 and 6)*. Then the first *L-1* pages in sequence are padded with all possible choices of pages one by one and Viterbi algorithm (Forney, 1973) is used to derive most probable states for each sequence and the probability of such sequence. For recommendation, we select the top *k* most probable pages and show them to user. We consider a recommendation as a hit if the user will go to that page in the remaining requests of her/his session (i.e. *L* and pages after that).

3. EXPERIMENTAL RESULTS

In this study we have used the usage data of DePaul University. In this data set, sessions of 13745 users on 683 pages of CTI web site of DePaul University has formed a 13745×683 matrix. Each member of this matrix shows the time each user had spent on each page. We have used about 70 percent of the data for training and the remainder for test and evaluation. This data matrix is so sparse and makes it difficult to mine the data.

To solve the sparsity problem, we have used two alternatives. By analyzing the URLs, we have found that some of the pages are closely related to each other or can be in one category. As a result, in the first solution we have aggregated columns of the matrix using the page URLs. We refer to this data set as the *aggregated data*. Reducing the sparsity of the matrix by aggregation, it was still sparse.

The second alternative was to use principal component analysis (PCA) on the data matrix. We have used PCA on both main and aggregated data sets but the results were still unsatisfying. To preserve the 70 percent of accuracy with PCA, we obtained about 380 components while for keeping 95 percent of accuracy, the number of components reached to about 580 components. The obtained eigenvalues were so closed to each other. We finally used PCA on aggregated data and it resulted to 53 components with 71 percent of accuracy.

To enhance the performance of algorithms in recommender systems, we have removed about 20 percent of pages from each session randomly and assigned the incomplete remaining sessions to the achieved clusters from train data.

We have applied some of the measures suggested in (Bose et al, 2006) and additional measures, taken from information retrieval literature, to evaluate the goodness of recommendations. Theses measures are:

- Predictive Ability (PA): Percentage of pages in the test sessions for which the model was able to make recommendations. This is a measure of how useful the model is. It is like recall in Information Retrieval.
- Prediction Strength (PS): Average number of recommendations made for a page.
- Hit Ratio (HR): Percentage of hits with respect to number of the sessions. If a recommended page is actually requested later in the session, we declare a hit. The hit ratio is thus a measure of how good the model is in making recommendations.
- Precision (Pr): Percentage of hits with respect to the number of recommendations for each session.
- F-Score (FS): A proportion of precision and predictive ability which is taken from Information Retrieval:

$$\text{FScore} = \frac{(\text{PredictiveAbility} \times \text{Precision})}{(\text{PredictiveAbility} + \text{Precision})}$$

To be ideal, predictive ability, precision and so F-Score should be one. It is also better to have a higher hit ratio and prediction strength.

At first, we have repeated the simple k-means algorithm with different number of clusters on the main data to gain the proper number of clusters. The results are shown in Figure 2 in which we can see that the sum of intraset distance is reduced by increasing the number of clusters until about 20 clusters and then the reduction rate decreases. For the recommendations we have used the cosine similarity measure. The results, compared to other methods, are shown in Table 1.

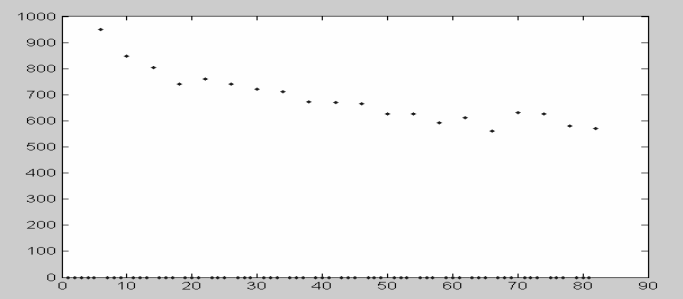

Figure 2. Sum of intraset distance of test data for all clusters considering different number of clusters

In relational fuzzy subtractive clustering, we need to keep a large distance matrix which its dimensions are equal to the number of training data points. The dimensions of matrixes in MATLAB are restricted and the training data point number was bigger than this dimension. As a consequence, we just could use 50 percent of our training data for this algorithm which caused weaker results with respect to other algorithms. To gain a proper result, we have increased accept and reject ratios from 0.1 to 0.9 with 0.1 step size. The best results were for accept ratio of 0.1 and reject ratio of 0.2 which created 8 clusters. We have used the proposed

method for fuzzy clustering in Section 2.1.2, using membership matrix calculations of RFSC, for recommendations. The results of this algorithm were worst than simple k-means algorithm although we expected the opposite. We can conclude that the results could be better by considering different values of accept and reject ratios, especially less values.

The fuzzy MFA algorithm is very time-consuming and we have used 10 clusters for this experience. Although this algorithm is based on fuzzy k-means clustering, it does not use a straight approach to find the distance of data points. We have used the distance and membership degree calculations of fuzzy k-means algorithm to calculate the evaluation data membership degrees. This algorithm has a better outcome with respect to other clustering algorithms which is reasonable because of the global optimization method.

Although the MFA algorithm takes a long time to converge, it is not as time-consuming as hidden Markov model. For the HMM, 70 percent of data, which is selected randomly, is used for training and the rest is left for testing. Due to high complexity of the training algorithm we bounded the algorithm to 50 iterations. After training, we used the sessions in test data for evaluation. In this way, for each sequence in test data we have a 366 probability. Figure 3 shows the correct recommendation of HMM for different L values for recommendations in length of k percent. According to the figure, the model performs better for $L=4$. Based on the clustering, 10 to 15 pages are recommended for each user. In this model, we have used 11 recommendations for each user.

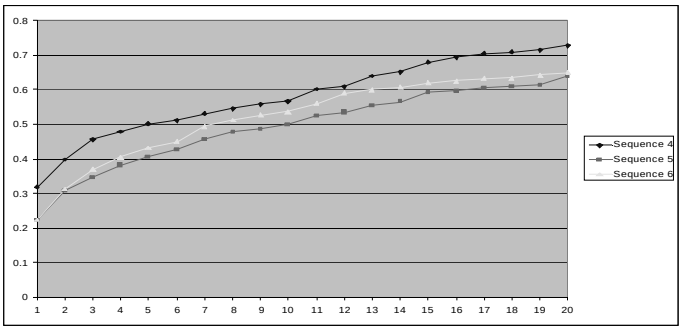

Figure 3. Ratio of correct recommendations to number of recommendations for sequences with different lengths in HMM

Table 1. Goodness measures of recommendations in different models

Model Name	PS	HR (%)	PA (%)	Pr (%)	F-Score (%)
k-means (20 Clusters)	12.34	-	42	6.8	11.7
k-means (10 Clusters)	12.48	-	44.2	7	12.1
RFSC (8 Clusters)	9.9	53	31	6	11
FMFA (10 Clusters)	9.98	65.5	43.15	7.8	14
HMM (20 Clusters)	11	60	38	11	18

According to the results of these algorithms in Table 1, we can see that HMM acts better than other algorithms. This result is reasonable because we utilize the order of the viewed pages in conjunction with pages which had been seen. In addition, we could gain better results by recommending more pages to the user. The most important disadvantage of this model is that it is very time-consuming.

The results of RFSC algorithm were worse than simple k-means results. This result seems to be illogical because the nature of recommender systems, which comes from human behavior in Web environment, is fuzzy and should be more compatible with fuzzy clustering algorithms. Using about half of the training data for this algorithm may have brought about this unreasonable result. Additionally, the number of clusters is less that other clustering algorithm which is probably a result of ignoring many different states for accept and reject ratios. Considering these states, especially less values, we could probably increase the quality of the results. The advantage of this algorithm is its high speed in convergence and nonnecessity to define the number of clusters. Moreover, this algorithm does not need the fuzzy partitioning condition and so it is less sensitive to noises. The disadvantage of this algorithm is consuming lots of memory for keeping the distance matrix between training data points.

Besides, we can see that the results of fuzzy MFA clustering algorithm were better than other clustering algorithms. This also is a reasonable outcome because this algorithm provides a global optimum solution for clustering problem. By increasing the recommendation number to user, we can observe an increase in this algorithm's preciseness. Furthermore, we could enhance the results by applying the fuzzy MFA algorithm on different number of clusters and choosing the best of them.

4. CONCLUSION AND FUTURE WORK

In this study, we have applied and compared different algorithms in Web usage-based recommender systems. These algorithms contained clustering algorithms, like simple k-means, relational fuzzy subtractive clustering and fuzzy MFA clustering, and sequence analysis algorithm with hidden Markov model. The fuzzy MFA algorithm had been applied for the first time on Web usage data.

As we have seen and described in the experimental results section, the HMM algorithm acted better than other algorithms in recommendations but it suffered from long learning time and after that Fuzzy MFA algorithm was better than other clustering algorithms.

For future works, we can improve these algorithms by:

- Using different values for accept and reject ratios in RFSC algorithm
- Utilizing the distance matrix for all training data points by distributing it into different matrixes in RFSC algorithm
- Using the cosine distance measure instead of Euclidean one in fuzzy MFA clustering algorithm
- Taking the advantage of viewing time of users in HMM

At last, it seems that the obtained results, especially the HMM results, are good enough for a usage-based Web recommender system, but we need more precise results. As a consequence, it is not enough to use only usage data for recommender systems and it is better to utilize the domain knowledge like the Web site structure or semantic information of Web site pages.

ACKNOWLEDGEMENT

We would like to thank Mr. Masud Moshtaghi for his helps in preparing this paper.

REFERENCES

Abraham, A., 2003. Business Intelligence from Web Usage Mining. *In Journal of Information and Knowledge Management (JIKM)*, Vol. 2, No. 4, pp 375-390.

Baum L. E., Petrie T., Soules G. and Weiss N., 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains, *Ann.Math. Statist,* Vol. 41, No. 1, pp. 164-171.

Bishop C. M., 2006. *Pattern Recognition and Machine Learning*. Springer Publishers, Singapore.

Bose, A. et al, 2006. Incorporating Concept Hierarchies into Usage Mining Based Recommendations. *Proceedings of WebKDD 2006*, Philadelphia, USA, pp 110-126.

Cadez, I. et al, 2003. Model-based clustering and visualization of navigation patterns on a web site. *In Data Mining and Knowledge Discovery*, Vol. 7, No. 4, pp 399–424.

Cooley, R. et al, 1999. Data preparation for mining World Wide Web browsing patterns. *In Journal of Knowledge and Information Systems (KAIS)*, Vol. 1, No. 1, pp 5-32.

Etzioni, O., 1996. The World Wide Web: Quagmire or gold mine? *In Communications of the ACM*, Vol. 39, No. 11, pp 65-68.

Forney G. D., 1973. The Viterbi algorithm, *Proceedings of the IEEE,* pp 268–278.

Joshi, A. and Krishnapuram, R., 2000. On mining web access logs. *Proceedings of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD).* Dallas, USA, pp. 63-69.

Mobasher, B. et al, 2002. Using sequential and non-sequential patterns for predictive web usage mining tasks. *Proceedings of the IEEE International Conference on Data Mining.* Maebashi City, Japan, pp. 669–672.

Nasraoui, O. et al, 1999. Mining Web access logs using relational competitive fuzzy clustering. *Proceedings of the Eight International Fuzzy Systems Association World Congress.* Hsinchu, Taiwan.

Shahabi, C. et al, 1997. Knowledge discovery from users Web-page navigation. *Proceedings of Workshop on Research Issues in Data Engineering.* Birmingham, UK, pp. 20-29.

Song, C. et al, 2007. A Mean Field Annealing Algorithm for Fuzzy Clustering. *Proceedings of Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD).* Haikou, China, pp. 193-197.

Srivastava, J. et al, 2000. Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *In SIGKDD Explorations*, Vol. 1, No. 2, pp 12-23.

Suryavanshi, B. S. et al, 2005. An Efficient Technique for Mining Usage Profiles using Relational Fuzzy Subtractive Clustering. *Proceedings of International Workshop on Challenges in Web Information Retrieval and Integration (WZH'05).* Tokyo, Japan, pp. 23-29.