International Conference on Science, Engineering and Spirituality, ICSES'10

Web Usage Mining for Tracking Evolving User Profiles

Mr.B.V.Wakode
M.Tech CSE
GHRCE,Nagpur
wakode.bhushan@gmail.com

Dr.R.V.Dharaskar
Professor
GHRCE,Nagpur

Abstract *- Data on the Web is noisy, huge, and dynamic. This poses enormous challenges to most data mining techniques that try to extract patterns from this data. Web usage mining has recently attracted attention as a viable framework for extracting useful access pattern information, such as user profiles, from massive amounts of web log data for the purpose of web site personalization and organization. In this paper we present an approach for discovering and tracking evolving user profiles. We also describe how the discovered user profiles can be enriched with explicit information need that is inferred from search queries extracted from web log data.*

1. INTRODUCTION

Web usage mining [1] use various data mining or machine learning techniques to model and understand web user activity, as clustering for user session clustering [2], adoptive web site [3] for automatically synthesize index pages, association rule discovery modeling, probabilistic grammars models. Also Fuzzy relational user profile clustering techniques [4] [5], were used to discover user profiles. Most previous research efforts in web usage mining have worked with the assumption that the web usage data is static. But because of the dynamic nature of web data it is desirable to study and discover web usage patterns at a higher level, where such dynamic tendencies and temporal events can be distinguished. Mining evolving click streams is the subject of only a few recent research efforts [6]. All this approach were proposed within a supervised learning framework or focus on adaptation to a single user (predicting whether an object is relevant or not). On the other hand, the proposed work is based on an unsupervised learning framework that tries to learn mass anonymous user profiles on the server side.

Web usage mining has recently attracted attention as a viable framework for extracting useful access pattern information, such as user profiles, from massive amounts of Web log data for the purpose of Web site personalization and organization. Most efforts have relied mainly on clustering or association rule discovery as the enabling data mining technologies. Typically, data mining has to be completely re-applied periodically and offline on newly generated Web server logs in order to keep the discovered knowledge up to date.

2. TAXANOMY OF WEB MINING

Web mining is the use of data mining techniques to automatically discover and extract information from web documents and services. Web mining research can be classified categories: Web content mining (WCM), Web structure mining (WSM), and Web usage mining (WUM).

2.1 Web Content Mining

Web content mining describes the discovery of useful information from content of web pages or web document. According to the difference of mining objects, web content mining consists of text mining (including data as text, hypertext, html document as well as data in tables.) and multimedia data mining (including the data such as images, audio, and video).Currently, the web text mining are mainly used in summarization, classification, clustering and association analysis of the web document-sets, browser navigation of science literature and trend prediction with web document.

2.2 Web Structure Mining

Data that describes the organization of the content. Intra page structure information includes the arrangement of various HTML or XML tags within a given page. This can be represented as a tree structure, where the HTML tag becomes the root of the tree. The principal kind of inter page structure information is hyperlink connecting one page to another.

2.3 Web Usage Mining

Usage data represent a Web site's usage, such as a visitor's IP address, time and date of access, complete path (files or directories) accessed, referrers address, and other attributes that can be included in a Web access log. The goal of web usage mining is to find the user's access patterns quickly, such as the frequent traversal paths, frequent access page-set, and user clustering. Web usage mining mines the data generated by the web users visiting recorders while the interaction with the web. The mining processes include data preparation, mining process, and process analysis.

Web usage mining can use various data mining or machine learning techniques to model and understand Web user activity. Clustering was used to segment user sessions into clusters or profiles that can later form the basis for personalization. The notion of an adaptive Web site was proposed, where the user's access pattern can be used to automatically synthesize index pages. New fuzzy relational clustering techniques were used to discover user profiles that can overlap, whereas robust clustering was proposed to mine profiles that are resistant to noise that is naturally present in click-stream data.

Traditionally web usage mining is performed as

1. collection of Web data such as activities/click-streams

recorded in Web server logs,

2. preprocessing of Web data such as filtering crawlers requests, requests to graphics, and identifying unique sessions,
3. analysis of Web data, also known as Web Usage Mining [4], to discover interesting usage patterns or profiles, and
4. Interpretation/evaluation of the discovered profiles.

After interpretation of discovered profile there is need of tracking the evolution of discovered profiles. Tracking the evolution of discovered profiles will generate better understanding of evolution of user access pattern and seasonality.

Most previous research efforts in Web usage mining have worked with the assumption that the Web usage data is static. However, the dynamic aspects of Web usage have recently become important. This is because Web access patterns on a Web site are dynamic due not only to the dynamics of Web site content and structure but also to changes in the user's interests and, thus, their navigation patterns. Thus, it is desirable to study and discover Web usage patterns at a higher level, where such dynamic tendencies and temporal events can be distinguished. Mining evolving click-streams [6] is the subject of only a few recent research efforts. Learning evolving concepts adds another layer of difficulty to the process of online learning, since concepts can no longer be assumed to be constant. In an evolving scenario, with time, past training examples may become obsolete and therefore need to be replaced by more recent examples.

3. PROFILE DISCOVERY BASED ON USAGE MINING

3.1 Preprocessing the Web Log File to Extract User Sessions

Each access log entry consists of :(i) User's IP address,(ii) Access time, (iii) Request method ("GET", "POST", …, etc), (iv) URL of the page accessed, (v) Data transmission protocol (typically HTTP/1.0), (vi) Return code,(vii) Number of bytes transmitted. First, we filter out log entries that are not germane for our task. These include entries that: (i) result in any error (indicated by the error code), (ii) use a request method other than "GET", or (iii) accesses to image files (.gif, .jpeg, ,…, etc.), which hare typically embedded in other pages and are only transmitted to the user's machine as a byproduct of the access to a certain web page which has already been logged. Next, analogous to [6], the individual log entries are grouped into user sessions. A user session is defined as a sequence of temporally compact accesses by a user. Since web servers do not typically log usernames (unless identd is used), we define a user session as accesses from the same IP address such that the duration of time elapsed between any two consecutive accesses in the session is within a pre specified threshold. Each URL in the site is assigned a unique number. Thus ith session is encoded as N_u dimensional binary attribute vector $s^{(i)}$ with the property

$$s_j^{(i)} = \begin{cases} 1 & \text{if the user accessed the } j^{th} \text{ URL} \\ & \text{during the } i^{th} \text{ session} \\ 0 & \text{otherwise} \end{cases}$$

3.2 Clustering Sessions into an Optimal Number of Categories

To cluster user sessions, we use H-UNC [10], a divisive hierarchical version of a robust clustering approach (Unsupervised Niche Clustering (UNC)) that uses a Genetic Algorithm (GA) to evolve a population of candidate solutions through generations of competition and reproduction. The main outline of the H-UNC algorithm is sketched in the following. The reason that we use H-UNC instead of other clustering algorithms is that unlike most other algorithms, H-UNC can handle noise in the data and automatically determines the number of clusters. In addition, evolutionary optimization allows the use of any domain-specific optimization criterion and any similarity measure, in particular a subjective measure that exploits domain knowledge or ontologies. However, unlike purely evolutionary search-based algorithms, H-UNC combines evolution with local Piccard updates to estimate the scale σ_i of each profile, thus converging fast (about 20 generations). HUNC is outlined as follows.

ALGORITHM: Unsupervised Niche Clustering Algorithm (UNC) [30]:
INPUT: data records (in this case user sessions)
OUTPUT: Cluster representatives (profile p_i = set of URLs), scales σ_i

-Randomly select an initial population of N_p candidate representatives from input data;
-Set initial scales σ_i = small fraction (1/10) of upper bound estimate on inter-point distance in data set (in this case upper bound =1);
-Repeat for G generations {
-Update the distance d_{ij} of each data record x_j relative to each candidate cluster representative p_i using distance defined in Section C;
-Update the robust weight w_{ij} of each data record x_j relative to each candidate cluster representative p_i: $w_{ij} = e^{-d_{ij}/(2\sigma_i^2)}$
-Update the scale σ_i for each candidate cluster representative (derived by setting $\partial f_i / \partial \sigma_i = 0$): $\sigma_i = \frac{\sum_j w_{ij} d_{ij}}{\sum_j w_{ij}}$
-Update fitness f_i of each candidate cluster representative p_i: $f_i = \frac{\sum_j w_{ij}}{\sigma_i}$
-FOR i = 1 TO $N_p/2$ DO {
-Select randomly without replacement a candidate parent p_i from the population;
-Select randomly without replacement another candidate parent p_k from the population;
-Obtain children c1 and c2 by performing crossover and mutation between the chromosome strings of p_i and p_k;
-Update the scale σ_i and the fitness f_i of each child;
-Apply Deterministic Crowding as replacement policy to fill new population:
-First, assign each child to closest parent;
-IF child's fitness > closest parent's fitness THEN
- child replaces parent in the new population;
-ELSE
- parent remains in the new population;
}
}

3.3 Similarity Measure Used in Clustering

The similarity score between an input session s and the i^{th} profile p_i can be computed using the cosine similarity as follows (where Nu is the total number of URLs):

$$S_{si}^{cosine} = \frac{\sum_{k=1}^{N_{ij}} p_{ik} s_k}{\sqrt{\sum_{k=1}^{N_{ij}} p_{ik} \sum_{k=1}^{N_{ij}} s_k}}.$$

For hierarchical Web site structure following similarity measures are taken into consideration:

$$S_{si}^{web} = \max\left\{ \frac{\sum_{i=1}^{N_{ij}} \sum_{k=1}^{N_{ij}} p_{ik} S_u(i,k) s_k}{\sum_{k=1}^{N_{ij}} p_{ik} \sum_{k=1}^{N_{ij}} s_k}, S_{si}^{cosine} \right\}.$$

where $S_u(i.j)$ is a URL to the URL similarity function that is computed based on the amount of overlap between the paths P_i and P_j leading from the root of the Web site (the main page) to any two URLs i and j. This is given by

$$S_u(i,j) = \begin{cases} 1 & if\ i = j, \\ \min\left(1, \frac{|P_i \cap P_j|}{\max(1, \max(|P_i|, |P_j|) - 1)}\right) & otherwise. \end{cases}$$

3.4 Post processing and Enrichment of Session Cluster Into Multifaceted User Profiles

After automatically grouping sessions into different clusters, we summarize the session categories in terms of user profile vectors [3], [4] p_i. The k^{th} component/weight of this vector (pik) captures the relevance of URL_k in the i^{th} profile, as estimated by the conditional probability that URL_k is accessed in a session belonging to the i^{th} cluster (this is the frequency with which URL_k was accessed in the sessions belonging to the i^{th} cluster).The profiles are then converted to binary vectors (sets) so that only URLs with weights > 0:15 remain. The model is further extended to a robust profile [2], [3] based on robust weights (w_{ij}) computed in the UNC algorithm that assign only sessions with high robust weights (that is, $w_{ij} > w_{min}$) to a cluster's core. The core of a profile consists only of sessions that are very similar to the representative profile. Thus, noisy sessions are eliminated from the re-computation of profiles. Each profile pi is discovered along with an automatically determined measure of scale σ_i that represents the amount of variance or dispersion of the user sessions in a given cluster around the cluster representative (profile). This measure will later serve an important role in determining the boundary of each cluster and thus allows us to automatically determine whether two profiles are compatible or not.

4. TRACKING EVOLVING USER PROFILES

Tracking different profile events across different time periods can generate a better understanding of the evolution of user access patterns and seasonality. Note that both profiles and click streams are typically evolving, since the profiles are nothing more than summaries of the click streams, which are themselves evolving. Each profile p_i is discovered along within automatically determined measure of scale σ_i that represents the amount of variance or dispersion of the user sessions in a given cluster around the cluster representative. This measure issued to determine the boundary around each cluster (an area located at a distance σ_i from the profile pi) and thus allows us to automatically determine whether two profiles are compatible. Two profiles are compatible if their boundaries overlap. The notion of compatibility between profiles is essential for tracking evolving profiles. After mining the Web log of a given period, we perform an automated comparison between all the profiles discovered in the current batch and the profiles discovered in the previous batch by a sequence of SQL queries on the profiles that have been stored in a database, as shown in the "TrackProfiles" Algorithm. A typical query for retrieving corresponding profiles between Periods T_I and T_{I+1} is "SELECT ThisProfile, TothisProfile FROM ProfileTrailHERE Period = T_I."

ALGORITHM: TrackProfiles
Input: - Discovered Profiles for all Time Periods stored in Database //(profile = set of relevant URLs, and scale σ_i)
- Beginning Time Period T_1, Ending Time Period T_k
Output: ProfileTrail: Profile-to-Profile tracking Table from Time Period T_1 to Time Period T_k // (e.g. Table 4)

For I = Time Period T_1 to Time Period T_k do
For J = first profile in Time Period I to last profile in Time Period I do
For K = first profile in Time Period I+1 to last profile in Time Period I+1 DO {
Distance[k]= S_{web}(profile$_I$, profile$_K$);
IF Distance[k] <σ_J THEN Insert into ProfileTrail (Period, ThisProfile, TothisProfile) values(I, Profile[J], K);
}

We define a profile evolution event as a coarse categorization of possible real evolution scenarios that relate how profiles that are discovered during a certain period relate to profiles discovered in another period. The above comparison process determines which new profiles are compatible with the old profiles and which new profiles are incompatible with any previous profile. These last two cases, respectively, give rise to two kinds of events: Persistence and Birth. A third event Death arises in case an old profile does not find a compatible profile from the new batch. It is also possible to track profile reemergence in the long term. This is the case of an old profile that disappears and then reappears when it is found to be compatible with a new profile in the current batch. This event is labeled as Atavism. We can visualize the temporal dynamics of profiles birth, persistence, death, and atavism (rebirth) by labeling the x-axis with the periods corresponding to the different Web log batches that undergo Web usage mining: period 1, period 2, etc. On the other hand, the y-axis is used to indicate the profile index: New profiles are vertically expanded by adding new indices on top of existing ones. Finally, we generate a plot depicting the Web site user trend evolution by adding a special symbol whenever profile y appears in period x and possibly adding event labels such as Birth, Death, and Atavism, as these occur.

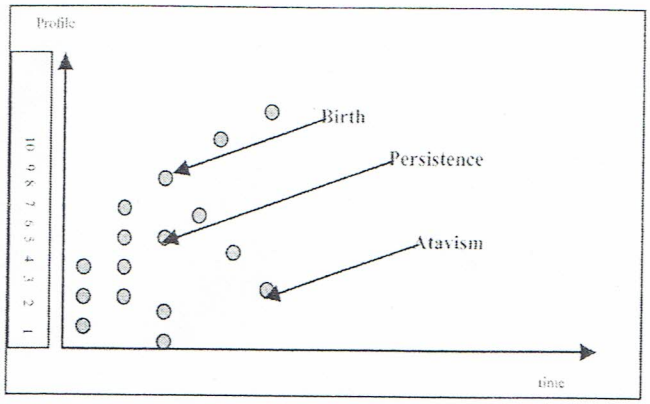

Fig.1 Profile Evolution.

Note that this tracking takes advantage of a database management system to accelerate the access to archived user profiles (which, as a summary, are a negligible number compared to the original input data). Moreover, this process is done offline and is only periodically done (not adding any burden on the data mining/clustering itself), since it is an offline analysis of the results of Web usage mining to help track the user profiles evolution in retrospect. The choice of the basic period length can be either arbitrary or based on the domain knowledge and intuition (like whether changes have been made to the Web site or whether new events related to the Web site domain may have occurred). Thus, the right period length should be determined by trial and error.

The analysis of profile evolution, as shown in Fig. 1, can improve our understanding of the user activity trends and detect seasonality in their access patterns, especially over a long time span. It also helps in implementing a dynamic recommendation strategy, for instance, by caching frequently reemerging (atavistic) profiles. Dead profiles can be relegated to the secondary memory for possible reemergence, whereas persistent profiles can be kept in the primary memory for fast access and then relegated to the secondary memory when they die. Similarly, dead profiles that have been persistent during an earlier period should be distinguished from dead profiles that have never been persistent, that is, volatile profiles.

5. CONCLUSION

Web usage mining is growing rapidly since its inception, and new methodologies are being developed using various approaches. In this paper, we have summarized the different types of web mining and provide approach which will be useful for tracking user profiles. Tracking evolving user profiles can generate better understanding of user access pattern and seasonality; which will be helpful for website personalization, website modification, business intelligence, and system modification.

REFERENCES

[1] Lappas G (2007), "An Overview of Web Mining in Societal Benefit Areas", The ninth IEEE International Conference on E-Commerce Technology and fourth IEEE International Conference on Enterprise Computing, E-Commerce and E-Services. Vol. 22-26, pp.683-690.

[2] Pal S K (2002), "Web Mining in Soft Computing Framework: Relevance, State of the Art and Future Directions", IEEE Transaction on Neural Networks, Vol.13, pp.1163-1177.

[3] Perkowitz M, Etzioni O (1997), "Adaptive Web Sites: Automatically Learning for User Access Pattern", Proc. Sixth International WWW Conf.,Vol.5, pp.205-218.

[4] Nasraoui O, Krishnapuram R, Frigui H, Joshi A (2002), "Extracting Web User Profiles Using Relational Competitive Fuzzy Clustering", Int'l J. Artificial Intelligence Tools, Vol.9, pp.509-526.

[5] Yan C, Shen J, Peng Q, Pan D (2005), "PARALLEL WEB MINING FOR LINK PREDICTION IN CLUSTER SERVER", Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Vol.4, pp.2291-2295.

[6] Nasraoui O, Cardona C, Rojas C, Gonzalez F (2003), "Mining Evolving User Profiles in Noisy Web Click stream Data with a Scalable Immune System Clustering Algorithm", Proc. Workshop Web Mining as a Premise to Effective and Intelligent Web Applications, pp. 71- 81.

[7] Luo Jianli, Shen Jie, Xu Youzhi, (2004), "A Log Mining Model Based On Distributed Web Servers", Computer Applications and Software, Vol. 21, No. 9, pp. 30-35.

[8] Nasraoui O, Rojas C, Cardona C (2006), "A Framework for Mining Evolving Trends in Web Data Streams Using Dynamic Learning and Retrospective Validation," Computer Networks, Special issue on Web dynamics, Vol. 50, pp. 23-30.

[9] Srivastava J, Cooley R, Deshpande M (2000), "Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data",SIGKDD Explorations, Vol. 1, no. 2, pp. 1-12.

[10] Desikan P, Srivastava J (2004), "Mining Temporally Evolving Graphs",Proc. Workshop Web Mining and Web Usage Analysis, Vol.4, pp.2284-2287.

[11] Christos B, Giorgos K, Ioannis M (2007), "A Web Content Manipulation Technique Based on Page Fragmentation", Journal of Network and Computer Applications, Vol. 30, pp.563-585.

[12] Ganesan P, Garcia-Molina H, Widom J (2003), "Exploiting Hierarchical Domain Structure to Compute Similarity", ACM Trans. Information Systems,Vol. 21, No. 1, pp. 64-93.

[13] Oberle D, Berendt B, Hotho A, Gonzalez J (2003), "Conceptual User Tracking", Proc. First Int'l Atlantic Web Intelligence Conf. (AWIC '03).

[14] Nasraoui O,Goswami S (2006), "Mining and Validating Localized Frequent Itemsets with Dynamic Tolerance", Proc. Sixth SIAM Int'l Conf.Data Mining (SDM '06), pp. 578-582.

[15] Nasraoui O (2008)," A Web Usage Mining Framework for Mining Evolving User Profiles in Dynamic Web Sites", IEEE International Conference on Knowledge and Data Engineering.

[16] Li M, Chen A (2008), "A Web Mining Based Measurement and Monitoring Model of Urban Mass Panic in Emergency Management", Fifth International Conference on Fuzzy Systems and Knowledge Discovery, Vol. 40, pp. 23-30.

[17] Suresh R M (2007), " A study on the ontology based web mining for digital library", IET-UK International Conference on Information and Communication Technology in Electrical Sciences (ICTES 2007), Dr. M.G.R. University, Chennai, Tamil Nadu, India. Dec. Vol.20, pp.1096-1100.

[18] Yin C, Liu S, Chen L, Ye X(2008), "E-Government Web Mining Tool Design and Implementation Based on the Semantic Web", IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Vol.20, pp.1296-1305.