Ontology-Based Partitioning of Data Steam for Web Mining: A Case Study of Web Logs

Jason J. Jung

School of Computer and Information Engineering, Inha University,
253 Yonghyun-dong, Incheon, Korea 402-751
j2jung@intelligent.pe.kr

Abstract. This paper presents a nevel method partitioning steaming data based on ontology. Web directory service is applied to enrich semantics to web logs, as categorizing them to all possible hierarchical paths. In order to detect the candidate set of session identifiers, semantic factors like semantic mean, deviation, and distance matrix are established. Eventually, each semantic session is obtained based on nested repetition of top-down partitioning and evaluation process. For experiment, we applied this ontology-oriented heuristics to sessionize the access log files for one week from IRCache. Compared with time-oriented heuristics, more than 48% of sessions were additionally detected by semantic outlier analysis.

1 Introduction

As the concern for searching relevant information from the web has been exponentially increasing, the very large amount of log data have been generated in web servers. Thus, many applications have been focusing on various ways to analyze them in order to recognize the usage patterns of users and discover other meaningful patterns [1], [5]. Among the whole steps of web user profiling mentioned in [2], we have taken the session identification for segmenting web log data in consideration. For partitioning each user activity into sequences of entries corresponding to each user visit, mainly two kinds of sessionization heuristics, which are time-oriented heuristics [3] and navigation-oriented heuristics [4] have been introduced. However, knowledge extractable from sessions identified by those heuristics is limited like frequent and sequential patterns represented by URLs. It means that web logs has to be sessionized with semantic enrichment based on ontology in order to find out more potential and meaningful information like a user's preference and intention. More importantly, web caching(or proxy) servers have to track streaming URL requests from multiple clients, because they have to increase predictability for prefetching web content that is expected in next request. Enriching web logs with their corresponding semantic information has been attempted in some studies [6], [10] such as mapping URLs to set of concepts as a feature vector and a specific value, respectively. We present conceptualizing an URL information itself by using web directory and introduce representing conceptualized URLs as tree-like information.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3036, pp. 247–254, 2004.

2 Data Model of Web Log and Problem Statement

Several standard data models of web logs, generally, have some problems to analyze these web logs such as their anonymity, rotating IP addresses connections through dynamic assignment of ISPs, missing references due to caching, and inability of servers to distinguish among different visits. Therefore, we note the problem statements concentrated for semantic sessionization in this paper, as follows.

- **Weakness of IP address field as session identifier.** The same IP address field in a web logs (within the time window or not) can not guarantee that those requests are caused by only one user, and reversely, requests from the different IPs can be generated by a particular user.
- **Simultaneous user requests based on multiple intention.** It means we have to consider multiple intention of users by classifying mixed logs according to the corresponding semantics.

Each request consists of timestamp, IP address, and URL fields. URL field is divided into base url and reminder, which are the host name of web server and the rest part of full URL, respectively. Then, we assume that each URL is semantically characterized by its base URL. For example, we are given a web log composed of eight requests ordered by timestamps from t_1 to t_8. We denote the URL set of sequential requests by $< \ldots, b\_url_i{+}r_j, \ldots >$ mapped to the timestamps $< \ldots, t_i, \ldots >$. These logs are partitioned with respect to an IP address ip_i. After partitioning, we compare semantic distance between base URLs in a set of requests, because we regard a semantic session as the sequence of URL having similar semantics. In other words, we investigate if an user's intention is retained or not.

3 Ontology-Oriented Heuristics for Sessionization

An ontology, a so-called semantic categorizer, is an explicit specification of a conceptualization. It means that ontologies can play a role of enriching semantic or structural information to unlabeled data. Web directories like Yahoo and Cora can be used to describe the content of a document in a standard and universal way as ontology [7]. Besides, web directory is organized as a topic hierarchical structure which is an efficient way to organize, view, and explore large quantities of information that would, otherwise, be cumbersome [9].

In this paper we assume that all URLs can be categorized by a well-organized web directory service. There are, however, some practical obstacles to do that, because most of web directories are forced to manage a non-generic tree structure in order to avoid a waste of memory space caused by redundant information [8]. We briefly note that problems with categorizing an URL with web directory as an ontology are the following:

- **The multi-attributes of an URL.** An URL can be involved in more than a category. The causal relationships between categories makes their

hierarchical structure more complicated. As shown in Fig. 1 (1), an URL can be included in some other categories, named as A or B.

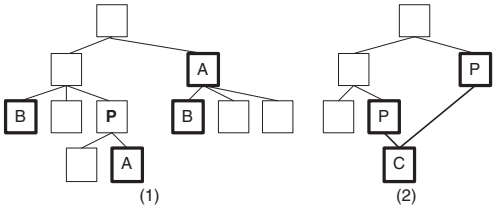

Fig. 1. (1) The multi-attribute of URLs; (2) The subordinate relationship between two categories

– **The relationship between categories.** A category can have more than a path from root node. As shown in Fig. 1 (2), the category C can be a subcategory of more than one like P. Furthermore, some categories can be semantically identical, even if they have different labels.
  - Redundancy between semantically identical categories
  - Subordination between semantically dependent categories

In order to simply handle these problems, we categorize each URL to all possible categories causally related with itself. Therefore, an URL url_i is categorized to a category set $Category(url_i)$, and the size of this category set depend on the web directory. Each element of a category set is represented as a path from the root to the corresponding category on web directory. Let the base URLs {b_url$_1$, b_url$_2$, b_url$_3$} semantically enriched to {<a:b:d, a:b:f:k>, <a:c:h>, <a:b:f:j>}. The leftmost concept "a" is indicating the root of web directory and these base URLs are categorized to <d, k>, <h>, and <j>, respectively. In particular, due to multi-attribute of base URL b_url$_1$, $Category$(b_url$_1$) is composed of two different concepts.

We define semantic factors measuring the relationship between two log data. All possible categorical and ordered paths for the requested URL, above all, are obtained, after conceptualizing this URL by web directory. Firstly, the semantic distance is formulated for measuring the semantic difference between two URLs. Let an URL url_i categorized to the sets $\{path_i | path_i^m \in Category(url_i), m \in [1, \ldots, M]\}$ where M is the number of total categorical paths. As simply extending Levenshtein edit distance, the semantic distance $\Delta^\diamond$ between two URLs url_i and url_j is given by

$$\Delta^\diamond[url_i, url_j] = \arg \min_{m=1,n=1}^{M,N} \frac{\min\left((L_i^m - L_C^{(m,n)}), (L_j^n - L_C^{(m,n)})\right)}{\exp(L_C^{(m,n)})} \qquad (1)$$

where L_i^m, L_j^n, and $L_C^{(m,n)}$ are the lengths of $path_i^m$, $path_j^n$, and common part of both of them, respectively. As marking paths representing conceptualized URLs

on trees, we can easily get this common part overlapping each other. $\Delta^{\diamond}$ compares all combination of two sets ($|path_i| \times |path_j|$) and returns the minimum among values in the interval [0, 1], where 0 stands for complete matching. Exponent function in denominator is used in order to increase the effect of $L_C^{(m,n)}$. Second factor is to aggregate URLs during a time interval. Thereby, semantic distance matrix $D_{\Delta^{\diamond}}$ is given by

$$D_{\Delta^{\diamond}}(i,j) = \begin{bmatrix} \dots & \dots & \dots \\ \dots \Delta^{\diamond}[url_{t_i}, url_{t_j}] & \dots & \dots \\ \dots & \dots & \dots \end{bmatrix} \tag{2}$$

where the predefined time interval T is the size of matrix and diagonal elements are all zero. Based on $D_{\Delta^{\diamond}}$, the semantic mean $\mu^{\diamond}$ is given by

$$\mu^{\diamond}(t_1, \dots, t_T) = \frac{2\sum_{i=1}^{T}\sum_{j=i}^{T} D_{\Delta^{\diamond}}(i,j)}{T(T-1)} \tag{3}$$

where $D_{\Delta^{\diamond}}(i,j)$ is the (i,j)-th element of distance matrix. This is the mean value of upper triangular elements except diagonals. Then, with respect to the given time interval T, the semantic deviation $\sigma^{\diamond}$ is derived as shown by

$$\sigma^{\diamond}(t_1, \dots, t_T) = \sqrt{\frac{2\sum_{i=1}^{T}\sum_{j=i}^{T}\left(D_{\Delta^{\diamond}}(i,j) - \mu^{\diamond}(t_1, \dots, t_T)\right)^2}{T(T-1)}} \tag{4}$$

These factors are exploited to quantify the semantic distance between two random logs and statistically discriminate semantic outliers such as the most distinct or the N distinct data from the rest in the range of over pre-fixed threshold, with respect to given time interval.

When we try to segment web log dataset, log entries are generally time-varying, more properly, streaming. In case of streaming dataset, not only semantic factors in a given interval but also the distribution of the semantic mean $\mu^{\diamond}$ is needed for sessionization. This will be described in the Sect. 4. We, hence, simply assume that a given dataset is time-invariant and its size is fixed in this section.

In order to analyze semantic outlier for sessionization, we regard the minimize the sum of partial semantic deviation $\mu^{\diamond}$ for each session as the most optimal partitioning of given dataset. Thereby, the principle session identifiers $PSI = \{psi_a | a \in [1, \dots, S-1], psi_a \in [1, \dots, T-1]\}$ is defined as the set of boundary positions, where the variables S and T are the required number of sessions and the time interval, respectively.

The semantic outlier analysis for sessionizing static logs SOA_S as objective function with respect to PSI is given by

$$SOA_S(PSI) = \sum_{i=1}^{S} \mu_i^{\diamond} \tag{5}$$

where $\mu_i^{\diamond}$ means partial semantic deviation of i^{th} segment. In order to minimize this objective function, we scan the most distinct pairs, in other words, the largest value in the semantic distance matrix $D_{\Delta^{\diamond}}$, as follows:

$$\Delta_{MAX}^{\diamond}[T_a, T_b] = \arg \max_{i=1,j=1}^{T} D_{\Delta^{\diamond}}(i,j) \tag{6}$$

where $\arg\max_{i=1}^{T}$ is the function returning the maximum values during a given time interval $[T_a, T_b]$. When we obtain $D_{\Delta^{\diamond}}(p,q)$ as the maximum semantic distance, we assume there must be at least a principle session identifier between p^{th} and q^{th} URLs. Then, the initial time interval $[T_a, T_b]$ is replaced by $[T_p, T_q]$, and the maximum semantic distance in reduced time interval is scanned, recursively. Finally, when two adjacent elements are acquired, we evaluate this candidate *psi* by using $SOA_S(psi)$. If this value is less than $\sigma^{\diamond}$, this candidate *psi* is inserted in PSI. Otherwise, this partition by this candidate *psi* is cancelled. This sessionization process is top-down approaching, until the required number of sessions S is found. Furthermore, we can also be notified the oversessionization, which is a failure caused by overfitting sessionization, detected by the evaluation process $SOA_S(PSI)$.

4 Session Identification from Streaming Web Logs

Actually, on-line web logs are continuously changing. It is impossible to consider not only the existing whole data but also streaming data. We define the time window W as the pre-determined size of considerable entry from the most recent one. Every time new URL is requested, this time window have to be shifted. In order to semantic outlier analysis of streaming logs, we focus on not only basic semantic factors but also the distribution of the semantic mean with respect to time window, $\mu^{\diamond}(W^{(T)})$.

As extending SOA_S, the objective function for analyzing semantic outlier of dynamic logs SOA_D is given by

$$SOA_D^{W^{(i)}}(PSI) = \sum_{k=1}^{S} \mu_k^{\diamond}|_{W^{(i)}} \tag{7}$$

where the $W^{(i)}$ means that the time window from i^{th} URL is applied. We want to minimize this $SOA_D(PSI)$ by finding the most proper set of principle session identifiers. The candidate psi_i is estimated by the difference between the semantic means of contiguous time windows and predefined threshold ε, as shown by

$$\left| \mu^{\diamond}(W^{(i)}) - \mu^{\diamond}(W^{(i-\tau)}) \right| \geq \varepsilon \tag{8}$$

where τ is the distance between both time windows and assumed to be less than the size of time window $|W|$. Similar to the evaluation process of SOA_S, once a candidate psi_i is obtained, we evaluate it by comparing $SOA_D^{W^{(i)}}$ and

$SOA_D^{W^{(i-1)}}$. Finally, we can retrieve PSI to sessionize streaming web logs. In case of streaming logs, more particularly, a candidate psi meeting the evaluation process can be appended into unlimited size of PSI.

5 Experiments and Discussion

For experiments, we collected the sanitized access logs from sv.us.ircache.net, one of web cache servers of IRCache. These raw files, generated from 20 March 2003 to 26 March 2003, consist of 11 attributes and about 9193000 entries. We verified sessionizing process proposed in this paper on a PC with a 1.2 GHz CPU clock rate, 256 MB main memory, and running FreeBSD 5.0. During data cleansing, logs whose URL field is ambiguous (wrong spelling or IP address) are removed, as referring to web directory.

Table 1. The number of sessions by time-oriented heuristics and ontology-oriented heuristics (static and dynamic logs) from logs for seven days (20-36 March 2003).

	1	2	3	4	5	6	7
Time-oriented	1563	1359	1116	877	1467	1424	1384
Ontology-oriented	907	923	692	421	807	783	844
(Static logs, SOA_S)	(58%)	(68%)	(62%)	(48%)	(55%)	(55%)	(61%)
Ontology-oriented	983	1051	939	683	1118	827	1105
(Dynamic logs, SOA_D)	(63%)	(77%)	(84%)	(78%)	(76%)	(58%)	(80%)
Common Session Boundary	47%	51%	49%	48%	57%	32%	74%

We compared two sessionizations based on time oriented and ontology oriented heuristics, with respect to the number of segmented sessions and the reasonability of association rules extracted from them. In case of ontology-oriented sessionization, fields related with time such as “Timestamp” and “Elapsed Time” were filtered. Time-oriented heuristics simply sessionized log entries between two sequential requests whose difference of field “Timestamp” is more than 20 milliseconds with respect to the same IP address. On the other hand, for ontology-oriented heuristics, the size of time window W was predefined as 50. The numbers of sessions generated in both cases are shown in Table 1. Time-oriented heuristics estimate denser sessionization than two ontology-oriented approaches. It means that ontology-oriented heuristics based on SOA_S or SOA_D, generally, can make URLs requested over time gap semantically connected each other. They, SOA_S or SOA_D, decreased the number of sessions to, overall, 58.14% and 73.71%, respectively, compared to time-oriented heuristics. Even though ontology-oriented heuristics searched fewer sessions, the rate of common session boundaries (the number of common sessions matched with time-oriented heuristics over the number of sessions of SOA_D) is average 51.1%. It shows that more than 48% of sessions not segmented by time-oriented heuristics can be detected

by semantic outlier analysis. While time oriented sessionization is impossible to recognize patterns of users who is easily changing their preferences or simultaneously trying to search various kinds of information on the web, ontology-oriented method can discriminate these complicated patterns.

Table 2. Evaluation of the reasonability of the extracted ruleset (hit ratio (%))

	1	2	3	4	5	6	7
Time-oriented	0.06	0.32	0.46	0.41	0.51	**0.52**	0.49
Static logs, SOA_S	0.05	0.45	0.66	0.72	**0.76**	0.74	0.75
Dynamic logs, SOA_D	0.05	0.46	0.52	0.67	0.70	**0.75**	0.72

We also evaluated the reasonability of the rules extracted from three kinds of session sequences. According to the standard *least recently used* (LRU), we organized the expected set of URLs, which means the set of objects that cache server has to prefetch. The size of this set is constantly 100. As shown in Table 2, we measured the two hit ratios by both of their sessionizations for seven days. The maximum hit ratios in three sequences were obtained 0.52, 0.76, and 0.75, respectively. Ontology-oriented sessionization SOA_S acquired about 24.5% improvement of prefetching performance, compared with time-oriented. Moreover, we want to note that the difference between SOA_S and SOA_D. For the first three days, the hit ratio of SOA_S was higher than that of SOA_D by over 5%. Because of streaming data, SOA_D showed the difficulty in initializing the ruleset. After initialization step, however, the performances of SOA_S and SOA_D were converged into a same level.

6 Conclusions and Future Work

In order to mine useful and significant association rules from web logs, many kinds of well-known association discoverying methods have been developed. Due to the domain specific properties of web logs, sessionization process of log entries is the most important in a whole step. We have proposed ontology-oriented heuristics for sessionizing web logs. In order to provide each requested URL with the corresponding semantics, web directory service as ontology have been applied to categorize this URL. Especially, we mentioned three practical problems for using real non-generic tree structured web directories like Yahoo. After conceptualizing URLs, we measured the semantic distance matrix indicating the relationships between URLs within the predefined time interval. Additionally, factors like semantic mean and semantic deviation were formulated for easier computation. We considered two kinds of web logs which are stationary and streaming. Therefore, two semantic outlier analysis approaches SOA_S and SOA_D were introduced based on semantic factors. Through the evaluation process, the de-

tected cadidate semantic outliers were tested whether their sessionization is reasonable or not. According to results of our experiments, investigating semantic relationships between web logs is very important to sessionize them. Classifying semantic sessions, 48% of total sessions, brought about 25% higher prefetching performance, compared with time-oriented sessionization. Complex web usage patterns seemed to be meaninglessly mixed along with "time" can be analyzed by ontology.

References

1. Cooley, R., Srivastava, J., Mobasher, B.: Web Mining: Information and Pattern Discovery on the World Wide Web. Proc. of the 9th IEEE Int. Conf. on Tools with Artificial Intelligence (1997)
2. Mobasher, B., Cooley, R., Srivastava, J.: Automatic personalization based on Web usage mining. Comm. of the ACM **43**(8) (2000)
3. Berendt, B., Mobasher, B., Nakagawa, M., Spiliopoulou, M.: The Impact of Site Structure and User Environment on Session Reconstruction in Web Usage Analysis. Proc. of the 4th WebKDD Workshop at the ACM-SIGKDD Conf. on Knowledge Discovery in Databases (2002)
4. Chen, Z., Tao, L., Wang, J., Wenyin, L., Ma, W.-Y.: A Unified Framework for Web Link Analysis. Proc. of the 3rd Int. Conf. on Web Information Systems Engineering (2002) 63–72
5. Cooley, R., Mobasher, B., Srivastava, J.: Data Preparation for Mining World Wide Web Browsing Patterns. Knowledge and Information Systems **1**(1) (1999) 5–32
6. Dai, H., Mobasher, B.: Using ontologies to discover domain-level web usage profiles. Proc. of the 2nd Semantic Web Mining Workshop at the PKDD 2002 (2002)
7. Labrou, Y., Finin, T.: Yahoo! as an Ontology: Using Yahoo! Categories to Describe Documents. Proc. of the 8th Int. Conf. on Information Knowledge Management (1999) 180–187
8. Jung, J.J., Yoon, J.-S., Jo, G.-S.: Collaborative Information Filtering by Using Categorized Bookmarks on the Web. Proc. of the 14th Int. Conf. on Applications of Prolog (2001) 343–357
9. McCallum, A., Nigam, K., Rennie, J., Seymore, K.: Building Domain-Specific Search Engines with Machine Learning Techniques. AAAI Spring Symp. (1999)
10. Berendt, B., Spiliopoulou, M.: Analysing navigation behaviour in web sites integrating multiple information systems. The VLDB Journal **9**(1) (2000) 56–75