Web Log Mining for Improvement of Caching Performance

Rudeekorn Soonthornsutee[1], Pramote Luenam

***Abstract*— The objective of this study is to build a model of cache replacement policy for improvement of web caching performance. The integration approach of cluster analysis and classification are used to create a classifier for predicting the cache life time. The data set was collected from the cache of the National Institute Development Administration's proxy servers. The data collection period was from October 2009 to March 2010. There are four main tasks in this study. First, the access log from proxy servers were collected and preprocessing tasks were performed. Second, the access log data were partitioned into clusters based on users' access patterns. Third, classifier models of the cache replacement policy were built and their accuracies were compared. Finally, the efficiency of the selected classifier was compared with other cache replacement algorithms. Results show that overall classification accuracy of the model is satisfactory and the model is efficient and very good in performance.**

***Index Terms*— web log mining, web caching, cache replacement algorithm**

I. Introduction

THE explosive growth of the Internet and the World Wide Web results in network congestion and server overloading. Web caching has been used as one of the effective techniques to reduce network traffic, thereby decrease user access latencies. However, the cache storage space is limited. Some pages must be removed when the cache is full. As a result, the efficiency is dropping from what supposed to be, because the deleted page may be requested again [8]. A lot of studies have been done to improve the Web caching performance [1], [3], [6]. For example, in the study of [5] Web mining technique is applied to predict the future web access. In the study of [7] classification and association rules techniques are used to provide the behavior of website utilization. Similar to several prior studies, we have applied data mining techniques for building the model of cache replacement policy. Nonetheless, instead of relying on one particular technique, we have used the integration approach of two data mining techniques: clustering and classification. The clustering is used to place similar websites into related groups, while the classification is applied to predict the cache life time and then adjust the replacement priority to cache blocks.

Rudeekorn Soonthornsutee is with School of Applied Statistics, National Institute of Development Administration, Bangkok, Thailand; (e-mail: rudeekorn_s.agi@g-able.com).

Pramote Luenam is with School of Applied Statistics, National Institute of Development Administration, Bangkok, Thailand; (e-mail: pramote.l@ics.nida.ac.th).

II. Materials and Methods

To build the model of cache replacement policy, four process steps -- as illustrated in the Fig. 1 -- were performed. First, the access log from the proxy servers were collected and preprocessing tasks were performed. Second, the access log data were partitioned into clusters based on users' access patterns. Third, classifier models of the cache replacement policy were built and their accuracies were compared. Finally, the efficiency of the selected classifier was compared with two traditional cache replacement algorithms: First-In, First-Out (FIFO), and Least Recently Use (LRU). The detailed steps are described as follows:

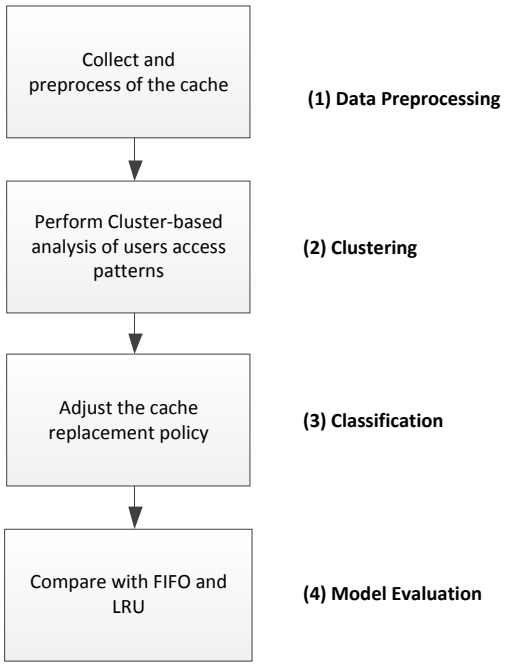

Fig. 1. Framework for model creation of cache replacement policy.

A. Data Preprocessing

The data set was collected from the cache of the National Institute Development Administration (NIDA) using a Blue Coat proxy server (SG 8100-10). The data collection period was from October 2009 to March 2010. Users of this cache include graduate and undergraduate students, faculty, and staff. The data set contains one million records which were randomly selected from an initial set of four million records.

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Each record contains 24 attributes, separated by a white space character, including: localtime, time-taken, c-IP, sc-status, s-action, sc-bytes, cs-bytes, cs-method, cs-uri-scheme, cs-host, cs-uri-port, cs-uri-path, cs-uri-query, cs-username, cs-auth-group, s-hierarchy, s-supplier-name, rs (content-type), cs (referrer), cs (user-agent), sc-filter-result, cs-categories, x-virus-id, and s-IP.

```
"[21/Jan/2010:18:26:31 +0700]" 1 10.10.121.69 200 TCP_HIT 18416
771 GET http ec.atdmt.com 80
/b/050050005TQA/160x600_QatarAirways_FlytoEurope_090909.jpg - -
- DIRECT 65.54.95.86 image/jpeg
http://view.atdmt.com/005/iview/137169601/direct;wi.160;hi.600/01?c
lick= "Mozilla/5.0 (Windows; U; Windows NT 6.1; th; rv:1.9.1.3)
Gecko/20100124 Firefox/3.5.3 GTB5 (.NET CLR 3.5.30729)" PROXIED
"Web Advertisements" - 10.100.100.15
```

Fig. 2. An example of access log record.

The irrelevant and redundant information are filtered out through the feature selection methods. From an initial set of 24, there are only 4 attributes selected. The attributes include localtime, c-ip, cs-host, and cs-categories. All of these attributes are considered as feature variables. These variables are then transformed into forms appropriate for mining (e.g., by performing transformation, calculation, and aggregation operations). In particular, the localtime variable [DD/MMM/YYYY:hh:mm:ss + nnnn] is split into date [DD/MMM/YYYY] and time [hh:mm:ss]. Ten additional derived variables are also formulated: 1) user's request time (hour-of-day), which is measured on a range of 0 to 23; 2) the day-of-week that the request was made; 3) the day-of-month that the request was made; 4) the week-of-year that the request was made; 5) the month-of-year that the request was made; 6) the total time taken for visiting website; 7) the number of requests per month, day, and hour; 8) the relative frequency of request history, which is calculated by deducting the number of requests per day in the past with the average number of requests for each day; 9) the relationship between time and number of requests in the past, which is determined from the correlation between the week that request was made and the number of requests per week for each website; 10) the cache priority, which determines the cache lifetime and is measured on a range of 1 to 10. This variable is what the model predicts for.

The data set was randomly partitioned into three parts: training, test, and validation set. Subsequently, 420,741 records were assigned as the training set. 315,840 records were assigned as the test set and 315,838 records were used as the validation set.

B. Clustering

In this process step, k-means clustering technique is used to group websites into clusters on the basis of users' usage behavior and the categories of websites. The k-means is the simplest clustering algorithm and widely used [9], [10]. The algorithm is used to cluster data based on attributes into k clusters. Each cluster has its center (as known as centroid) at point Cj. The centroid is calculated from mean distance of all records in the cluster [2], [4].

In this study, we make an assumption that users in the same cluster should have same surfing habits and patterns. Users' surfing habits can be determined by several factors such as the time of day of their access, and their most frequently visited websites. Based on the assumption, we define eight input variables: user's request time, the day-of-week, the day-of-month, the week-of-year, the month-of-year, c-IP (requester's IP address), cs-categories (web site's categories), and cs-host (web site's host). By using k-means to cluster, we find that eleven clusters is the best number of clustering. The characteristics of each cluster are illustrated in Table I.

TABLE I
CHARACTERISTICS OF THE ACCESS PATTERNS CLUSTERS.

Cluster	# of Records	Day of Week	(%)	Day	SD	Month	SD	Time	SD	Weeks	SD
1	26580	2	100	22.20	2.16	9.24	0.43	16.27	2.14	5726.22	2.15
2	39321	3	99.29	26.64	2.47	10.02	0.14	11.75	2.43	5730.04	0.27
3	37265	5	100	1.00	0.08	10	0.02	12.79	2.76	5726.00	0.11
4	38242	4	100	25.15	3.09	9.087	0.28	14.18	3.87	5725.68	1.40
5	41270	6	99.15	2.02	0.20	10.01	0.094	12.83	2.87	5726.05	0.47
6	27356	6	99.16	25.02	0.21	9	0.019	13.17	3.13	5725.00	0.10
7	24930	5	67.02	24.65	2.23	9.16	0.37	14.67	3.35	5725.83	1.84
8	17360	7	100	12.10	11.09	9.66	0.56	12.18	3.76	5725.85	1.24
9	89975	5	100	26.82	2.48	9.56	0.50	13.25	2.86	5727.82	2.48
10	35189	3	99.86	22.01	0.19	9	0	12.98	3.08	5725	0
11	43010	6	41.82	23.78	1.39	9	0	12.47	2.98	5725.00	0.09

TABLE I (CONTINUED)
CHARACTERISTICS OF THE ACCESS PATTERNS CLUSTERS.

Cluster	c-IP	(%)	cs-categories	(%)	cs-host	(%)
1	10.10.121.143	19.8	News/Media	22.8	www.manager.co.th	20.5
2	10.10.121.197	18.8	Search Engines/ Portals	23.2	static.sanook.com	14.8
3	10.10.121.33	6.3	News/Media	13.9	www.manager.co.th	10.5
4	10.10.121.143	8.2	none	14.6	my.kapook.com	4.8
5	10.10.121.37	14.8	Social Networking	17.5	www.manager.co.th	4.1
6	10.10.121.69	12.6	none	12.9	my.kapook.com	4.3
7	10.10.121.114	39.5	News/Media	100	www.manager.co.th	93.4
8	10.10.121.69	14.9	Shopping	11.1	www.weloveshopping.com	2.8
9	10.10.121.197	14.7	Search Engines/Portals	19.9	static.sanook.com	10.4
10	10.10.121.243	20.8	News/Media	26.2	www.thairath.co.th	24.4
11	10.10.121.186	54.4	Shopping	100	www.tarad.com	54.3

C. Classification

In this process step, the classification is used for predicting the cache life time and then adjusting the replacement priority to cache blocks. The cache priority is determined by the number of web requests per hour. The

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

higher the number of request, the longer the cache life it takes. The longer the cache life, the higher the priority it is given. The cache priority is assigned on a range of 1 to 10, whereas '1' is the lowest and '10' is the highest priority. To find most optimal classifier for the problems, we tried out three different algorithms: 1) c5.0 decision tree induction algorithm; 2) Chi-Square Automatic Interaction Detector (CHAID) algorithm; and 3) Neural Network technique. These classifiers are compared and chosen based on how well their predicted results are. We also define five input variables: 1) the cluster of the website; 2) the total number of requests of each site; 3) the relative frequency of request history; 4) the relationship between time and number of requests in the past; and 5) the number of requests per week.

Table II, III, and IV provide the comparisons of classification accuracy among three algorithms: C5.0, CHAID, and Neural Network technique. Results show that the C5.0 algorithm achieves an accuracy of 81.23%, 81.20%, and 81.09 % for the training set, test set, and validation set, respectively. CHAID algorithm achieves an accuracy of 73.64%, 73.56%, and 73.51% for the training set, test set, and validation set, respectively. Neural network algorithm achieves an accuracy of 73.62 %, 73.56%, and 53.52% for the training set, test set, and validation set, respectively.

Results show that C5.0 provides the most accurate model in predicting the cache life time. Thus, the algorithm is selected. To resolve the problem of more than one lowest-ranking candidate, we apply the LRU algorithm to the classifier. Typically, when the cache is full, the cache block that has lowest priority will be firstly removed. In some cases, however, there possibly are more than one lowest priority blocks. In that case, the removing order will be according to the LRU rule and the least recently used block will be firstly removed.

D. Model Evaluation

In order to evaluate the model, two traditional cache replacement algorithms -- FIFO and LRU -- were used as benchmarks to compare the results of the model. All algorithms were evaluated by comparing their hit ratio and miss ratio. The ratios are useful measures of the effectiveness and fairness of a cache replacement policy [11], [12]. The hit ratio is determined by the number of found requested data blocks in cache to the total number of requests made in a unit of time [13]. Generally, a high value of the hit ratio indicates high efficiency and overall good performance. On the contrary, miss ratio is a ratio of the number of unfound requested data blocks in cache to the total number of requests made in a unit of time and a high value of the miss ratio indicates low efficiency and overall poor performance.

III. RESULTS

Table V illustrates the comparison of the hit and miss rate among the C5.0 classifier+LRU, FIFO algorithm, and LRU algorithm. For predicting the web cache life time, the C5.0 classifier, together with LRU algorithm, achieves the highest hit rate of 63.03% and the lowest miss rate of 36.97%. Results indicate that the C5.0 classifier, when compared to FIFO and LRU, provides more efficiency and better performance.

TABLE II
THE OVERALL CLASSIFICATION ACCURACY OF C5.0 ALGORITHM.

'Partition'	1_Training		2_Test		3_Validation	
Correct	341,773	81.23%	256,474	81.20%	256,100	81.09%
Wrong	78,968	18.77%	59,366	18.80%	59,738	18.91%
Total	420,741		315,840		315,838	

TABLE III
THE OVERALL CLASSIFICATION ACCURACY OF CHAID ALGORITHM.

'Partition'	1_Training		2_Test		3_Validation	
Correct	309,822	73.64%	232,322	73.56%	232,185	73.51%
Wrong	110,919	26.36%	83,518	26.44%	83,653	26.49%
Total	420,741		315,840		315,838	

TABLE IV
THE OVERALL CLASSIFICATION ACCURACY OF NEURAL NETWORK ALGORITHM.

'Partition'	1_Training		2_Test		3_Validation	
Correct	309,734	73.62%	232,336	73.56%	232,212	73.52%
Wrong	111,007	26.38%	83,504	26.44%	83,626	26.48%
Total	420,741		315,840		315,838	

TABLE V
COMPARISON OF HIT RATIO AND MISS RATIO AMONG THREE REPLACEMENT ALGORITHMS: FIFO, LRU, AND C5.0 CLASSIFIER.

Algorithm	Hit Ratio	Miss Ratio
FIFO	37.93%	62.078%
LRU	57.52%	42.48%
C5.0 Classifier + LRU	63.03%	36.97%

IV. CONCLUSION

In this paper, we have used the integration approach of cluster analysis and classification to create a classifier model for predicting the web cache life time. The obtained classification accuracy is very good. We have also evaluated and compared the classifier model with other cache replacement algorithms. Results show that the model is more efficient and better in performance. As evidenced in our results, the model can be used to improve the Web caching performance.

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

REFERENCES

[1] I. Dzitac, "Advanced AI techniques for web mining," Proceedings of the 10th WSEAS international conference on Mathematical methods, computational techniques and intelligent systems. Cor fu Greece. 2008.
[2] D. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*. The MIT Press, 2001.
[3] R. Kosala and H. Blockeel, "Web Mining Research: A Survey," Department of Computer Science Katholieke Universiteit Leuven. Belgium, 2000.
[4] D. T. Larose, *Discovering knowledge in data: An Introduction to Data Mining*. Wiley-Interscience. NewYork, 2005.
[5] P. Jomsri, "Hit Rate Improvement in Proxy System using Data Mining Technique," M.S. thesis, Silpakorn University, 2006.
[6] P. K. Sankar, T. Varun,and M. Pabitra, "Web Mining in Soft Computing Framework: Relevance, State of the Art and Future Directions," IEEE Transactions on Neural Networks, 2002.
[7] P. Somrutai, "Web Mining Using Classification and Association Rules: a Case Study of KKU's Website," M.S. thesis, Khonkaen University, 2007.
[8] H. Srinath and S. S. Ramanna, "Web Caching: A technique to speed up access to web contents," Resonance Vol.7, No.7. , 2002.
[9] P. Bradley and U. Fayyad, "Refining Initial Points for K-Means Clustering," Proceedings of 15th International Conf. on Machine Learning, 1998.
[10] U. Fayyad, S. G. Piatetsky and P. Smyth, "Knowledge Discovery and Data Mining: Towards a Unifying Framework," The AAAI press, 1996.
[11] G. Tewari and K. Hazelwood, "Adaptive Web Proxy Caching Algorithms", Harvard University Technical Report, TR-13-04, Feb. 2004.
[12] N. Megiddo and D. S. Modha, "Outperforming LRU with an Adaptive Replacement Cache Algorithm," IEEE Computer Society, 2004, pp. 4-11.
[13] T. N. Liviana, "A Multi-Agent System for Optimization of Object Selection in Relational Database, Innovations and Advanced Techniques in Systems, Computing Sciences and Software Engineering (Springer)," University of Bridgeport, CT, USA, 2008, pp. 376-380.

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)