An Implementation of Rough Set in Optimizing Mobile Web Caching Performance

Sarina Sulaiman[1], Siti Mariyam Shamsuddin[2], Ajith Abraham[3]

[1,2] Soft Computing Research Group, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia.
[3]Centre for Quantifiable Quality of Service in Communication Systems, Norwegian University of Science and Technology, Trondheim, Norway.
sarina@utm.my[1], mariyam@utm.my[2], ajith.abraham@ieee.org[3]

Abstract

The stipulation of internet content rises dramatically in recent years. Servers have become extremely powerful and the bandwidth of end user connections and backbones grew constantly during the previous decade. Nonetheless, users frequently experience poor performance to access web sites or download files primarily if mobile devices have been used due to their limited storage, processing, display, power and communication resources. The causes are often performance which access directly on the servers (e.g. pitiable performance of server-side applications or during burst crowds) and network infrastructure (e.g. long geographical distances, network overloads, etc.). Hence, the goal of this study is to propose Rough Set (RS) as a knowledge representation for uncertainty in data of client behavior and mobile event specification with resource dependencies to reduce latency by prefetching selected resources to resolve the problems in handling dynamic web pages. We conducted the trace-based experiments on the RS approach for better classification outcomes.

1. Introduction

Caching is a technique used to store popular documents closer to the user. It uses algorithms to predict user's needs to specific documents and stores important documents. According to [1], caching can occur anywhere within a network, on the user's computer or mobile devices, at a server, or at an Internet Service Provider (ISP). Many companies employ web proxy caches to display frequently accessed pages to their employees, as such to reduce the bandwidth with lower costs [1]. Web cache performance is directly proportional to the size of the client community [2][1].The bigger the client community, the greater the possibility of cached data being requested, hence, the better the cache's performance [1].

Caching a document can also cause other problems. Most documents on the Internet change over time as they are updated. Static and Dynamic Caching are two different technologies that widely used to reduce download time and congestion[1]. *Static Caching* stores the content of a webpage which does not change. There is no need to request the same information repeatedly. This is an excellent approach to fight congestion. *Dynamic Caching* is slightly different. It determines whether the content of a page has been changed. If the contents have changed, it will store the updated version. This unfortunately can lead to congestion and thus it is possibly not a very good approach as it does require verification on the source of the data prior to updating. If these two technologies are implemented simultaneously, then the latency and congestion can be diminished.

Prefetching is an intelligent technique used to reduce perceived congestion, and to predict the subsequently page or document to be accessed [3]. For example, if a user is on a page with many links, the prefetching algorithm will predict that the user may want to view associated links within that page. The prefetcher will then appeal the predicted pages, and stores them until the actual request is employed. This approach will display the page significantly faster compared to page request without prefetching. The only drawback is that if the user does not request the pages, the prefetching algorithm will still implement the prediction of the subsequent pages, thus causes the network to be congested.

In this study, we proposed mobile Web caching scheme with an integration of Rough Set (RS). RS can help to select relevant features of client behavior and

mobile event specification to prefetch selected resources in handling dynamic web pages. The paper is structured as follows. A literature review is presented in Section 2 that describes on mobile Web caching, and RS. In Section 3, we show the experimental results and finally, Section 4 gives the concluding remark of our study.

2. Literature review

Section 2 describes related works on mobile caching, as well as a discussion on RS technique in optimizing the performance of web caching.

2.1. Related works on mobile Web caching

Caching is the most relevant techniques to improve storage system, network, and device performance. In mobile environments, caching can contribute to a greater reduction in the constraint of utilization resources such as network bandwidth, power, and allow disconnected operation [4]. A lot of studies are focus on developing a better caching algorithm to improve the choice of item to replace, and simultaneously, building up techniques to model access behavior and prefetch data. From 1990's until today, researchers on caching have produced different caching policies to optimize a specific performance and to automate policy parameter tuning. However, an adaptive and self-optimizing caching algorithms offer another advantage when considered mobile environments, where users of mobile devices should not expect to tune their devices to response the workload changes [4].

Caching is effectively for data with infrequent changes. In addition, caching data locally to mobile nodes helps the ability to retrieve data from a nearby node, rather than from a more distant base station [4]. By simply retrieving data using multiple short-range transmissions in wireless environments provides a reduction in overall energy consumed. Santhanakrishnan et al. [4] illustrated on the demand-based retrieval of the web documents in the mobile web. They proposed caching scheme; Universal Mobile Caching which performed the most basic and general form of caching algorithms and largely emphasize the impact of the adaptive policy. This scheme is suitable for managing object caches in structurally varying environments. Ari et al. [5] proposed Adaptive Caching using Multiple Experts (ACME) which individual experts were full replacement algorithms, applied to virtual caches, and their performance was estimated based on the observed performance of the virtual caches.

Wu et al. [6] introduced a rule-based modular framework for building self-adaptive applications in mobile environments. They developed techniques that combine static and dynamic analysis to uncover phase structure and data access semantics of a rule program. The semantic information is used to facilitate intelligent caching and prefetching for conserving limited bandwidth and reducing rule processing cost. Komninos and Dunlop [7] found that calendars can really provide information that can be used to prefetch useful Internet content for mobile users. However, a foreseeable problem with the current system is that the current adaptation algorithm adjusts the system gradually, and not immediately, to the needs of a user. Thus, if a dramatic change of circumstances was to occur, or if a user was to require information from a very specific and known source, it is likely the system would fail to provide the necessary information.

2.2. Mobile event specification

The events in a mobile information system can be any changes to the system itself or the way it is being used. Events can be primitive or composite [6]. A primitive event is to model a certain level of change on a single source (such as disk capacity, free memory, bandwidth, etc.). Primitive events can be combined using event operators to form composite events. Wu et al. [6] classified the events of interest in a mobile environment into resource events, mobility events, and environment events as depicted in Figure 1.

However, in this research, we considered the performance enhancement of mobile web caching based on the client behavior and the event specification including resource event for available disk space.

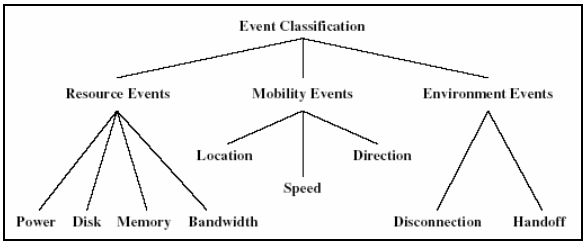

Figure 1. Event classification [6]

2.3. Performance analysis with Soft Computing technique

Artificial Intelligence researchers have explored different ways to represent uncertainty: belief networks, default reasoning, Dempster-Shafer theory,

Fuzzy set theory, Rough Set Theory (RST) [8]. The RST with the three-valued simplicity, lower, upper, and boundary approximation sets, works well on discrete and categorical data. RS can be useful even with missing data, changes of scale, and problems where membership grades are hard to define, and problems requiring changes in the partition.

The issues to be solved in this study are the learning task that will require explicit representation which deals with uncertainty. In this phase, we will look for ways to represent uncertainty in developing rules. Subsequently we will investigate how this uncertain knowledge can be exercised directly to evolutionary prefetching and learning.

2.3.1. Uncertainty. Uncertainty, as well as evolution, is a part of nature. When humans describe complex environments, they use linguistic descriptors of cognized real-world circumstances that are often not precise, but rather "Fuzzy". The theory of fuzzy sets provides an effective method of describing the behavior of a system, which is too complex to be handle with the classical precise mathematical analysis [9]. The theory of RS emerged as another mathematical approach for dealing with uncertainty that arises from inexact, noisy or incomplete information. RST focuses on the ambiguity caused by the limited distinction between objects in a given domain.

Uncertainty occurs in many real-life problems. It can cause the information used for problem solving being unavailable, incomplete, imprecise, unrealiable, contradictory, and changing [14]. In computerized system, uncertainty is frequently managed by using quantitative approaches that are computationally intensive.

Organizing uncertainty is a big challenge to knowledge-processing systems [14]. In some problems, uncertainty can possibly be neglected, though at the risk of compromising the performance of a decision support system. However, in most cases, the management of uncertainty becomes necessary because of critical system requirements or more complete rules are needed. In these cases, eliminating inconsistent or incomplete information when extracting knowledge from an information system may introduce inaccurate or even false results, especially when the available source information is limited. Ordinarily, the nature of uncertainty comes from the following three sources: inconsistent data, incomplete data and noisy data.

2.3.2. Rough Set (RS). Another approach to represent uncertainty is with RS. RS are based on equivalence relations and set approximations, and the algorithms for computing RS properties are combinatorial in nature. The main advantages of RST are as follows [14]:

- It does not need any preliminary or additional information about data;
- It is easy to handle mathematically;
- Its algorithms are relatively simple.

Wakaki et al. [11] used the combination of the RS-aided feature selection method and the support vector machine with the linear kernel in classifying Web pages into multiple categories. The proposed method gave acceptable accuracy and high dimensionality reduction without prior searching of better feature selection. Liang et al. [12] used RS and RS based inductive learning to assist students and instructors with WebCT learning. Decision rules were obtained using RS based inductive learning to give the reasons for the student failure. Consequently, RS based WebCT Learning improves the state-of-the-art of Web learning by providing virtual student/teacher feedback and making the WebCT system much more powerful. Ngo and Nguyen [13] proposed an approach to search results clustering based on tolerance RS model following the work on document clustering. The application of tolerance RS model in document clustering was proposed as a way to enrich document and cluster representation to increase clustering performance.

In general, the basic problems in data analysis that can be tackled using a RS approach are as follows [14]:

- Characterization of a set of objects in terms of attribute values;
- Finding the dependencies (total or partial) between attributes;
- Reduction of superfluous attributes (data);
- Finding the most significant attributes;
- Generation of decision rules.

Hence, our goal in this study is to find a suitable knowledge representation for mobile web caching log data.

2.3.3. A Framework of Rough Set. The framework of RS tends to classification the data. The RClass system integrates RST with an ID3-like learning algorithm [14] as shown in Figure 2. It includes three main modules: a consistency analyzer, a rough classifier and an induction engine. The consistency analyzer analyses the training data and performs two tasks; elimination of redundant data items, and identification of conflicting training data. The rough classifier has two approximators; the upper approximator and the lower approximator. The rough classifier is employed to treat inconsistent training data. The induction engine module has an ID3-like learning algorithm based on

the minimum-entropy principle. The concept of entropy is used to measure how informative an attribute is.

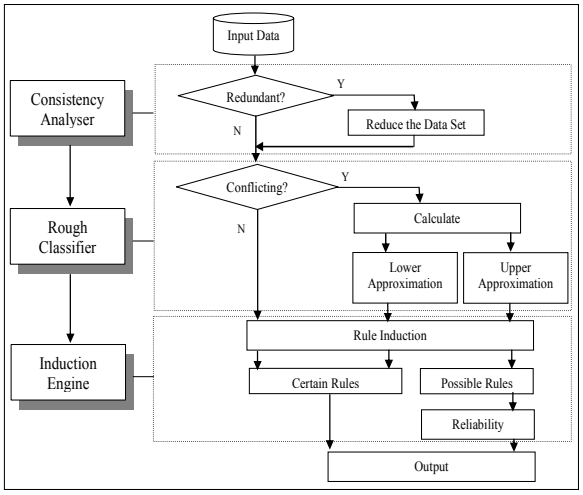

Figure 2. A framework of the RClass System [14]

3. Experimental results

Section 3 describes experimental results of dataset for HTTP requests and user behavior of a set of Mosaic clients running in the Boston University (BU), Computer Science Department [16].

3.1. BU log dataset

In this experiment, we use BU Web Trace [16] collected by Oceans Research Group at Boston University as our experiment dataset. BU traces records are collected of 9,633 files, instead of a population of 762 different users, and recording 1,143,839 requests for data transfer. Nevertheless, we only use a month (January 1995) records for 11 to 220 users, contained 33,804 records. We chose a Computer Science Department URL (http://cs-www.bu.edu) as a case for this experiment. After we cleaned the log data, 10,727 dataset is left.

Moreover, in our research we proposed a RS to reduce the rules of a log file and simultaneously enhancing the prediction performance of user behavior. RS is beneficial in probing the most significant attributes with crucial decision rules to facilitate intelligent caching and prefetching to safeguard limited bandwidth and minimize the processing cost.

The dataset is split in two, 70% (7,187 objects) for training and 30% (3,540 objects) for testing. To simplify data representation, a Naïve Discretization Algorithm (NA) is exploited and Genetic Algorithm (GA) is developed to generate the object rules. Next, Standard Voting Classifier (SVC) is selected to classify the log file dataset. The derived rules from the training are used to test the effectiveness of the unseen data. In addition, 3-Fold Cross Validation is implemented for validation of our experiment. First fold (K1) the testing data from 1 to 3540, second fold (K2) from 3541 to 7081 and third fold (K3) from 7082 to 10622. Data are stored in decision table. Columns represent *attributes*, rows represent *objects* whereas every cell contains *attribute value* for corresponding objects and attributes. A set of attributes are *URL, Machinename, Timestamp, Useridno, Sizedocument, Objectretrievaltime*, and *Cache* as a decision.

3.2. Data discretization and reduction

Training data is discretized using NA. This discretization technique is implemented a very straightforward and simple heuristic that may result in very many cuts, probably far more than are desired. In the worst case, each observed value is assigned its own interval. GA is used for reduct generation [10] as it provides more exhaustive search of the search space. Reducts generation have two options [15]; full object reduction and object related reduction. Full object reduction produces set of minimal attributes subset that defines functional dependencies, while reduct with object related produce a set of decision rules or general pattern through minimal attributes subset that discern on a per object basis. The reduct with object related is preferred due to its capability in generating reduct based on discernibility function of each object.

Table 1 illustrates the comparison results of generation of a log file dataset in different K-fold (K1, K2 and K3). The highest testing accuracy is 98.46% achieved through NA discretization method and GA with full reduct method. Number of reducts for K1, K2 and K3 are equivalent. Object related reduct, 22 and full reduct, 6. In our observation, the highest number of rules are GA with full reduct, 63311 for K1, K2 and K3 and the highest testing accuracy is GA with full reduct for K1, 98.46%.

Table 1. Comparison Reduct for K1, K2 and K3

Discretize Method	Reduct Method	K-fold	No of Reduct	No. of Rules	Testing Accuracy (%)
NA	GA (object related)	K1	22	26758	96.8644
		K2	22	26496	96.8644
		K3	22	26496	96.8079
	GA (full)	K1	6	63311	98.4618
		K2	6	63311	5.76271
		K3	6	63311	5.79096

3.3. Rules derivation

A unique feature of the RS method is its generation of rules that played an important role in predicting the output. ROSETTA tool has listed the rules and provides some statistics for the rules which are support, accuracy, coverage, stability and length. Below is the definition of the rule statistics [15]:

- The rule LHS support is defined as the number of records in the training data that fully exhibit property described by the IF condition.
- The rule RHS support is defined as the number of records in the training data that fully exhibit the property described by the THEN condition.
- The rule RHS accuracy is defined as the number of RHS support divided by the number of LHS support.
- The rule LHS coverage is the fraction of the records that satisfied the IF conditions of the rule. It is obtained by dividing the support of the rule by the total number of records in the training sample.
- The rule RHS coverage is the fraction of the training records that satisfied the THEN conditions. It is obtained by dividing the support of the rule by the number of records in the training that satisfied the THEN condition.

The rule length is defined as the number of conditional elements in the IF part. Table 2 shows the sample of most significant rules. These rules are sorted according to their support value. The highest support value is resulted as the most significant rules. From this three tables, the generated rule of {Sizedocument(0) => Cache(1)} is considered the most significant rules with the outcome of no complication (output=0) and with complication (output=1). This is supported by 3806 for LHS support and RHS support value. Subsequently, the impact of rules length on testing accuracy are evaluated based on rules set from Table 2. Consequently, the same rules are divided into two groups; 1≤ rules of length ≤2. It seems that the rules with length ≥1 contribute better classification compared to the rules with length ≤2.

3.4. Classification

According to the analysis, the result of better classification is made. Noted that, the core attributes and the significant rules can improve the accuracy of classification. Table 3 shows the result of classification performance of K1, K2 and K3 for the original table and the new decision table of log file dataset. Hence, Figure 3 depicts an overall accuracy for log file, 36.67% for all rules in original decision table and 96.85% for selected rules in new decision table. This result shows a different of overall accuracy up to 60.18% between the original decision table and new decision table.

Table 2. Sample for sorted of highest rule support values from data decision table for K1, K2 and K3

Rule	LHS Support	RHS Support	LHS Length	RHS Length
K1				
Sizedocument(0) => Cache(1)	3806	3806	1	1
Objectretrievaltime(0.000000) => Cache(1)	3805	3805	1	1
Sizedocument(2009) => Cache(0)	233	233	1	1
Sizedocument(717) => Cache(0)	128	128	1	1
K2				
URL(http://cs-www.bu.edu/lib/pics/bu-logo.gif) AND Sizedocument(0) => Cache(1)	1009	1009	2	1
URL(http://cs-www.bu.edu/lib/pics/bu-logo.gif) AND Objectretrievaltime(0.00000) => Cache(1)	1009	1009	2	1
Machinename(beaker) AND Sizedocument(0) => Cache(1)	308	308	2	1
Machinename(beaker) AND Objectretrievaltime(0.00000) => Cache(1)	308	308	2	1
K3				
URL(http://cs-www.bu.edu/lib/pics/bu-logo.gif) AND Objectretrievaltime(0.00000) => Cache(1)	989	989	2	1
URL(http://cs-www.bu.edu/lib/pics/bu-logo.gif) AND Sizedocument(0) => Cache(1)	989	989	2	1
Machinename(beaker) AND Sizedocument(0) => Cache(1)	306	306	2	1
Machinename(beaker) AND Objectretrievaltime(0.00000) => Cache(1)	306	306	2	1

Table 3. Classification performance of K1, K2 and K3 for both original decision table and new decision table of log file dataset

Decision Table	Rule Set	K-fold	Accuracy (%)	Overall Accuracy (%)
New decision table	Selected rules	K1	96.8644	96.85
		K2	96.8644	
		K3	96.8079	
Original decision table	All rules	K1	98.4618	36.67
		K2	5.76271	
		K3	5.79096	

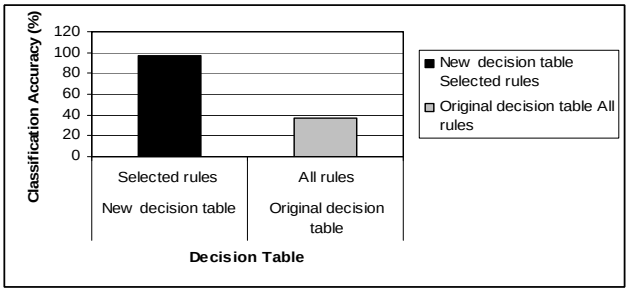

Figure 3. Overall classification accuracy for both original decision table and new decision table of log file dataset

4. Conclusion

In this paper, we present the implementation of RS technique to optimize the performance of mobile Web caching. Moreover, an empirical study has been conducted for searching optimal classification. A RS framework for log dataset is illustrated mutually with an analysis of reduced and derived rules, with entrenchment of their implicit properties for better classification outcomes.

In the future, we plan to conduct more experiments on different testing data. Furthermore, we will propose the new integration of soft computing and evolutionary computation on how to deal with multiple knowledge of mobile Web caching to contribute in reducing latency of mobile network.

5. Acknowledgment

This work is supported by MOSTI and RMC, UTM. Authors would like to thank BioCache and SCRG Research Group, FSKSM for their continuous support and devotion in making this study a success.

6. References

[1] Curran, K., and Duffy, C., "Understanding and Reducing Web Delays", *Int. J. Network Mgmt*, 15, 2005, pp. 89-102.

[2] Saiedian, M., and Naeem, M., "Understanding and Reducing Web Delays". *IEEE Computer Journal*, December 2001:34(12).

[3] Fan, L., Jacobson, Q., Cao, P., and Lin, W., Web prefetching between low-bandwidth clients and proxies: potential and performance., *Proceedings of the 1999 ACM SIGMETRICS International Conference on Measurement and Modelling of Computer Systems, Atlanta, Georgia, USA,* 1999, pp 178–187.

[4] Santhanakrishnan, G, Amer, A., and Chrysanthis, P.K., Towards Universal Mobile Caching, *Proceedings of MobiDE'05, Baltimore, Maryland, USA*, 2005, pp.73-80.

[5] Ari, I., Amer, A., Gramacy, R., Miller, E. L., Brandt, S., and Long, D. D. E., Adaptive Caching using Multiple Experts. In *Proc. of the Workshop on Distributed Data and Structures*, 2002.

[6] Wu, S., Chang, C., Ho, S., and Chao, "H., Rule-based intelligent adaptation in mobile information systems", *Expert Systems with Applications*, 2007, DOI:10.1016/j.eswa.12.014.

[7] Komninos, A. and Dunlop, M.D., "A calendar based Internet content pre-caching agent for small computing devices", *Pers Ubiquit Comput*, 2007, DOI 10.1007/s00779-007-0153-4.

[8] Russell, S. J. and Norvig, P., *Artificial Intelligence a Modern Approach*. Prentice Hall, Upper Saddle River, New Jersey, 1995.

[9] Zadeh, L., "Fuzzy sets". *Information and Control*, Vol. 8, 1965, pp. 338-353.

[10] Wróblewski, J. Finding minimal reducts using genetic algorithms. *Proceedings of Second International Joint Conference on Information Science*, 1995, pp. 186-189.

[11] Wakaki, T., Itakura, H.,Tamura, M., Motoda, H., and Washio, T., "A Study on Rough Set-Aided Feature Selection for Automatic Web-page Classification", *Web Intelligence and Agent Systems: An International Journal 4*, 2006, pp.431–441, IOS Press.

[12] Liang, A.H., Maguire, B. and Johnson, J., *Rough Set WebCT Learning*. Springer-Verlag Berlin Heidelberg, 2000, pp.425-436.

[13] Ngo, C. L. and Nguyen, H. S. *A Tolerence Rough Set Approach to Clustering Web Search Results.* Springer-Verlag Berlin Heidelberg, 2004, pp.515-517.

[14] Triantaphyllou, E. and Felici, G., *Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques*, Massive Computing Series, Springer, Heidelberg, Germany, 2006, pp.359-394.

[15] Noor Suhana. *Generation of Rough Set(RS) Significant Reducts and Rules for Cardiac Dataset Classification*. Master thesis, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia, Malaysia, 2007.

[16] BU Web Trace, http://ita.ee.lbl.gov/html/contrib/BU-Web-Client.html.