Improving the Performance of a Proxy Cache Using Expectation Maximization with Naive Bayes Classifier

P. Julian Benadit, F. Sagayaraj Francis and U. Muruganantham

Abstract The Expectation Maximization Naive Bayes classifier has been a centre of attention in the area of Web data classification. In this work, we seek to improve the operation of the traditional Web proxy cache replacement policies such as LRU and GDSF by assimilating semi supervised machine learning technique for raising the operation of the Web proxy cache. Web proxy caching is utilized to improve performance of the Proxy server. Web proxy cache reduces both network traffic and response period. In the beginning section of this paper, semi supervised learning method as an Expectation Maximization Naive Bayes classifier (EM-NB) to train from proxy log files and predict the class of web objects to be revisited or not. In the second part, an Expectation Multinomial Naïve Bayes classifier (EM-NB) is incorporated with traditional Web proxy caching policies to form novel caching approaches known as EMNB-LRU and EMNB-GDSF. These proposed EMNB-LRU and EMNB-GDSF significantly improve the performances of LRU and GDSF respectively.

Keywords Web caching · Proxy server · Cache replacement · Classification · Expectation Maximization Naive Bayes classifier

P. Julian Benadit (✉) · F. Sagayaraj Francis
Department of Computer Science and Engineering,
Pondicherry Engineering College, Pondicherry 605014, India
e-mail: benaditjulian@gmail.com

F. Sagayaraj Francis
e-mail: fsfrancis@pec.edu

U. Muruganantham
Department of Computer Science and Engineering,
Pondicherry University, Pondicherry 605014, India
e-mail: mahianandh18@gmail.com

L.C. Jain et al. (eds.), *Computational Intelligence in Data Mining - Volume 2*,
Smart Innovation, Systems and Technologies 32, DOI 10.1007/978-81-322-2208-8_33

1 Introduction

As the World Wide Web and users are maturing at a very rapid rate, the performance of web systems becomes rapidly high. Web caching is the one of the best methods for improving the performance of the proxy host. The idea behind in web caching is to maintain the most popular web log data that likely to be re-visited in the future in a cache, such that the performance of web system can be improved since most of the user requests can directly access from the cache. The main idea of web caching algorithm is the cache replacement policy. To ameliorate the execution of web caching, researchers have proposed a number of cache replacement policies Table 1. Many of these conventional replacement algorithms take into account several factors and assign a key value or priority for each web document stored in the cache. However, it is difficult to have to have an omnipotent replacement policy that performs well in all places or for all time because each replacement policy has a different structure to optimize the different resources. Moreover, several factors can influence the cache replacement policy to have a better replacement decision and it is not an easy task because one parameter is more important than the other one.

Due to this restriction, there is a need for an efficient method to intelligently handle the web cache by satisfying the objectives of web caching requirement. Thus the motivation, for incorporating the intelligent methods in the web caching algorithms. Another motivation to intelligent web caching algorithms is to train the availability of web access log files. In our previous surveys, the intelligent techniques have been applied in web caching algorithm. These studies typically build a prediction model by training the web log files. By progressing to use of the prediction model, the caching algorithms become more effective and adaptive to the web cache environment compared to the conventional web caching algorithms. However, these studies didn't take into account the user design and feature request when making the prediction model. Since the users are the origin of web access, it is necessary to establish a prediction model, whether the object can be revisited in future or not. In this paper, we use the web logs file to train by the Expectation

Table 1 Cache replacement policies

Policy	Key parameters	Eviction
LFU	No. of references	The least frequently accessed
LRU	Time since last access	The least recently accessed first
GDS	Document size Sp	Least valuable first according to value $p = Cp/Sp + L$
	Document cost Cp	
	An inflation value L	
GDSF	Document size Sp	Least valuable first according to value $p = (Cp \cdot fp/Sp)\ \beta + L$
	Document cost Cp	
	Number of non-aged references fp time since last access	
	An inflation value L	

Maximization Naïve Bayes classifier (EM-NB) [1] and to classify whether the web objects can be revisited in future or not. Based on this method, we can obtain user interest web objects that can be revisited in future or not. We then proposed the semi supervised learning mechanism EM-NB to classify the web log files and then it is incorporated with traditional caching algorithm called EMNB-LRU and EMNB-GDSF to improve its web caching performance.

The organization of this paper is as follows. In the following part, we survey the related study in web caching. In Sect. 3 we give a brief introduce of Expectation Maximization Naive Bayes classifier model. Section 4 we introduce the proposed novel web proxy caching approach and show how it integrates with the caching Algorithms. Experiment Results are described in Sect. 5. Performance Evaluation is presented in Sect. 6. Finally, we conclude our paper in Sect. 7.

2 Related Works

Web caching plays a vital role in improving web proxy server performance. The essence of web caching is so called "replacement policy", which measure the most popular of previously visited documents, retaining it in the cache those popular documents and replaces rarely used ones. The basic idea of the most common caching algorithms is to assign each document a key value computed by factors such as size, frequency and cost. Using this key value, we could rank these web documents according to corresponding key value. When a replacement is to be carried, the lower ranked web documents will be evicted from the cache. Among these key value based caching algorithms; GDSF [2] is the most successful one. It assigns a key value to each document in the cache as $K(h) = L + F(h) * C(h)/s(h)$, where L is an inflation factor to avoid cache Pollution, C(h) is the cost to fetch, F(h) is the past occurrence frequency of h and S(h) is the size of h. Accessibility of web log files that can be used as training data promotes the growth of intelligent web caching algorithms [3–5].

In preceding papers exploiting supervised learning methods to cope with the matter [3, 6–8]. Most of these recent studies use an Adaptive Neuro Fuzzy Inference System (ANFIS), Naïve Bayes (NB), Decision tree (C4.5), in worldwide caching Table 2. Though ANFIS training might consume a wide amount of time and need further process overheads. Also Naïve Bayes, result in less accuracy in classifying the large web data sets and similarly Decision Tree also result in less prediction accuracy in training large data sets. So In this paper, we attempted to increase the performance of the web cache replacement strategies by integrating semi supervised learning method of Expectation Maximization Naive Bayes classifier (EM-NB). In conclusion, we achieved a large-scale evaluation compared with other intelligent classifier like ANFIS, Naïve Bayes, and Decision Tree (C4.5) in terms of precision and recall on different log files and the proposed methodology has enhanced the performance of the web proxy cache in terms hit and byte hit ratio.

Table 2 Summary of intelligent web caching approaches

Name	Principle	Limitation
NB [3]	Constructed from the training data to estimate the probability of each class given the document feature values of a new instance	• Violation of independence assumption
		• Zero conditional probability problem
C4.5 [6]	A model based on decision trees consists of a series of simple decision rules, often presented in the form of a graph	• Not good for predicting the values of a continuous class attribute
		• Low prediction accuracy, high variance
ANFIS [7]	Neuro-fuzzy system (ANFIS) has been employed with the LRU algorithm in cache replacement	• The training process requires a long time and extra computations
		• The byte hit ratio is not good enough

3 Expectation Maximization Naive Bayes Classifier

One of the LU learning techniques uses the Expectation–Maximization (EM) algorithm [9]. EM is a popular iterative algorithm for maximum likelihood estimation problems with missing data. The EM algorithm consists of two steps, the Expectation step (or E-step), and the Maximization step (or M-step). The E-step basically fills in the missing information based on the current approximation of the parameters. The M-step, which maximizes the likelihood, re-estimates the parameters. This leads to the next iteration of the algorithm, and so on.

The ability of EM to work with missing information is exactly what is needed for learning from labelled and unlabelled examples. The web document in the labelled set (denoted by L) all have class labels (or values). The web document in the unlabeled.

Set (denoted by U) can be considered as having missing class labels. We can use EM to estimate them based on the current model, i.e., to assign probabilistic class labels to each document *di* in *U*, (i.e., Pr(*cj*|*di*)). Subsequently a number of iterations, all probabilities will converge. Notice that the EM algorithm is not really a specific "algorithm", but is a framework or strategy. It only carries a base algorithm iteratively. We will use the naïve Bayesian (NB) algorithm as the base algorithm, and run it iteratively. The parameters that EM estimates the class prior probabilities (see Eq. 1). In this paper, we use a NB classifier in each iteration of EM, (Eqs. 1 and 2) for the E-step, and (Eq. 3) for the M-step. Specifically, we first build a NB classifier *f* using the labeled examples in *L*. We then use of to classify the unlabeled examples in U, more accurately, to ascribe a probability to each class for every unlabelled example (i.e., Pr(*cj*|*di*)), which takes the value in [0, 1] instead of {0, 1}.

Let the set of classes be $C = \{c1, c2 \ldots c|C|\}$. That is, it assigns *di* the class probabilities of Pr(*c*1|*di*), Pr(*c*2|*di*), …, Pr(*c*|*C*||*di*). This is different from the example in the labelled set L, where each web document belongs to only a single class c_k (i.e. $\Pr(c_k|d_i) = 1(\text{Revisited again})$ and $\Pr(c_j|d_i) = 0(\text{Not Revisited again})\, j \neq k$.

3.1 Implementation of Expectation Maximization Naive Bayes Classification Algorithm

3.1.1 Training Phase

Input: Collection of training documents; Set of target values (categories, topics)

Output: Files containing the probabilities of $Pr(c_j)$ and $Pr(w_r|c_j)$

Algorithm EM (L, U)

1. Take an initial naïve Bayesian classifier f from only the labelled set L using Equation 1and 2

$$pr(w_t \mid c_j;\hat{\Theta}) = \frac{\lambda + \sum_{I=1}^{|D|} N_{ti} pr(c_j \mid d_i)}{\lambda \mid V \mid + \sum_{s=1}^{|V|} \sum_{I=1}^{|D|} N_{si} pr(c_j \mid d_i)} \quad (1)$$

$$pr(c_j;\hat{\Theta}) = \frac{\sum_{I=1}^{|D|} pr(c_j \mid d_i)}{\mid D \mid} \quad (2)$$

2**. Repeat // E-step**

3. For Each Example d_i in U do

4. Using the current classifier f to compute $Pr(c_j \mid d_i)$ using equation 3

$$pr(c_j \mid d_i;\hat{\Theta}) = \frac{pr(c_j \mid \hat{\Theta})\pi_{k=1}^{|di|} pr(wd_{i,k} \mid c_j;\hat{\Theta})}{\sum_{r=1}^{|c|} pr(c_j|\hat{\Theta})\pi_{k=1}^{|di|} pr(wd_{i,k}|c_j;\hat{\Theta})} \quad (3)$$

5**. End //M-step**

6. Learn a new initial naïve Bayesian classifier f from L ∪ U .

7. Until the classifier parameters stabilize return the classifier f from the last Iteration.

4 Proposed Novel Web Proxy Caching Approach

The proposed system will present a framework (Fig. 1) for novel Web proxy caching approaches based on machine learning techniques [3–5]. In Our Proposed work we use the semi supervised mining algorithm for classifying the datasets that can be revisited again or not in the future. The mining steps consist of two different phases for classifying the data sets; in the first phase (Fig. 1) we preprocess to remove the irrelevant information in the proxy log data sets, Different techniques are given at the preprocessing stage such as data cleansing, data filtering and

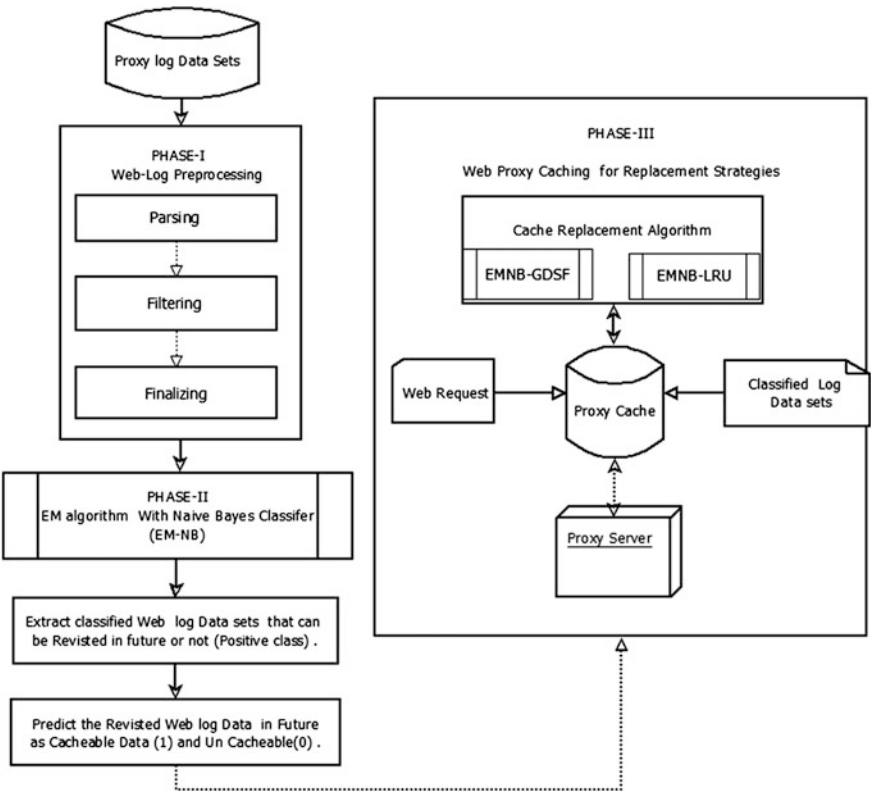

Fig. 1 Working flow of web proxy caching approach based on Expectation Maximization Naïve Bayes classifier

information consolidation. Once this task has been accomplished, the proxy data sets have been trained in the second phase (Fig. 1). By the classifier EM-NB, which predicts whether the web objects that can be revisited once more in the future or not. In the third phase (Fig. 1), the Predicted web object has been integrated to the traditional web proxy caching algorithm like LRU and GDSF for replacement strategies in order to improve the hit and byte hit ratio of the conventional replacement algorithm.

4.1 Expectation Naïve Bayes Classifier—Greedy Dual Size Frequency

The main advantage of the GDSF principle is that it executes well in terms of the hit ratio. But, the byte hit ratio of GDSF principle is too reduced. Thus, the EM-NB classifier is integrated with GDSF for advancing the performance in terms of the hit and byte hit ratio of GDSF [2]. The suggested novel proxy caching approach is called (EM-NB)—GDSF. In (EM-NB)—GDSF, a trained EM-NB classifier is used to predict the classes of web objects either objects may be re-visited later or not. After this, the classification, assessment is integrated into cache replacement policy (GDSF) to give a key value for each object in the cache buffer; the lowest values are removed first. The proposed (EM-NB)—GDSF.

4.2 Expectation Naïve Bayes Classifier—Least Recently Used

LRU policy is the most common web proxy caching scheme among all the Web proxy caching algorithms [10]. But, LRU policy suffers from cache pollution, which means that unpopular data's will remain in the cache for a long period. For reducing cache pollution in LRU, an EM-NB classifier is joint with LRU to form a novel approach Called (EM-NB)—LRU.

4.2.1 Algorithm for Greedy Dual Size Frequency—Least Recently Used

Input: Requested objects from proxy trace **Output:** Total Caching cost using LRU mechanism **Steps:** 1. Receive request for object P 2. IF P is already present in cache THEN 3. Server request internally from cache 4. Goto Step 16 5. ELSE IF P is not in present in cache 6. Serve P request from origin server 7. Increment total cost by p 8. IF P can be accommodated in cache THEN 9. Bring P into cache 10. ELSE 11. P cannot be accommodated 12.Evict least recently requested object from cache and replace it with P. 13. END IF 14. Incremental total cost by c 15. END IF 16. Repeat steps 1 to 14 for next object request of trace data. **5 Experimental Results**	**Input:** Requested objects from proxy trace **Output:** Total Caching cost using GDSF mechanism **Steps:** 1. Receive request for object P 2. IF P is already present in cache THEN 3. Serve request P internally from cache 4. ELSE IF P is not in present in cache 5. Serve request P from origin server 6. Increment total cost P; $H(p) \leftarrow L + f(p) \times c(p) / s(p)$ 7. IF P can be accommodated in cache THEN 8. Bring P into cache 9. ELSE 10. While there is not enough free cache for P 11. do $L \leftarrow \min\{H(q) \mid q$ is in cache Evict q which satisfies $H(q) = L$ and replace it with P. 12. END IF 13. $H(p) \leftarrow L + f(p) \times c(p) / s(p)$ 14. END IF 15. Repeat steps 1 to 14 for next object request of trace data.

5 Experimental Results

5.1 Proxy Log File Collection

We received data from the proxy log files of the Web object requested in some proxy Servers found in UC, BO2, SV, SD, and NY nearby the United States of the IR cache network for 15 days [11]. An access proxy log entry generally consists of

the consequent fields: timestamp, elapsed time, log tag, message protocol code, size, user identification, request approach, URL, hierarchy documents and hostname, and content type.

5.2 Web Pre-processing

Web pre-processing is a usage mining technique that involves transforming web log data into a structured format. WWW log information is frequently incomplete, inconsistent and likely to contain many errors. Web preprocessing prepares log data for further classification using machine learning classifier. It takes three different steps such as Parsing, Filtering and Finalizing. After the pre-processing, the final format of our Web log files consists of a URL-ID, Timestamp, Delay Time, and Size as presented in Table 3.

5.3 Training Phase

The training datasets are prepared the desired features of Web objects are taken out from pre-processed proxy log files. These features comprise of URL-ID, Timestamp, Delay Time, size. The sliding window of a request is that the period, a far and later once the demand were created. In additional, the sliding window ought to be about the signify time that the information usually stays during a cache (SWL is 15 min).

Once the dataset is prepared Table 4, the machine learning techniques (MNB) is applied depending on the concluded dataset to categorize the World Wide Web objects that will be re-visited or not. Each proxy dataset is then classified into training data (75 %) and testing data (25 %). Therefore, the dataset is normalized according into the series [0, 1]. When the dataset is arranged and normalized, the machine learning methods are applied using WEKA 3.7.10.

Table 3 Sample of pre-processed dataset

URL-ID	Timestamp	Frequency	Delay time (ms)	Size (bytes)	No. of future requests
1	1082348905.73	1	1,088	1,934	2
2	1082348907.41	1	448	1,803	3
4	1082349578.75	2	1,488	399	0
1	1082349661.61	2	772	1,934	0
6	1082349688.90	1	742	1,803	1
4	1082349753.72	1	708	1,233	3

Table 4 Sample of training dataset

Recency	Frequency	Size	SWL frequency	No. of future request	Target output
900	1	1,934	1	2	1 (cacheable)
900	1	1,803	1	3	1 (cacheable)
900	2	399	2	0	0 (uncacheable)
900	2	1,934	2	0	0 (uncacheable)
1,226.15	1	1,803	1	1	1 (cacheable)
1,145.08	1	1,233	1	3	1 (cacheable)
900	1	2,575	2	1	1 (cacheable)

5.4 Web Proxy Cache Simulation

The simulator WebTraff [12] can be modified to rendezvous our suggested proxy caching approaches. WebTraff simulator is used to evaluate distinct replacement Policies such as LRU, LFU, GDS, GDSF, FIFO and RAND policies. The trained, classified datasets are integrated with WebTraff to simulate the suggested novel World Wide Web proxy caching approaches. The WebTraff simulator receives the arranged log proxy document as input and develops file encompassing performance measures as outputs.

6 Performance Evaluation

6.1 Classifier Evaluation

Precision (Eq. 4) and recall (Eq. 5) are more suitable in such applications because they measure how accurate and how complete the classification is on the positive class (re-visited object). It is convenient to introduce these measures using a confusion matrix Table 5. A confusion matrix contains information about actual and predicted results given by a classifier.

Based on the confusion matrix, the precision (p) and recall (r) of the positive class are defined as follows:

$$\text{Precision (p)} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4}$$

Table 5 Confusion matrix for a two-class problem

	Predicted positive	Predicted negative
Actual positive	True positive (TP)	False negative (FN)
Actual negative	False positive (FP)	True negative (TN)

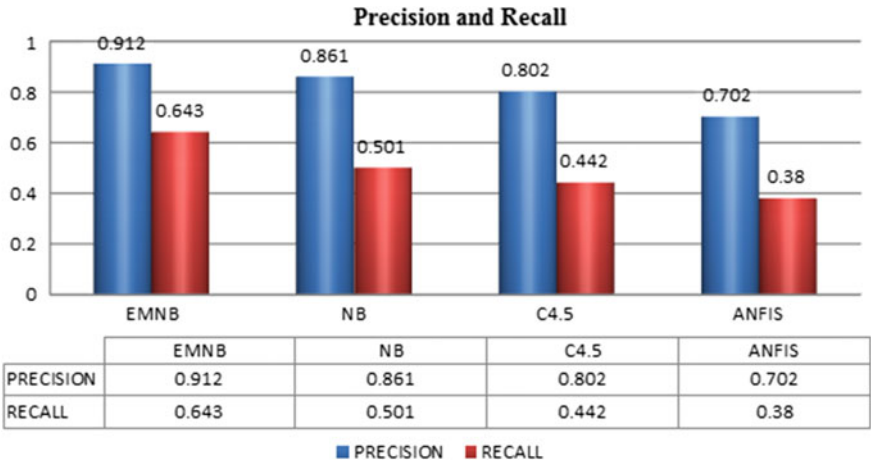

Fig. 2 Comparison of recall and precision

$$\text{Recall (r)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5}$$

In words, precision p is the number of correctly classified positive examples divided by the total number of examples that are classified as positive. Recall r is the number of correctly classified positive examples divided by the total number of actual positive examples in the test set. From the (Fig. 2) apparently displays that the EMNB accomplishes the best Precision and recall for all datasets.

In summation, the computational time for training EMNB is faster than NB and C4.5, ANFIS for all datasets Table 6. Thus, we can conclude that the applications of EMNB in web proxy caching are more valuable and effective when associated with other machine learning algorithm.

Table 6 The training time (in seconds) for different datasets

Datasets	Training time (s)			
	EMNB	NB	C4.5	ANFIS
BO2	0.11	0.12	0.36	20.39
NY	0.23	0.35	0.85	22.66
UC	0.36	0.59	1.03	18.54
SV	0.11	0.33	0.69	16.18
SD	0.55	1.32	2.90	16.92
AVG	0.272	0.542	1.232	18.93

Table 7 Examples of performance metrics used in cache replacement policies

Metric	Description	Definition
Hit ratio	Hit ratio the number of requests satisfied from the proxy cache as a percentage of the total request	$\frac{\sum_{i\in R} h_i}{\sum_{i\in R} f_i}$
Byte hit ratio	Byte hit ratio the number of byte transfer from the proxy cache as a percentage of total number of bytes for the entire request	$\frac{\sum_{i\in R} s_i \cdot h_i}{\sum_{i\in R} s_i \cdot f_i}$

Notation
s_i = size of document i
f_i = total number of requests for document i
h_i = total number of hits for document i
R = set of all accessed documents

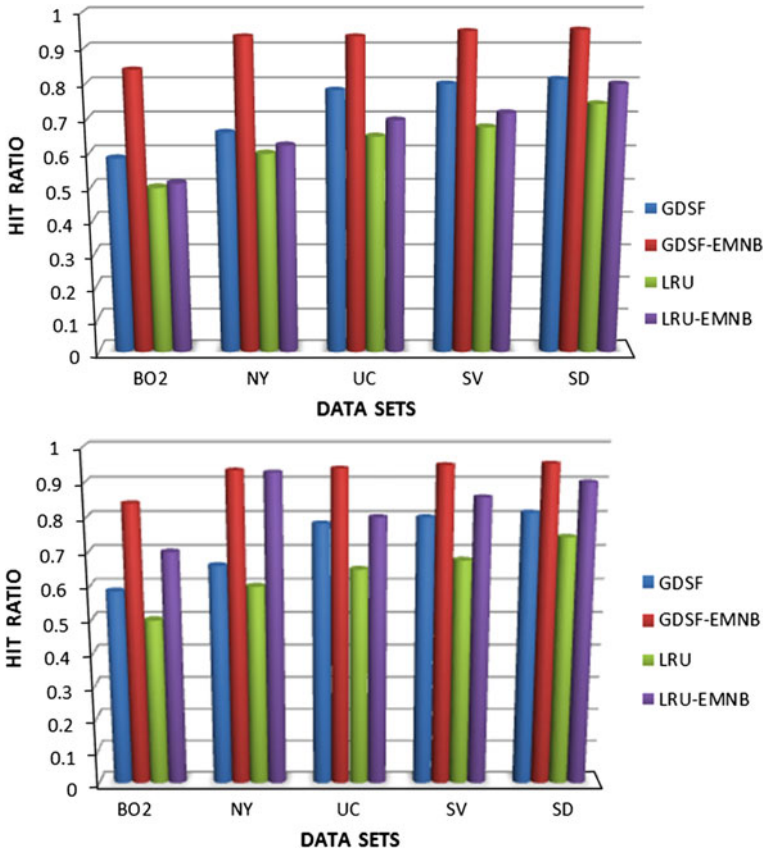

Fig. 3 Hit ratio and byte hit ratio for different dataset

6.2 Evaluation of Integrated Web Proxy Caching

6.2.1 Performance Measure

In web caching, hit ratio (HR) and byte hit ratio (BHR) Table 7 are two commonly utilized metrics for assessing the performance of web proxy caching strategies [2, 4, 9].

HR is well-defined as the ratio of the number of demands served from the proxy cache and the complete number of demands. BHR denotes to the number of bytes assisted from the cache, riven up by the complete number of byte assisted. The results in Fig. 3 Specify that EMNB-GDSF increases GDSF performance in terms of HR and EMNB-LRU over LRU is in terms of HR and in terms of BHR.

7 Conclusion

This work proposes two new web proxy caching approaches, namely EMNB-LRU, and EMNB-GDSF for improving the operation of the conventional World Wide Web proxy caching algorithms. Primarily, EMNB discovers from World Wide Web proxy log file to forecast the categories of objects to be revisited or not. Experimental results have revealed that EMNB achieves much better Precision and performance much faster than the other classifiers. In addition, in future we can consider incorporating the clustering approach to process web logs, so that a more accurate user interest model could be obtained by the EMNB and other intelligent classifiers.

References

1. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, pp. 101–103. Morgan Kaufmann, Burlington (2001)
2. Cherkasova, L.: Improving WWW Proxies Performance with Greedy-Dual-Size-Frequency Caching Policy. Technical Report HPL-98-69R1. Hewlett-Packard Laboratories, Nov 1998
3. Ali, W., Shamsuddin, S.M., Ismail, A.S.: Intelligent Naïve Bayes-based approaches for web proxy caching. Knowl. Based Syst. **31**, 162–175 (2012)
4. Romano, S., ElAarag, H.: A neural network proxy cache replacement strategy and its implementation in the squid proxy server. Neural Comput. Appl. **20**, 59–78 (2011)
5. Kumar, C., Norris, J.B.: A new approach for a proxy-level web caching mechanism. Decis. Support Syst. **46**, 52–60 (2008)
6. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, Burlington (1993)
7. Ali Ahmed, W., Shamsuddin, S.M.: Neuro-fuzzy system in partitioned client side web cache. Expert Syst. Appl. **38**, 14715–14725 (2011)
8. Chen, H.T.: Pre-fetching and re-fetching in web caching system. Algorithms and Simulation, Master thesis, Trent University, Peterborough, Ontario (2008)
9. Liu, B.: Web Data Mining: Exploiting Hyperlinks, Contents, and Usage Data, pp. 173–176. Springer, Berlin (2007)

10. Podlipnig, S., Boszormenyi, L.: A survey of web cache replacement strategies. ACM Comput. Surv. **35**, 374–398 (2003)
11. NLANR.: National Lab of Applied Network Research (NLANR), and Sanitized Access Logs. Available at http://www.ircache.net/2010
12. Markatchev, N., Williamson, C.: WebTraff: a GUI for web proxy cache workload modeling and analysis. In: Proceedings of the 10th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, pp. 356–363. IEEE Computer Society (2002)
13. Kin-Yeung, W.: Web cache replacement policies a pragmatic approach. IEEE Netw. **20**, 28–34 (2006)