Diminish Latency Issues and Congestion using ZIPF Estimator

Ankita Khanna¹, Komal Arora²
¹M.Tech student, CSE Department, SPGOI, Rohtak
²Asstt. Professor, CSE Department, SPGOI, Rohtak
¹ankita.khanna01@gmail.com, ²grewalkomal14@gmail.com

Abstract
The continued growth of Wide Web has resulted in congestion over the network, servers overload and latency issues. Some techniques have been emerged like Caching which is used in the Web environment to reduce time latency, as it allows fast retrieval of frequently access documents but due to some factors like increasing number of users, irregular updation, scarce system resources of cache servers reduced the ideal effectiveness of Web caching. Other Technique is Prefetching where web pages that the user is likely to access in the future are transferred to her cache without being requested. This paper implements the data mining techniques to determine the next page access. For making the correct prefetching decision we have combined the data mining techniques with zipf estimator and then applied it to determine the probability of page being pre-fetched.

Keywords: Caching, Prefetching, Web Mining Techniques & Zipf Estimator.

1. Introduction
The huge adoption of internet has changed in every way of our life from communication to conducting businesses. There is no doubt that the internet is growing in terms of users and what it can achieve. The number of hours that people spend on the internet is increasing with each day. This is the trend for people accessing the internet at home or “anywhere else”. In this modern world everything is getting electronic from studying to jobs recruitment, from shopping to payments & many more. Technology has altered the way, man lives his life. Everybody is getting more and more dependent on internet without stepping out & doing all his work through internet from home only. There are numerous small elements contained within the Web, but bits of large ones only. Some websites contains lakhs of pages, but lakhs of websites only contain a handful of pages. Some sites contain millions of links, but numbers of sites have just two or three. Billions of web users flock to few selected sites, giving bit attention to billions of others.

Various data mining methods are used to discover the hidden information in the Web. One of the main directions of research in the Web is to reduce the congestions over network, time latencies users experience when navigating through Websites.

Web caching [3] is an effective technique to diminish the network traffic, thereby minimizing the user access latency as it efficiently transfers copies of popular documents from Web servers closer to the Web clients. In general, Web clients see shorter delays when requesting a webpage, network manager’s view less traffic and Web servers see lower request rates. But, as per recent studies there are limited benefits from this technique. Web Caching performs poor because of some main factors like the vast number of documents available in the Web, the quick rate of their change and the diverse needs across users and overtime.

Web pre-fetching [8] has proved to be an attractive solution to the problem of latency. Pre-fetching reduces user access time, but at the same time, ore bandwidth and increased traffic it required. When it is employed, Web pages that the user is likely to access in the close future are transferred to her cache without being requested. If any of the pre-fetched pages
are requested, it will already be in the local cache, thus reducing the latency to minimum. Pre-fetching is complementary to the caching mechanism.

The organization of the paper is as follows: In section II, Data Mining Techniques are discussed. In section III, we discuss Zipf Estimator, followed by section IV, Implementation workflow is discussed & related flowcharts to analyse the rank and frequency of the next page to be access. In section V, we draw a conclusion and address the future work.

2. DATA MINING TECHNIQUES

2.1 Clustering

Clustering is a technique to group together a set of items having similar characteristics. In the Web Usage domain, there are two kinds of interesting clusters to be discovered: usage clusters and page clusters. Clustering of users tends to establish groups of users exhibiting similar browsing patterns. Such knowledge is especially useful for inferring user demographics in order to perform market segmentation in E-commerce applications or provide personalized Web content to the users. On the other hand, clustering of pages will discover groups of pages having related content. This information is useful for Internet search engines and Web assistance providers. In both applications, permanent or dynamic HTML pages can be created that suggest related hyperlinks to the user according to the user's query or past history of information needs.

2.2 Markov Model

Markov models are used in the identification of the next page to be accessed by the Web site user based on the sequence of previously accessed pages. After dividing user sessions into a number of clusters, Markov model analyses are carried out on each of the clusters.

S. Jespersen [5] said that Markov assumptions are used as the basis to mine the structure of browsing patterns. Markov-based structures for web usage mining are best suited for tasks demanding less accuracy such as pre-fetching, personalization, and targeted ads. Many of the papers proposed using association rules or Markov models for next page prediction. Faten Khalil [7] proposes an improved approach, based on a combination of Markov models and association rules that result in better prediction accuracy and more coverage.

2.3 Association Rule

M. Eirinaki [1] use Association rules are used in order to discover the pages which are visited together even if they are not directly connected, which can reveal associations between groups of users with specific interest. This information can be used for example for restructuring Web sites by adding links between those pages which are visited together. Using this knowledge the trends of the activity of the users can be determined and predictions to the next visited pages can be calculated.

Let I= \{I_1, I_2, \ldots, I_m\} be a set of m distinct attributes, T be transaction that contains a set of items such that T \subseteq I, D be a database with different transaction records Ts. An association rule is an implication in the form of \text{X}=>\text{Y}, where \text{X, Y} \subseteq I are sets of items called itemsets, and \text{X} \cap \text{Y} =\emptyset. \text{X} is called antecedent while \text{Y} is called consequent, the rule means \text{X} implies \text{Y}.

There are two important basic measures for association rules, support(s) and confidence (c). Since the database is large and users concern about only those frequently purchased items, usually thresholds of support and confidence are predefined by users to drop those rules that are not so interesting or useful. Support(s) of an association rule is defined as the percentage/fraction of records that contain \text{X} \cup \text{Y} to the total number of records in the database.
Suppose the support of an item is 0.1%, it means only 0.1 percent of the transactions contain purchasing of this item.

Confidence of an association rule is defined as the percentage/fraction of the number of transactions that contain $X \cup Y$ to the total number of records that contain $X$. Confidence is a measure of strength of the association rules, suppose the confidence of the association rule $X => Y$ is 80%, it means that 80% of the transactions that contain $X$ also contains $Y$ together.

3. ZIPF Estimator

Zipf estimator is based on Zipf law [10]. Zipf’s Law states that frequency of terms in a set of text collection follows a power law distribution. By the Zipf estimator the probability of accessing the next page can be computed efficiently.

"Zipf's law" has been given to the following approximation of the rank-frequency relationship:

$$rf = c$$

(1)

Here $r$ is the rank of a word-type, $f$ is the frequency of occurrence of the word-type, and $c$ is a constant, dependent on the corpus (often around one-tenth of the total size of [i.e., number of word-tokens in] the corpus).

When stated algebraically, Zipf's law is given in the form of equation (1), but the law is probably most familiar in the graphic representation of a mathematically equivalent form:

$$\log r + \log f = \log c$$

(2)

More generally, analytic geometry tells us that the equation of an arbitrary line whose slope is $-B$ can be written as:

$$B\log r + \log f = \log c$$

(3)

One such line is pictured by the solid line, which has a slope of -0.92. (The relationship of this line to the data points will be discussed later.) If we write equation (3) in a form like that of equation (1), we have:

$$r^B f = c$$

(4)

Note that if $B$ takes on the particular value 1, then equation (4) becomes identical with equation (1). Thus, equation (4) is a generalization of Zipf’s law, and we shall refer to it as the "generalized Zipf’s law."

It should be noted that Zipf’s law only approximates the relationship between rank $r$ and frequency $f$ for any actual corpus. Zipf’s work suggests that the approximation is much better for the middle ranks than for the very lowest and the very highest ranks, and his work with samples of various sizes suggests that the corpus should consist of at least 5000 words in order for the product $rf$ to be reasonably constant, even in the middle ranks.
Rule generator is divided into two phases.

**Phase I:** Rule Formulator: After preprocessing of web log, cleaner version of proxy log is formed called Data mart. Data is extracted from the data mart and data mining operations are applied in this phase. For better accuracy, integration of clustering, k order Markov model and association rule is done.

**Phase II:** Rule Selector: After data mining techniques are applied patterns or rules are formulated, which are extracted by Rule Selector phase. In this Zipf law is applied. As per the law, ranking is done based on the frequency count and then probability is calculated based on the known ranks. Thus this phase is further divided into Rank Analysis and Probability Calculator. For calculating the probability Zipf Estimator is implemented.

The Implemented architecture is shown in Figure 5.2. Primary focus in this paper is laid on implementation of Rule Generator phase. Rule generator extracts information from data mart.

**Figure: Architecture showing the working of PPE**

### 4. Implementation Workflow

An architecture developed in the implementation stage suggested the prefetching the web pages from WWW on the proxy server using data mining techniques & Zipf estimator can make the pronounceable change in the predicting and prefetching the web pages from the proxy server.

Furthermore it covers:

- Demonstrate the use of Zipf estimator to calculate the probability of web page that is to be prefetched by the proxy server.
- To evaluate the effectiveness of the Zipf estimator.
Flowchart for rule formulator:

1. Start
2. Take the Clustered set
3. Calculate the frequency of accessed pages
4. Set the pass frequency
5. Get the user session after pruning
6. Apply the K-order markov model
7. For each state of markov model
8. If production is ambiguous?
9. Collect all the session satisfying the state
10. Construct the association rule to resolve the ambiguity
11. Store the association rules with states
12. Stop
Flow chart for rule selector:-

5. Conclusion

This paper aims on reducing the congestion over the network & increasing access latencies with increase in the number of users on network putting emphasis on data mining techniques & Zipf estimator so that the page can be prefetched in the cache before it explicitly required thereby reducing traffic over the network. Here, web is viewed as a system for the prediction and prefetching the useful information which is helpful for the user. Depending upon the probability of the next page, the page can be prefetched locally on the proxy server. When the user request for that page the page is given directly to the user rather than going to the web server. For making the correct prefetching decision in this paper we have combined the approach of some data mining techniques such as clustering, markov and association rule mining and then applied the zipf estimator to determine the probability of page being prefetched.

References


