Classification Pruning for Web-request Prediction

Ian TianYi Li
School of Computing Science
Simon Fraser University
Burnaby, BC Canada V5A 1S6
tlie@cs.sfu.ca

Qiang Yang
School of Computing Science
Simon Fraser University
Burnaby, BC Canada V5A 1S6
qyang@cs.sfu.ca

Ke Wang
School of Computing Science
Simon Fraser University
Burnaby, BC Canada V5A 1S6
wangk@cs.sfu.ca

ABSTRACT

N-gram and repeating pattern based prediction rules have been used for next-web request prediction. However, there is no rigorous study of how to select the best rule for a given observation. The longer patter may not be the best pattern, because such patters are also more rare. In this paper, we propose several rule-pruning methods that enable us to build efficient, compact and high-quality classifiers for web-request prediction.

Keywords

Web-log Mining. Web Request Prediction

1. Introduction

In this work, we focus on web-log based prediction of future HTTP requests. Predictions on near-future user behavior can be very useful for a number of purposes. Much work has been done on recommendation systems that rely on prediction models to make inferences on users' interests. Many researchers have studied how to use of web-log based prediction for pre-sending or pre-fetching of web objects in anticipation of users potential requests. An important issue is to use n-grams or frequent sequential patterns for future web request prediction. The work in this area includes [2, 3, 4, 5, 6]. Once a prediction system is built, one can pre-send documents for clients. Albrecht et al. [1] used a Markov model to make predictions in order to send documents ahead of time. These predictions are shown to reduce the access latency for web users.

In this work, we view n-gram or frequent subsequence based algorithms for web-request prediction as the task of classification. In this view, the existing work in web-request prediction suffers from the fact that there is no clear quality metric on what constitutes a good prediction rule. Previous work have used the minimum support notion for this purpose, where the minimum support of an n-gram prediction rule is a lower bound threshold on how many cases the rule has to cover in the training data set. For example, in [6] the minimum support, where support of rule LHS → RHS is the probability that the pattern {LHS, RHS} holds in the training data, is set to be five occurrences of the pattern. However, this usage is ad-hoc. For example, in the event that two different n-grams for different n apply to a same testing case, which n-gram rule should we trust more? Traditional wisdom is to pick the longer rule. However, longer rules are supported by fewer cases in training data, and are thus less *confident* in the statistical sense. As we will see, they do not always give the best result.

We propose to use a rigorous statistical metric to measure the quality of the rules, using the potential error rate derived from both the support and confidence (percent of correct predictions) factors. The result is a "pruning method" for classifiers, which gives significant improvements over previous methods.

In this paper, in order to unify the notion of n-gram and the notion of longest repeating patterns, we use a single term "prediction rule" to denote the rules under consideration for both the n-gram rules and the longest repeating subseuences. Further, we study all methods by always using a default rule, a rule which has empty LHS. This default rule is the most popular object in the training log. When no matching is possible, we can always use it for prediction.

2. Different Rule Pruning Strategies

2.1 Longest Match

The longest-match method chooses the rule with the longest left-hand-side (LHS) that matches a case. The rationale of Longest Match method is based on the conclusion that longer surfing path will contain more accurate and richer information about the user access pattern than the shorter ones [3]. A problem with this method is that it always prefers to rules with a longer LHS, regardless of how many cases the rule covers. We know that although higher-order n-grams will be more specific and accurate, their support decreases exponentially with n, essentially making them less confident.

2.2 Most confident Selection

With the Most Confident Selection, we always choose a rule with the highest confidence among all the applicable association rules. If we have a tie, we choose the longer rule. The confidence of a rule LHS→RHS is the conditional probability that RHS holds given LHS. The rationale of the Most Confidence Selection is based on the assumption that the testing data will share the same characteristics as the training data, which we built our classifier on. So if a rule has a higher confidence in the training data, then this rule will also show a higher confidence in the testing data, which means the class predicted by that rule will be most likely to occur next.

2.3 Pessimistic Selection

A problem with the previous methods is that in most real cases, training data will not reflect exactly the same aspects of the testing data. Therefore, they are prone to generating overfitting rules. To deal with over-specific rules, which are the longest-n-grams, we choose to use statistics pessimistic error estimates -- a powerful tool in statistics.

Given a training log file, we denote the number of correctly classified cases as C, the number of incorrectly classified cases as E, and the total number of cases classified by the rule as N. Then the confidence of the rule is C / N = C / (C + E) = 1 – E / N. The pessimistic confidence of the rule is:

Pessimistic_confidence = $1 - E_p / N = 1 - U_{CF} (E, N) / N$

Where E_p is the Pessimistic Estimated Error Rate using the formula $E_p = U_{CF} (E, N) / N$, and U_{CF} (E, N) is the Pessimistic Estimated Error.

For a given observation, the *pessimistic selection* method picks up the rule with the highest pessimistic confidence in all the applicable rules, regardless of the length of the LHS of a rule. We will see later that this method gives better result than the longest match method.

2.4 Last-Substring Index Tree (LSIT)

Finally, we propose a method to *compile* the pessimistic selection pruning method into a tree highly compact structure of the rules, enabling efficient use of CPU time and memory during run time

We introduce a tree-like structure, called the 'Last-Substring Index Tree' (LSIT), to store all the rules in the prediction model as nodes in the tree, and store the relative pessimistic confidence in the relative positions of the nodes.

We say that a rule LHS1→RHS1 is a parent of LHS2→RHS2, if LHS1 is a trailing substring of LHS2. We build the LSIT tree according to this parent child relation. Furthermore, we require that the children of all rules in this tree to strictly more confident than their parents. This allows for dramatic pruning of the trees, resulting a smaller but more efficient tree. When applying the LSIT tree to a given observed case (a sequence of objects), we trace the tree top-down, reaching a deepest node where the rules apply. The deepest rule will be selected.

Experiments have been done on a NASA data set to compare all the pruning methods as well as the individual n-gram methods with or without minimum support and minimum confidence. . The NASA data set contains one month worth of all HTTP requests to the NASA Kennedy Space Center WWW server in Florida. The performance is measured against the precision of the classifiers obtained. In the NASA data, we used the first 100,000 requests as training data set and the next 25,000 requests as the testing data set. In the figure, the most-confidence selection and the pessimistic selection methods under each value of n are the results of rule set with LHS equal or smaller than n. As can be seen, the Pessimistic Selection pruning method, represented by the top-curve in the figure, together with the LSIT classification tree, gives the best overall result.

We have also performed experiments using a number of other web logs, including one from EPA. Our results from these other logs also confirm the superiority of the pessimistic pruning method.

3. Conclusions

In this paper, we have presented an effective method for pruning n-gram rules and build a compact tree structure for web request prediction. We have shown that using the longest repeating subsequences algorithm and the n-gram based algorithms may not always give the best result. Our pessimistic rule pruning methods and the associated LSIT compression method predict with the highest accuracy.

4. Acknowledgement

We would like to thank Canadian Natural Science and Engineering Council (NSERC) and the IRIS for their support.

5. REFERENCES

[1] Albrecht, D. W., Zukerman, I., and Nicholson, A. E. 1999. **Pre-sending documents on the WWW: A comparative study**. *IJCAI99 – Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence.*

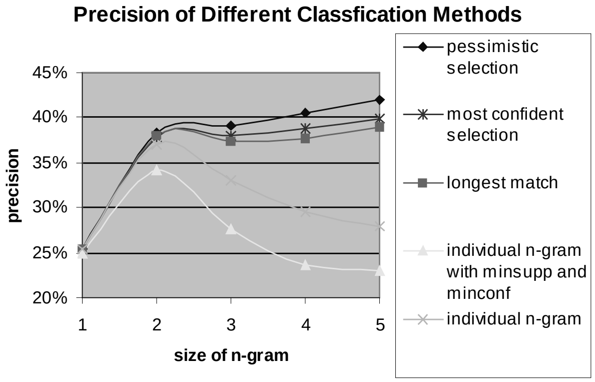
[2] T. M. Kroeger and D. D. E. Long. **Predicting future file-system actions from prior events**. In *USENIX 96*, San Diego, Calif., Jan. 1996

[3] Pitkow J. and Pirolli P. **Mining longest repeating subsequences to predict www surfing**. In *Proceedings of the 1999 USENIX Annual Technical Conference*, 1999.

[4] Schechter, S., Krishnan, M., and Smith, M.D. 1998, **Using path profiles to predict HTTP requests.** *Proceedings of the Seventh International World Wide Web Conference Brisbane, Australia.*

[5] Silverstein, C., Henzinger, M., Marais, H., and Moricz, M. 1998. **Analysis of a very large AltaVista query log**. *Technical Report 1998-014*, Digital Systems Research center, Palo Alto, CA.

[6] Z. Su, Q. Yang, and H. Zhang. **A prediction system for multimedia pre-fetching on the Internet.** In *Proceedings of the ACM Multimedia Conference 2000*. ACM, October 2000.


Precision of Different Classfication Methods
precision
size of n-gram
pessimistic selection
most confident selection
longest match
individual n-gram with minsupp and minconf
individual n-gram