Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August 2005

PARALLEL WEB MINING FOR LINK PREDICTION IN CLUSTER SERVER

CAI-RONG YAN[1], JUN-YI SHEN[1], QIN-KE PENG[2], DING PAN[1]

[1]Software Institute, Xi'an Jiaotong University, Xi'an 710049, Shanxi, China
[2]Systems Engineering Institute, Xi'an Jiaotong University, Xi'an 710049, Shanxi, China
E-MAIL: cryan@mail.xjtu.edu.cn

Abstract:

Many Web mining methods have recently been used to model user navigational behavior based on log files of the Web server. Cluster-based server architectures combine good performance and low cost, and are widely used for Web service. In this paper, we propose a parallel Web mining (PWM) algorithm for link prediction in the environment of Web cluster server consisting of several nodes that act as independent Web servers. According to the PWM algorithm, the transition probability matrixes are firstly obtained from the Web log files of each node by adopting the Markov chain model, compressed under the constraint of the probability threshold ε and parallel threshold α, and then sent to the central node which will combine these independent results to get an integrated result by some rules. By different accuracy requirement, the PWM algorithm can be divided into simple PWM algorithm (S-PWM), faster but less accurate, and complex PWM algorithm (C-PWM), slower but more accurate. Furthermore, a related incremental parallel Web mining (I-PWM) algorithm is put forward too. The experimental results show that PWM algorithm can not only alleviate the communication cost by sending the mined transition probability matrix and decrease the time complexity by disposing in parallel but also hardly affect the accuracy of the Web mining result.

Keywords:

Parallel Web mining; link prediction; cluster server; Markov chain model

1. Introduction

The increasing demand of Web service can not be matched with the increase in the server capability and network speed. Therefore, many alternative solutions, such as cluster-based Web servers, P2P technologies and Grid computing have been developed to reduce the response time observed by Web users. Accordingly, mining distributed Web data is becoming recognized as a fundamental scientific challenge. Data mining algorithms working on very large data sets take very long times on conventional computers to get results. One approach to reduce response time is sampling. But, in some case reducing data might result in inaccurate models. The other approach is parallel computing. High performance computers and parallel data mining algorithms can offer a very efficient way to mine very large data sets by analyzing them in parallel. In this area parallel computers are used not only for Quantitative Computing but also for Qualitative Computing.

Many parallel algorithms have been developed in recent years in the area of Web log mining, such as parallel clustering algorithm using maximal large item sets [1] and parallel association rules mining [2]. In addition, there have been proposed some parallel mining models [3,4,5,6], for example, Papyrus is a parallel data mining system developed for clusters and super-clusters of workstations, Kensington Enterprise data mining is a Parallel and Distributed Knowledge Discovery (PDKD) system based on a three-tier client/server architecture. As far as we know, there is not a parallel Web mining model for link prediction that is very important to prefetch pages in cluster server.

A parallel mining algorithm for link prediction based on the parallel data mining model is proposed in this paper. It consists of two steps: first mining data in the local servers directly using Markov chain model and then combining the mined results to get an integrated result. Such parallel algorithm exchanges less data so as to reduce the communication cost of network and mines data in parallel to avoid large-scale computing.

2. Web mining model

2.1. Distributed data mining

Figure 1 shows two architectures for distributed data mining: data mining based on warehouse and parallel data mining [7]. The former first collects data from different servers into a data warehouse and then begins to mine information using the conventional classification, clustering and some other methods. The latter mines data in the local

servers directly and then combines the mined results to get an integrated result. Many researches show that the latter will spend less communication cost to exchange data and less time to mine large-scale data [7].

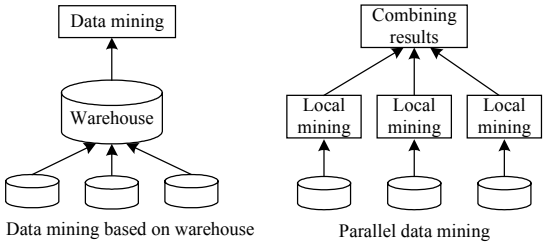

Figure 1. Two architectures for distributed data mining

There are three main strategies in the exploitation of parallelism in data mining algorithm, independent parallelism, control parallelism and SPMD parallelism [5]. To independent parallelism, processes are executed in parallel in an independent way; generally each process has access to the whole data set. To control parallelism (or task parallelism), each process executes different operations on (a different partition of) a data set. To SPMD parallelism, a set of processes execute in parallel the same algorithm on different partitions of a data set and processes exchange partial results.

In typical cluster environment, all files are distributed to every node of cluster server. Any request, according to some scheduling algorithm, such as round-robin and least-connection, can be forwarded to any node. Each node, as an independent Web server, records the access log that can be used as the source of data mining. That is, Web log is distributed in different servers and each server has its own processor and Memory. So SPMD parallelism is a suitable strategy to mine information in cluster server.

2.2. Link prediction model

Markov chain model is widely used in the area of link prediction of Web mining [6,8]. A discrete Markov chain model can be defined by the triple <S,A,λ>. S corresponds to the state space, A is a matrix representing transition probabilities from one state to another and λ is the initial probability distribution of the states in S. The fundamental property of Markov model is the dependency on the previous state. If the vector S(t) denotes the probability vector for all the states at time 't', then:

$$S(t) = S(t-1)A \tag{1}$$

Markov chains can be applied to Web link sequence modeling. In this formulation, a Markov state can correspond to the URL of a request. The matrix A can be estimated using many methods. Without loss of generality, the maximum likelihood principle is applied in this paper to estimate A and λ. Each element of the matrix A(s,s') can be estimated as follows:

$$A(s,s') = \frac{C(s,s')}{\sum_{s''} C(s,s'')}, \qquad \lambda(s) = \frac{C(s)}{\sum_{s'} C(s')} \tag{2}$$

C(s,s') is the count of the number of times s' follows s in the training data. Sometimes, a state is seldom to be related with other states so that there may exist some values that are so small and can be ignored. We can set a threshold ε and let the probability less than ε be zero so as to compress a transition probability matrix to be a small and high efficient matrix.

3. Parallel Web mining

3.1. PWM algorithm

In the environment of cluster server, there are several Web servers that provide service synchronously. Now, parallel Web mining algorithm based on Markov chain model (simply called PWM) are proposed to do with link prediction.

Suppose the following parameters:

n: the number of cluster nodes, S_i represents any one node ($1 \le i \le n$).

p_{ij}: the transition probability from state i to state j in the cluster ($1 \le i,j \le |S|$, where |S| refers to the number of the states).

p_{ijk}: the transition probability from state i to state j in node S_k ($1 \le k \le n$).

D_{ik}: the number of transition from state i to any state in node S_k.

D_i: the number of transition from state i to any state in the cluster.

ε: the threshold of probability.

ε.a: the threshold of probability in any one node ($0 \le a \le 1$).

Theorem 1. $\sum_{k=1}^{n} D_{ik} \cdot p_{ijk} = \sum_{k=1}^{n} D_{ik} \cdot p_{ij}$

Proof. Since, the transition number before and after integration should be unchanging.

That is, the sum of the products of D_{ik} and p_{ijk} in node S_k should be equal to the product of the sum of D_{ik} in node S_k and p_{ij}.

Theorem 2. If the transition probability p_{ij} is not less than ε ($p_{ij} \ge \varepsilon$), then there necessarily exists a node S_k and its p_{ijk} is not less than ε ($p_{ijk} \ge \varepsilon$).

Proof. Assume that for any node S_k, $p_{ijk} < \varepsilon$.

Then, $\sum_{k=1}^{n} D_{ik} \cdot p_{ijk} < \sum_{k=1}^{n} D_{ik} \cdot \varepsilon \le \sum_{k=1}^{n} D_{ik} \cdot p_{ij}$.

According to Theorem 1,

$\sum_{k=1}^{n} D_{ik} \cdot p_{ijk} = \sum_{k=1}^{n} D_{ik} \cdot p_{ij}$.

Therefore, the assumption does not accord with the conclusion.

That is, the assumption is wrong and the conclusion is right.

According to Theorem 2, we can get the conclusion: if all $p_{ijk} < \varepsilon$, then $p_{ij} < \varepsilon$.

Theorem 3. If $p_{ij} \ge \varepsilon$ and exists node $S_1,\ldots,S_k,\ldots,S_t(1\le k\le t, 1\le t\le n)$, $p_{ijk} \le \varepsilon.\alpha$, then the integrated probability of the left nodes p_{ij}' at least is p_{ij}*.

$$p_{ij}' = \frac{\sum_{k=t+1}^{n} D_{ik} \cdot p_{ijk}}{\sum_{k=t+1}^{n} D_{ik}},$$

$$p_{ij}^* = \frac{\sum_{k=1}^{n} D_{ik} \cdot \varepsilon - \sum_{k=1}^{t} D_{ik} \cdot \varepsilon \cdot \alpha}{\sum_{k=t+1}^{n} D_{ik}}$$

Proof. According to Theorem 1, we can get:

$\sum_{k=1}^{n} D_{ik} \cdot p_{ij} = \sum_{k=1}^{n} D_{ik} \cdot p_{ijk} \ge \sum_{k=1}^{n} D_{ik} \cdot \varepsilon$

That is, $\sum_{k=1}^{t} D_{ik} \cdot p_{ijk} + \sum_{k=t+1}^{n} D_{ik} \cdot p_{ijk} \ge \sum_{k=1}^{n} D_{ik} \cdot \varepsilon$.

That is, $\sum_{k=1}^{t} D_{ik} \cdot \varepsilon \cdot \alpha + \sum_{k=t+1}^{n} D_{ik} \cdot p_{ijk} \ge \sum_{k=1}^{n} D_{ik} \cdot \varepsilon$.

Then $p_{ij}' \ge p_{ij}^* = \frac{\sum_{k=1}^{n} D_{ik} \cdot \varepsilon - \sum_{k=1}^{t} D_{ik} \cdot \varepsilon \cdot \alpha}{\sum_{k=t+1}^{n} D_{ik}}$.

According to the above theorems, we can get the following rules.

Rule 1. If p_{ijk} for all nodes is less than ε.a, according to Theorem 2, p_{ij}=0.

Rule 2. If p_{ijk} for nodes $S_1,\ldots,S_t(1\le k\le t, 1\le t\le n)$ is less than ε.a and $p_{ij}' < p_{ij}^*$, according to theorem 3, p_{ij}=0.

Rule 3. If p_{ijk} for nodes $S_1,\ldots,S_t$ is less than ε.a and $p_{ij}' \ge p_{ij}^*$, then p_{ij} can be expressed by p_{ij}' approximatively. If PWM algorithm requires a more accurate result, the real probability p_{ijk} $(1\le k\le t,\ 1\le t\le n)$ (not be compressed), should be sent to the central node.

Rule 4. If p_{ijk} for all nodes is not less than ε.a, according to Theorem 1, we can get p_{ij}.

The following is the description of PWM algorithm.

Algorithm: parallel Web mining for link prediction in cluster server (PWM).

Input: Web log files of the cluster nodes, parameter ε and α, the number of nodes n.

Output: transition probability matrix M.

Step1: According to Markov chain model described above, we can get transition probability matrix of each node M_k and the array A_k which records the transition number of each state to other states.

Step2: According to parameter ε and α, compress M_k to M'_k whose elements are not less than ε.α.

Step3: Send M'_k and A_k to the central node.

Step4: According to the above rules, we can get the matrix M. If the result needs to be more accurate, go to Step 5; else the process is over.

Step5: According to Rule3, if $p_{ij}' \ge p_{ij}^*$, the central node will request the real probability from the nodes whose $p_{ijk} < \varepsilon.a$.

Assume the number of states is m, then the maximum number of the transition probability larger than zero will be m*m. So the communication complexity is O(m*m*n). The time complexity of Web mining in single node depends on the Markov chain model and the detailed information can be got in the reference [8]. The time complexity of combining the mining result is O(m*m*n). If the algorithm only includes Step1 to Step4, then the smaller α is, the more precise the result will be. We will call the PWM algorithm not including step 5 simple PWM (S-PWM) and the PWM including step5 complex PWM (C-PWM).

3.2. Incremental PWM algorithm

In order to update the existing mining result at any moment, the incremental mining is widely used in the area of data mining. The incremental parallel Web mining based on Markov chain model (simply called I-PWM) is also proposed as follows.

If the transition probability of state i to state j is p_{ij}, the transition number of state i is D_i, the new transition number of state i to state j is D_{ij}' and the new transition number of state i to other states is D_i', then the new transition probability from state i to state j is $(D_{ij}'+D_i*p_{ij})/(D_i'+D_i)$. The following is the algorithm of I-PWM.

Algorithm 2: incremental parallel mining algorithm (I-PWM).

Input: historic transition probability matrix M_1, historic transition number array A_1, new transition probability matrix M_2, new transition number array A_2.

Output: integrated transition probability matrix M_3, integrated transition number array A_3.

Step1: $A_3[i]= A_1[i]+ A_2[i]$.

Step2: Compute M_3 According to the following conditions.

If $M_1[i,j]>\varepsilon$ and $M_2[i,j]>\varepsilon$,

then $M_3[i,j] = (M_1[i,j]*A_1[i]+M_2[i,j]*A_2[i])/ A_3[i]$.

If $M_1[i,j]=0$ and $M_2[i,j]>\varepsilon$,

then $M_3[i,j] = (M_2[i,j]*A_2[i])/ A_3[i]$.
If $M_1[i,j]>\varepsilon$ and $M_2[i,j]=0$,
then $M_3[i,j] = (M_1[i,j]*A_1[i])/ A_3[i]$.
If $M_1[i,j]=0$ and $M_2[i,j]=0$, then $M_3[i,j]=0$.
Step3: If $M_3[i,j]<\varepsilon$, then $M_3[i,j]=0$.

The time complexity of I-PWM algorithm is O(m*m). Obviously, the result is not quite accurate because the I-PWM algorithm ignores some small probability values. For example, when $M_1[i,j]=0$, $M_2[i,j]>\varepsilon$ and $M_3[i,j]> \varepsilon$, or when $M_1[i,j]>\varepsilon$, $M_2[i,j] =0$ and $M_3[i,j]> \varepsilon$, the result $M_3[i,j]$ is a little smaller than the real result.

4. Experimental results

4.1. Evaluation metrics

There are two metrics to evaluate a parallel algorithm, accuracy and cost of computation [3].

Definition 1. Accuracy: used to measure the testing mining result compared with the standard mining result. Assume that the standard result is R and the testing result is R'. So the accurate of the testing mining algorithm can be defined as 1-|R'-R|/R.

Definition 2. Cost of Computation: used to measure the speedup of parallel algorithms and the communication cost of distributed algorithms. In this paper, the speedup can be represented by the ratio between parallel mining time and mining time based on warehouse. The communication cost can be represented by the communication complexity of the parallel algorithm.

At one extreme, we can ship all the data to a single node. We assume that this produces the most accurate result. In general, this is also the most expensive. At the other extreme, we can process all the data locally to get local mining results, and then combine the local results at the central node to obtain the final result. In general, this approach is less expensive, but also less accurate. In parallel data mining, there is a fundamental trade-off between the accuracy and the cost of the computation. We will mainly discuss the accuracy and speedup of the PWM algorithm.

Experiments were performed on a Web log recorded between 10st and 14st of November, 2004 on the university of Xi'an Jiaotong University Web site, which is 709.55MB in size. According to the round-robin scheduling algorithm of cluster server, the log records are divided into n parts just to simulate n log files of cluster nodes. In the course of distribution, all sessions are kept persistent.

4.2. Results

The original transition matrix is 258*258 and very sparse because many transition probability values are equal to zero. The following will compare the results between Web mining based on warehouse and parallel Web mining. There are two kinds of http services to Web cluster server: session persistence (requests from the same session must be forwarded to the same node) and session non-persistence (requests from the same session can be forwarded to a different node). To Web mining based on warehouse, the two service approaches will not produce any difference. However, to parallel Web mining, they will be different. By setting four different thresholds for compression, we can get the experimental results.

The result of Web mining based on warehouse is quite accurate and can be considered a standard. We will put emphasis on the parallel Web mining for session persistence and session non-persistence. From figure 2, we can find that the result of session persistence is closer to the standard result. Furthermore, we can see that when ε increases, the number of transition will be decreased.

The accuracy of the algorithm has a close relation with the parameter a if the PWM algorithm does not include step 5. When the value of a increases, the accuracy will decrease. Figure 3 shows the change of transition number and accuracy as parameter a increases.

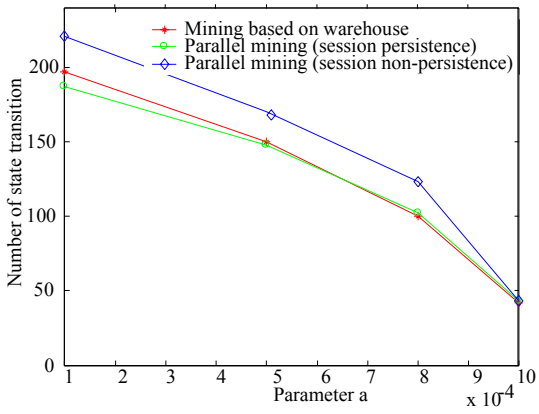

Figure 2. Comparison of Web mining based on warehouse and parallel Web mining

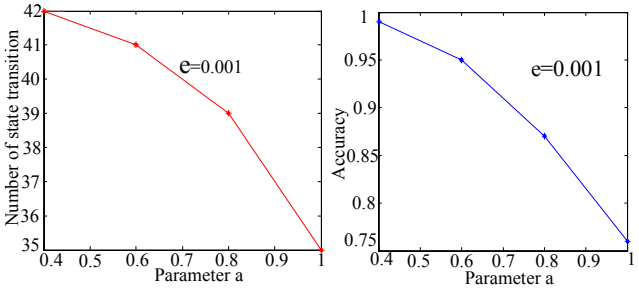

Figure 3. Accuracy of S-PWM algorithm in different parameter a

Figure 4 shows the accuracy comparison between S-PWM and C-PWM. Obviously, the accuracy of C-PWM algorithm will not change with the parameter a. When parameter a becomes smaller, the first traffic will be larger and the second traffic will be smaller; on the contrary, when a becomes larger, the first traffic will be smaller and the second traffic will be larger.

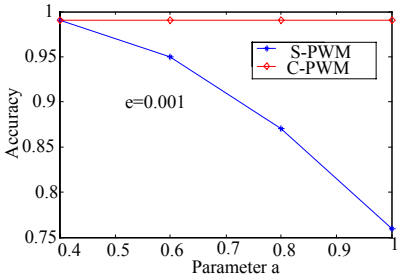

Figure 4. Accuracy comparison between S-PWM and C-PWM

5. Conclusions

Data mining algorithms and underlying techniques can be parallelized to make them effective in the analysis of very large or distributed data sets. The cluster server is an important parallel computing environment. This paper presents an algorithm (PWM) for predicting Web link from Web log files in parallel. The PWM algorithm makes full use of the distribution characteristics of log files in cluster server, obtains the transition probability matrixes from cluster nodes synchronously and then integrates the results. In addition, an incremental algorithm (I-PWM) is also put forward. The experiments show that the PWM algorithm can be used not only for Quantitative computing but also for Qualitative Computing.

References

[1] Mehmet Sayal and Peter Scheuermann, "Distributed Web Log Mining Using Maximal Large Itemsets", Knowledge and Information System, Vol 3, No. 4, pp. 389-404, 2001.

[2] Eui-Hong Han, George Karypis, and Vipin Kumar, "Scalable parallel data mining for association rules", IEEE Transactions on Knowledge and Data Engineering, Vol 12, No. 3, pp. 337-352, 2000.

[3] R. L. Grossman, S. Bailey, A. Ramu, and et al, "Paryrus: A System for Data Ming over Local and Wide Area Clusters and Super-Clusters", Proceedings of Supercomputing, IEEE, 1999.

[4] M. Cannataro and D. Talia, "Parallel and Distributed Knowledge Discovery on the Grid: a Reference Architecture", 4th Int. Conf. On Algorithm and Architecture for Parallel Processing ICA3PP, HongKong, pp. 662-673, December 2000.

[5] Mohammed J.Zaki and Yi Pan, "Introduction: Recent Developments in Parallel and Distributed Data Ming", Distributed and Parallel Databases, Vol 11, No. 2, pp. 123-127, 2002.

[6] Jianhan Zhu, Jun Hong and John G. Hughes, "Using Markov Chains for Link Prediction in Adaptive Web Sites", 1th Int. Conf. on Computing in an Imperfect World, London, pp. 60-73, 2002.

[7] Luo Jianli, Shen Jie, Xu Youzhi, and et al, "A Log Mining Model Based On Distributed Web Servers", Computer Applications and Software, Vol 21, No. 9, pp. 30-35, 2004.

[8] Ramesh R. Sarukkai, "Link Prediction and Path Analysis Using Markov Chains", 9th int. World Wide Web conf. on Computer networks, Amsterdam, pp. 377-386, 2000.