International Journal of Innovative
Computing, Information and Control
Volume 4, Number 11, November 2008
 ISSN 1349-4198
pp. 2875–2887

AN EFFICIENT FUFP-TREE MAINTENANCE ALGORITHM FOR RECORD MODIFICATION

Tzung-Pei Hong[1,2], Chun-Wei Lin[3] and Yu-Lung Wu[4]

[1]Department of Computer Science and Information Engineering
National University of Kaohsiung
Kaohsiung 811, Taiwan
[2]Department of Computer Science and Engineering
National Sun Yat-sen University
Kaohsiung 804, Taiwan
tphong@nuk.edu.tw

[3]Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan 701, Taiwan
p7895122@mail.ncku.edu.tw

[4]Department of Information Management
I-Shou University
Kaohsiung 84008, Taiwan
wuyulung@isu.edu.tw

Received October 2007; revised February 2008

Abstract. *The Frequent-Pattern-tree (FP-tree) is an efficient data structure for association-rule mining without the generation of candidate itemsets. It is used to compress a database into a tree structure, which stores only large items. When the underlying data is updated, the FP-tree, however, needs to process all the transactions in a batch way. In this paper, we thus attempt to extend the FP-tree construction algorithm for the efficient handling of record modification. An expeditious FP-tree (FUFP-tree) structure is used to ease the tree update process. An FUFP-tree maintenance algorithm is also proposed for reducing the execution time in reconstructing the tree when records are modified. Experimental results show that the proposed FUFP-tree maintenance algorithm for record modification runs faster than the batch FP-tree construction algorithm for handling updated records and generates nearly the same tree structure as the FP-tree algorithm. The proposed approach can thus achieve a good trade-off between execution time and tree complexity.*
Keywords: Data mining, FP-tree, FUFP-tree, Record modification, Maintenance

1. **Introduction.** Data mining involves applying specific algorithms to extract patterns, features or rules from data sets in a particular representation. Many mining approaches have been proposed to achieve this purpose [1-3,6,7,9-11,13,14,16-18]. For example, Agrawal and his co-workers proposed several mining algorithms based on the concept of large itemsets to find association rules from transaction data [1-3].

Cheung *et al.* proposed a noticeable incremental mining algorithm, called the Fast Updated Algorithm (FUP) [4] for avoiding the shortcomings of batch mining. The FUP algorithm modified the Apriori mining algorithm [2] and adopted the pruning techniques used in the DHP (Direct Hashing and Pruning) algorithm [14]. It first calculated large itemsets mainly from newly inserted transactions, and compared them with the previous large itemsets from the original database. According to the comparison results, FUP