Closed Constrained Gradient Mining in Retail Databases

Jianyong Wang, *Member*, *IEEE*, Jiawei Han, *Senior Member*, *IEEE*, and Jian Pei

Abstract—Incorporating constraints into frequent itemset mining not only improves data mining efficiency, but also leads to concise and meaningful results. In this paper, a framework for closed constrained gradient itemset mining in retail databases is proposed by introducing the concept of *gradient constraint* into closed itemset mining. A tailored version of CLOSET+, LCLOSET, is first briefly introduced, which is designed for efficient closed itemset mining from sparse databases. Then, a newly proposed weaker but antimonotone measure, top-X average measure, is proposed and can be adopted to prune search space effectively. Experiments show that a combination of LCLOSET and the top-X average pruning provides an efficient approach to mining frequent closed gradient itemsets.

Index Terms—Data mining, frequent closed itemset, association rule, gradient pattern.

1 Introduction

Frequent pattern mining is a well-studied data mining problem; however, it is still unsatisfactory in the sense that its traditional problem formulation is unsuitable to answer comparative analysis queries, such as: "*What items are frequently sold together with some brands of TV which can make 20 percent more profit than the average profit of all kinds of TV?*" Recently, there have been many studies dedicated to constraint-based frequent-itemset mining [5], [6], which demonstrates a promising direction for solving the above problem. A typical such example is DualMiner [1], which uses both monotone and antimonotone constraints to prune the search space and answer questions like "*find all frequent itemsets where the total price is at least $50.*" However, although pushing some monotone or antimonotone constraints into frequent itemset mining often generates compact and interesting result sets, not all constraints have the monotone or antimonotone property so that they can be used directly to prune search space.

Take our previous query, "*What items are frequently sold together with some brands of TV which can make 20 percent more profit than the average profit of all kinds of TV?*" as an example. An itemset that is frequently sold with some highly profitable brands of TV does not imply that any of its superitemsets or subitemsets can also do so, that is, the constraint in the query is neither monotone nor antimonotone. However, this kind of constraints can make the result set more interesting from the application point of view. For example, a typical frequent itemset mining algorithm will usually mine from a retail database many such frequent itemsets sold together with a TV as $\{\mathrm{TV}, \mathrm{VCR}\}$, $\{\mathrm{TV}, \mathrm{DVD}\}$, and so on. Assume the average profit of a TV is \$10. When a TV is sold with a VCR, its average profit is \$8; however, when it is sold with a DVD, its average profit is \$20 (here, we assume some highly profitable brands of TV will be sold together with a DVD with a high probability). If we want to mine the frequent itemsets sold with a TV which can make 50 percent more profit than the average, the itemset $\{\mathrm{TV}, \mathrm{VCR}\}$ will not be included in the result set because it cannot make 50 percent more profit than average.

In this paper, we introduce several constraints related to *gradient* computation into frequent closed itemset mining. These constraints include 1) the *support threshold* (which is a typical antimonotone constraint), 2) the *probe itemset* (such as TV in the above example) that is used as the basis for comparison among various potential gradient itemsets (such as $\{\mathrm{TV}, \mathrm{VCR}\}$ and $\{\mathrm{TV}, \mathrm{DVD}\}$), and 3) the *gradient threshold* (such as 150 percent in the above example), which can be used to remove the nongradient itemsets (such as $\{\mathrm{TV}, \mathrm{VCR}\}$ which cannot make 50 percent more profit than average). Differently from the traditional frequent closed itemset mining, we call this framework *frequent closed constraint-based gradient itemset mining*. Our solution to this problem consists of two parts: an efficient closed itemset mining method, LCLOSET (stands for Lightweight frequent CLOsed itemSET mining), and an effective gradient pruning method, Top-X average, by converting the gradient threshold into a weaker but antimonotone attribute.

In the rest of this paper, we describe the problem definition in Section 2 and introduce the closed gradient mining algorithm by focusing on the Top-X average method in Section 3. Then, we present the performance study in Section 4 and conclude the study in Section 5.

- *J. Wang is with the Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China. E-mail: jianyong@tsinghua.edu.cn.*
- *J. Han is with the Department of Computer Science, University of Illinois at Urbana-Champaign, 201 N. Goodwin Avenue, 2132 Siebel Center for Computer Science, MC-258, Urbana, IL 61801-2302. E-mail: hanj@cs.uiuc.edu.*
- *J. Pei is with the School of Computing Science, Simon Fraser University, 8888 University Drive, Burnaby, BC, V5A 1S6, Canada. E-mail: jpei@cs.sfu.ca.*

Manuscript received 22 Mar. 2005; revised 2 Oct. 2005; accepted 23 Jan. 2006; published online 20 Apr. 2006.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-0102-0305.

TABLE 1
A Retail Database *RDB*

Tid	Set of items	Measure
10	a, c, e, f, m, p	25
20	a, c, d, e, f, m, p	39
30	a, b, c, e, f, g, m	50
40	b, e, f, i	26
50	b, c, e, n, p	45
60	k, l	0

TABLE 2
The Projected Database $RDB|_e$

Tid	Set of items	Ordered set of items	Measure
10	a, c, f, m, p	f, c, a, m, p	25
20	a, c, d, f, m, p	f, c, a, m, p, d	39
30	a, b, c, f, g, m	f, c, a, b, m, g	50
40	b, f, i	f, b, i	26
50	b, c, n, p	c, b, p, n	45

2 Problem Formulation

A retail database *RDB* consists of a set of retail transactions and a retail transaction *RT* is a triple $\langle tid, I, m\rangle$, where *tid* is the corresponding transaction identifier, $I = \{i_1, i_2, \ldots, i_l\}$ is a set of items (we call it an l-itemset if it contains l items) and m is a measure such as profit in dollars. A retail transaction $\langle tid, I, m\rangle$ *contains* an itemset X if $X \subseteq l$. The number of transactions containing X in *RDB* is called the *support* of itemset X, denoted as $sup(X)$. The sum of measures of the transactions containing X is called the *total measure* of X, denoted by $sum\_m(X)$, and $(sum\_m(X)/sup(X))$ is called the *average measure* of X, denoted by $avg\_m(X)$.

Given a probe itemset, P, a support threshold min_sup, and a gradient threshold min_grad, our algorithm will mine frequent closed constrained gradient itemsets defined as follows:

Definition 1 (Projected database). *Given a retail transaction RT, $\langle tid, I, m\rangle$, which contains P, $\langle tid, I - P, m\rangle$ is called a projected transaction with regard to P. The set of all such projected transactions in RDB with regard to P is called the projected database, denoted by $RDB|_P$.*

Definition 2 (Frequent closed itemset). *If the support of an itemset X in RDB is no less than min_sup, we call it a frequent itemset in RDB and, if there is no proper superset of X with the same support as X, X is called a frequent closed itemset in RDB.*

Definition 3 (Frequent closed constrained gradient). *If itemset X is a frequent itemset in $RDB|_P$ and $avg\_m(X)\ \theta\ (avg\_m(P) \times min\_grad)$ holds, where $\theta \in \{\geq, \leq, <, >\}$ (in this paper we use "$\geq$" as an illustration). We call X a frequent constrained gradient. If a frequent constrained gradient X is a closed itemset, it is called a frequent closed constrained gradient.*

The following property shows that all frequent gradient itemsets can be derived from the set of frequent closed gradient itemsets:

Property 1 (Nonredundancy and completeness). *For any frequent gradient itemset X, there exists a closed gradient Y such that $X \subseteq Y$, $sup(X) = sup(Y)$, and*

$$avg\_m(X) = avg\_m(Y).$$

Proof. If X itself is a closed itemset, the property naturally holds. If X is a nonclosed itemset, there must exist a closed itemset, Y, such that $X \subset Y$ and $sup(X) = sup(Y)$. $X \subset Y$ means the set of transactions containing X is a superset of the set of transactions containing Y, thus, if $X \subset Y$ and $sup(X) = sup(Y)$ hold in the mean time, X and Y must appear in the same set of retail transactions and have the same average measure, i.e., $avg\_m(X) = avg\_m(Y)$ also holds. □

Example 1. Table 1 shows a sample retail data set *RDB* that will be used as the running example in this paper. Here, we suppose the probe itemset contains only one item "*e*," $min\_sup = 2$, and $min\_grad = 1.2$. Table 2 shows the projected database with regard to "*e*." The third column in Table 2 is the set of items sorted in item frequency descending order. It can be seen that $avg\_m(e) = 37$ and the associated gradient threshold is 44.4.

2.1 Discussion of the Database Model

In the above retail transaction model, there is a measure associated with each transaction. Depending on the applications of the closed gradient mining algorithm, its measure may be different. The following are several possible cases:

1. The measure can be the profit of probe item "*e*." In this case, we want to mine which itemsets can make the probe item more profitable. Sometimes, for a certain probe item, it may have different prices (or profits) in different transactions. For example, item TV may have tens of brands and each brand usually has a different price (or profit).
2. It can also be the average profit of each item in the projected retail transaction; in this case, we want to mine the itemsets sold together with probe item "*e*," whose support is no less than min_sup and, on average, each item in these itemsets can gain no less than $(avg\_m(e) * min\_grad)$ profit. This is a usual case: In some stores, some promotion items are intentionally priced low to attract customers and promote the sales of some other highly profitable items. Here, the set of promotion items can be used as the probe itemset.
3. Our database model also covers multidimensional transaction data sets. We can treat each dimension value as a virtual item identifier and use our model to mine gradients on multidimensional data sets. In such a way, we can use our algorithm to answer such questions: In which city is TVs sold more profitable than the average? In this case, closed gradient mining can perform similar functions as Cube gradient mining [2].

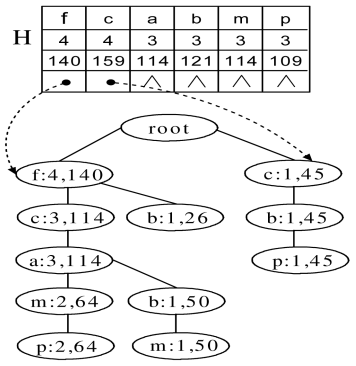

Fig. 1. The FP-tree in the running example.

3 Frequent Closed Gradient Mining

Our strategy to mine closed gradients is to push the gradient constraint into an efficient closed itemset mining algorithm. However, most previously proposed closed itemset mining methods are not specially designed for retail transaction data sets, which are usually very sparse and contain a large number of distinct items. We first briefly introduce LCLOSET, a tailored version of the CLOSET+ [8] algorithm for retail databases, then focus on how to use the gradient constraint to speed up the mining process.

3.1 LCLOSET: A Lightweight Closed Itemset Mining Algorithm

LCLOSET inherits most of the features of CLOSET+ except those not designed for sparse data sets, like the bottom-up physical tree projection method and the subset checking method based on the two-level hash indexed result tree. Specifically speaking, it adopts the prefix-tree structure [4], [9], [8] to represent the original data set and mines frequent itemsets under the divide-and-conquer and depth-first search paradigm. It uses several pruning methods to prune some unpromising search space, which have been popularly used in many closed itemset mining algorithms [10], [8], [3], [7]. Two salient features which are especially helpful for sparse data sets, *top-down pseudo-tree-projection* and *pseudo-projection-based upward checking*, are retained to build conditional databases and check if an itemset is closed or not. Due to limited space, we do not elaborate on most of these techniques and refer the interested readers to [8] for more details. However, as the FP-tree structure used in closed gradient mining is a little different from the one in [8], we will briefly introduce it in the context of gradient mining.

Example 2. The FP-tree of our running example is constructed as follows: Scan the database once to find the set of frequent items and sort them in support descending order to get the $f\_list = \langle f:4, c:4, a:3, b:3, m:3, p:3\rangle$. To insert a transaction into the FP-tree, the infrequent items are removed and the remaining items in the transaction are sorted according to the item ordering in f_list. Fig. 1 shows the FP-tree structure.

When we use the FP-tree structure to mine the closed gradients, the way of building the FP-tree is, to some extent, different from that in [8]. First, as Fig. 1 shows, besides the label and count, we also need to record for each item in the tree nodes and header table the sum of the measures of the transactions where the corresponding item appears. Second, we can prune some nongradient items when we compute the f_list and build FP-tree by applying the Top-K average Apriori property, which has been used to prune search space in cube gradient computing [2]. The Top-K (where K equals min_sup) average is a weaker but antimonotonic attribute: If an item appears in N transactions which are sorted in measure descending order and the top k transactions' average measure cannot satisfy the gradient threshold, any frequent closed itemsets containing this item will fail in satisfying the gradient threshold, thus we can safely remove it from the f_list and the FP-tree. We can also use the *binning* technique to compute the top-K average in an efficient way. As to the details of the top-K average property and the *binning* technique, please refer to [2].

3.2 Closed Constrained Gradient Mining

A naive method to mine closed gradients based on LCLOSET may be like this: By scanning the subtrees and finding the locally frequent items, we also compute the sum of measures for each local item and record them in the header table (as shown in the third row of the header table in Fig. 1) in order to figure out the average measure for a frequent closed itemset and judge if it satisfies the gradient threshold. But, this method does not make full use of the gradient constraint to improve the mining efficiency.

A straightforward thinking is to use the Top-K average property [2] to prune the search space. But, in many cases, the Top-K average property is too weak to prune the search space: Although some items can pass the Top-K average testing, finally, they cannot form any closed gradient itemsets. As a result, we need to design some more effective gradient pruning methods. Following, we will present a newly proposed top-X average pruning method, which is still a weak antimonotonic attribute, but is much stronger.

Given an FP-tree constructed according to the f_list, assume there are totally m leaf nodes labeled as I_1, the last item in f_list, and the set of their supports is $SS = \{S_1, S_2, \ldots, S_m\}$. By choosing any number (from 1 to m) of such nodes and combining them, we can get a *combined support* (i.e., the sum of the supports of the corresponding nodes). The complete set of all the combined supports is denoted by CSS, where

$$CSS = \{S_1, \ldots, S_m, S_1 + S_2, \ldots, S_{m-1} + S_m, \ldots, S_1 + S_2 + \cdots + S_m\}.$$

For example, in Fig. 1, "p" is the last item in f_list, there are two nodes in the FP-tree labeled as "p," here $SS = \{2, 1\}$, $CSS = \{2, 1, 3\}$.

Lemma 1. *The support of any frequent closed itemset containing item I_1 must belong to its combined support set CSS.*

Proof. Because I_1 is the last item in f_list, all of its nodes in the FPtree must be leaf nodes. Here, we use mathematical induction to prove the lemma. If 1-itemset $P_1 = "I_1"$ is a closed itemset, its node support set is $SS = \{S_1, S_2, \ldots, S_m\}$, and its support equals $(S_1 + S_2 + \cdots + S_m)$, which belongs to its CSS. For any closed 2-itemset P_2 which contains I_1 and another item I_2, assuming I_2 appears together with I_1 in

m_2 branches, according to the construction of the FP-tree, the set of their supports, SS_2, must be a subset of SS, P_2's support must be equal to the sum of all the supports in SS_2, which is also an element of CSS.

We assume, for any closed i-itemset containing I_1, P_i, the items in P_i appear together with I_1 in m_i branches, the set of their supports, SS_i, is a subset of SS, and P_i's support is an element of CSS. For any closed $(i+1)$-itemset containing P_i and another item I_{i+1}, P_{i+1}, assuming P_i and I_{i+1} appear together in m_{i+1} branches, let us prove SS_{i+1} is a subset of SS, and its support is an element of CSS. According to the construction of the FP-tree and the Apriori property, SS_{i+1} must be a subset of SS_i. Because SS_i is a subset of SS, SS_{i+1} must be also a subset of SS. Its support equals the sum of the elements in SS_{i+1}, which is also an element of CSS. □

Definition 4 (Top-X average). *For the m nodes labeled as I_1, sort them in average measure descending order. The Top-X average, denoted as $avg^x(I_1)$, is the average measure of the top X nodes, where X is the smallest number satisfying that the sum of the top X nodes' count is no smaller than min_sup.*

For example, in Fig. 1, $avg^x(p) = (45+64)/(1+2) \approx 36.33$ and here x equals 2.

Theorem 1. *If the Top-X average for the last item I_1 in f_list cannot satisfy the gradient threshold, there are no frequent closed gradient itemsets containing I_1.*

Proof. Lemma 1 says the support of any frequent closed itemsets containing item I_1 must fall into set CSS, if I_1's Top-X average fails to satisfy the gradient threshold, i.e., $avg^x(I_1) < (avg\_m(P) \times min\_grad)$, any frequent closed itemset containing I_1 either cannot pass the support threshold or cannot pass the gradient threshold. □

Let us look at the example in Fig. 1. According to Theorem 1, item p can be pruned from the f_list and the FP-tree because $avg^x(p)$ cannot satisfy the gradient threshold. Let us examine whether it is safe to prune item p. For item "p," there are three closed itemsets, "*fcamp*," "*cbp*," and "*cp*." "*fcamp*" has a support 2 and an average measure 32 which fails the gradient threshold. "*cbp*" has an average measure 45 which satisfies the gradient threshold, but its support is lower than min_sup. Also, the average measure of "*cp*" cannot satisfy the gradient threshold. Thus, item "p" can be safely pruned from the FP-tree. We can use Theorem 1 recursively to prune nongradient items from the FP-tree. For example, in Fig. 1, after pruning item "p" from the FP-tree, item "m" becomes the last item in f_list and $avg^x(m) = 38$, which still fails the gradient threshold, so all the nodes labeled as "m" can also be removed from the FP-tree safely.

It can be easily seen that the Top-X average is usually smaller than the Top-K average, so it will be a stronger anti-monotonic attribute in comparison with the Top-K average and will be more effective in pruning search space. For example, in Fig. 1 the Top-K average of item "m," $avg^k(m) = (50+39)/2 = 44.5$, can pass the gradient threshold; however, as we analyzed above, its Top-X average cannot satisfy the gradient threshold. Similarly to the computing of the Top-K average in [2], we can also use the binning technique to compute the Top-X average.

3.3 The FCCGM Algorithm

Both the LCLOSET algorithm and the Top-X average pruning method are based on the FP-tree structure. It is very natural to integrate the Top-X average pruning method with LCLOSET in order to efficiently mine the complete set of frequent closed constrained gradients. The so-derived algorithm is called FCCGM (abbreviated for Frequent Closed Constrained Gradient Mining). FCCGM starts with scanning *RDB* once to build the projected database $RDB|_P$ with regard to probe itemset P. In this process, according to the projected transactions, it computes $avg\_m(P)$ and each projected item's count and its Top-K average using binning technique, which will be used to remove the infrequent items and nongradient (i.e., its top-K average cannot satisfy the gradient threshold) items. The $proj\_RDB$ is scanned once to build the FP-tree and the Top-X average method is used to further prune some nongradient items from the FP-tree. Then, FCCGM uses LCLOSET to recursively mine the frequent closed gradient itemsets.

4 Performance Evaluation

All of our experiments were performed on an Intel Pentium IV processor computer with 256MB memory. We first compare LCLOSET with DCI-CLOSED [3] and LCM [7] using two real data sets, *retail* and *big-market*, then evaluate the effectiveness of the Top-X average pruning method and the scalability of the integrated algorithm, FCCGM, using a set of synthetic data sets. DCI-CLOSED and LCM are two of the most efficient closed itemset mining algorithms in terms of sparse data sets, which can be downloaded from http://fimi.cs.helsinki.fi/.

The *retail* data set contains 88,162 transactions and 16,471 distinct items, while *Big-market* is relatively much larger and contains 838,466 transactions and 38,336 distinct items. The synthetic data sets were generated using the IBM generator and a measure ranging from 1 to 100 was randomly generated and associated with each transaction. Three synthetic data sets, T10I4D100K, T20I10D100K, and T40I10D100K, were used to evaluate the Top-X method, which all contain 100,000 transactions and 1,000 distinct items but have different average transaction lengths, i.e., 10, 20, and 40, respectively. To test the scalability of the algorithm, the T10I4Dx series of data sets were generated by varying the base size from 200K to 1000K transactions and fixing the average transaction length at 10 and average itemset length at 4.

Our experimental results are presented as follows:

4.1 Evaluation of LCLOSET

Fig. 2 shows the runtime comparison results among LCLOSET, DCI-CLOSED, and LCM. For a small *retail* data set, we can get the relationship among the runtime of the three algorithms: DCI-CLOSED $<$ LCLOSET $<$ LCM at a high absolute support (e.g., 2,048), LCM $<$ LCLOSET $<$ DCI-CLOSED at a low absolute support (e.g., 4), while all three algorithms have similar runtime at a moderate

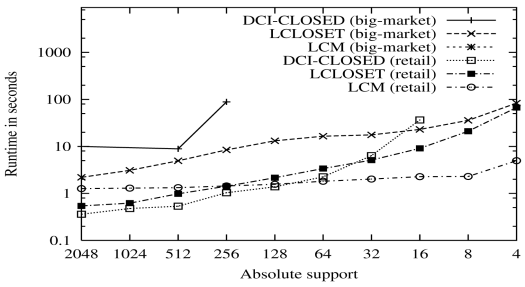

Fig. 2. Runtime.

support (e.g., $32 \leq min\_sup \leq 256$). The *big-market* data set is about 10 times larger than the *retail* data set. For this large *big-market* data set, LCM aborted after some time of running even at a high support of 2,048, thus there is no curve for it in Fig. 2. This implies LCM may have some difficulty in dealing with large data sets. Compared with DCI-CLOSED, LCLOSET is significantly faster for *big-market* data set. Fig. 3 compares the number of frequent itemsets with that of frequent closed itemsets for sparse data sets. It shows that, at a low support, the number of frequent closed itemsets can be an order of magnitude smaller, which implies that mining closed itemsets for sparse data sets is still very compelling.

4.2 Effectiveness of Top-X Average Pruning Method

Fig. 4 and Fig. 5 evaluate the effectiveness of the Top-X method by varying support threshold and fixing gradient threshold at 2 for T10I4D100K and T20I10D100K, 2.2 for T40I10D100K, and by varying gradient threshold and fixing support threshold at 0.8 percent for T10I4D100K and T20I10D100K, 2 percent for T40I10D100K, respectively. We can see that, in both cases, the Top-X method is very effective in accelerating the mining process and, in many cases, it makes the FCCGM algorithm run several times faster.

4.3 Scalability Test

The scalability test with the T10I4Dx series of data sets shows that the FCCGM algorithm has very good scalability: Given a support threshold and a gradient threshold, we find a linear increase in the running time with the increase of base size.

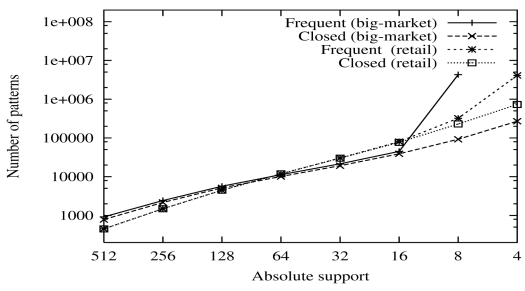

Fig. 3. Number of patterns.

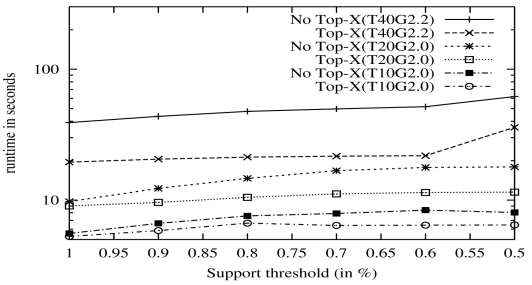

Fig. 4. Effectiveness of the Top-X method by varying the support

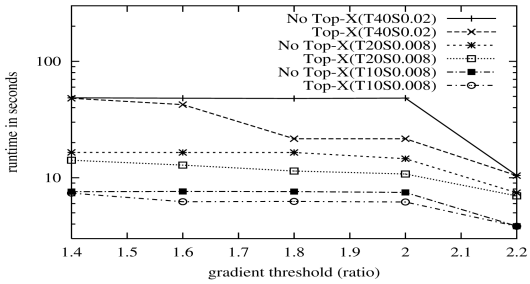

Fig. 5. Effectiveness of the Top-X method by varying the gradient

5 Conclusions

In this paper, we proposed a new problem formulation, *frequent closed constrained gradient mining*, which incorporates the gradient constraint with the traditional frequent closed itemset mining in order to generate some interesting patterns. Under this model, we proposed the Top-X method, which can be easily integrated with the properly devised closed itemset mining algorithm LCLOSET, and derived an efficient gradient mining algorithm FCCGM. Our performance study clearly shows the effectiveness of the algorithm design.

Acknowledgments

The work was supported in part by the US National Science Foundation NSF IIS-02-09199/IIS-03-08215. Jianyong Wang was supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 60573061. Jian Pei was supported in part by NSERC Discovery Grant and the US National Science Foundation NSF IIS-0308001. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] C. Bucila, J. Gehrke, D. Kifer, and W. White, "Dualminer: A Dual-Pruning Algorithm for Itemsets with Constraints," *Data Mining and Knowledge Discovery*, vol. 7, pp. 241-272, 2003.

[2] G. Dong, J. Han, J.M.W. Lam, J. Pei, and K. Wang, "Mining Multi-Dimensional Constrained Gradients in Data Cubes," *Proc. 2001 Int'l Conf. Very Large Data Bases (VLDB '01)*, pp. 321-330, Sept. 2001.

[3] C. Lucchese, S. Orlando, and R. Perego, "DCI-CLOSED: A Fast and Memory Efficient Algorithm to Mine Frequent Itemsets," *Proc. 2004 ICDM Int'l Workshop Frequent Itemset Mining Implementations (FIMI '04)*, Nov. 2004.

[4] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," *Proc. 2000 ACM-SIGMOD Int'l Conf. Management of Data (SIGMOD '00)*, pp. 1-12, May 2000.
[5] R. Ng, L.V.S. Lakshmanan, J. Han, and A. Pang, "Exploratory Mining and Pruning Optimizations of Constrained Associations Rules," *Proc. 1998 ACM-SIGMOD Int'l Conf. Management of Data (SIGMOD '98)*, pp. 13-24, June 1998.
[6] J. Pei, J. Han, and L.V.S. Lakshmanan, "Mining Frequent Itemsets with Convertible Constraints," *Proc. 2001 Int'l Conf. Data Eng. (ICDE '01)*, pp. 433-442, Apr. 2001.
[7] T. Uno, M. Kiyomi, and H. Arimura, "LCM ver. 2: Efficient Mining Algorithms for Frequent/Closed/Maximal Itemsets," *Proc. 2004 ICDM Int'l Workshop Frequent Itemset Mining Implementations (FIMI '04)*, Nov. 2004.
[8] J. Wang, J. Han, and J. Pei, "CLOSET+: Searching for the Best Strategies for Mining Frequent Closed Itemsets," *Proc. 2003 ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '03)*, pp. 236-245, Aug. 2003.
[9] Y. Xu, J.X. Yu, G. Liu, and H. Lu, "From Path Tree to Frequent Patterns: A Framework for Mining Frequent Patterns," *Proc. 2002 Int'l Conf. Data Mining (ICDM '02)*, pp. 514-521, Dec. 2002.
[10] M.J. Zaki and C.J. Hsiao, "CHARM: An Efficient Algorithm for Closed Itemset Mining," *Proc. 2002 SIAM Int'l Conf. Data Mining (SDM '02)*, pp. 457-473, Apr. 2002.

Jianyong Wang received the PhD degree in computer science in 1999 from the Institute of Computing Technology, the Chinese Academy of Sciences. Since then, he has worked as an assistant professor in the Department of Computer Science and Technology, Peking (Beijing) University in the areas of distributed systems and Web search engines and visited the School of Computing Science at Simon Fraser University, the Department of Computer Science at the University of Illinois at Urbana-Champaign, and the Digital Technology Center and Department of Computer Science and Engineering at the University of Minnesota, mainly working in the area of data mining. He is currently an associate professor in the Department of Computer Science and Technology, Tsinghua University, Beijing, China. He is a member of the IEEE, the IEEE Computer Society, and the ACM SIGKDD.

Jiawei Han is a professor in the Department of Computer Science at the University of Illinois at Urbana-Champaign. He has been working on research into data mining, data warehousing, stream data mining, spatiotemporal and multimedia data mining, biological data mining, social network analysis, text and Web mining, and software bug mining, with more than 300 conference and journal publications. He has chaired or served on many program committees of international conferences and workshops. He also served or is serving on the editorial boards for *Data Mining and Knowledge Discovery*, the *IEEE Transactions on Knowledge and Data Engineering*, the *Journal of Computer Science and Technology*, and the *Journal of Intelligent Information Systems*. He is currently serving as the founding editor-in-chief of the *ACM Transactions on Knowledge Discovery from Data* (*TKDD*), and on the Board of Directors for the Executive Committee of the ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD). He is an ACM fellow and a senior member of the IEEE. He has received many awards and recognition, including the ACM SIGKDD Innovation Award (2004) and the IEEE Computer Society Technical Achievement Award (2005).

Jian Pei received the PhD degree in computing science from Simon Fraser University, Canada, in 2002. He is currently an assistant professor of computing science at the same university. In 2002-2004, he was an assistant professor of computer science and engineering at the State University of New York (SUNY) at Buffalo. His research interests can be summarized as developing effective and efficient data analysis techniques for novel data intensive applications. Particularly, he is currently interested in various techniques of data mining, data warehousing, online analytical processing, and database systems, as well as their applications in bioinformatics. His current research is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the US National Science Foundation (NSF). Since 2000, he has published more than 70 research papers in refereed journals, conferences, and workshops, has served on the organization committees and the program committees of more than 60 international conferences and workshops, and has been a reviewer for some leading academic journals. He is a member of the ACM, ACM SIGMOD, and ACM SIGKDD.