Mining Temporal Indirect Associations

Ling Chen[1,2], Sourav S. Bhowmick[1], Jinyan Li[2]

[1] School of Computer Engineering,
Nanyang Technological University, Singapore, 639798
[2] Institute for Infocomm Research, Singapore, 119613

Abstract. This paper presents a novel pattern called *temporal indirect association*. An indirect association pattern refers to a pair of items that rarely occur together but highly depend on the presence of a mediator itemset. The existing model of indirect association does not consider the *lifespan* of items. Consequently, some discovered patterns may be invalid while some useful patterns may not be covered. To overcome this drawback, in this paper, we take into account the lifespan of items to extend the current model to be temporal. An algorithm, *MG-Growth*, that finds the set of mediators in *pattern-growth* manner is developed. Then, we extend the framework of the algorithm to discover temporal indirect associations. Our experimental results demonstrated the efficiency and effectiveness of the proposed algorithms.

1 Introduction

Association rule mining was initially introduced by Agrawal et al. [1]. Traditional association rules discover knowledge from frequent itemsets, i.e., a set of items frequently occur together. However, it has been noted that some of the infrequent itemsets may provide useful insight about the data as well. In [6], a particular type of patterns called *indirect associations* was proposed. A pair of items, x and y, is said to be indirectly associated via an itemset M if they rarely occur together while their respective occurrence highly depends on the presence of the itemset M.

As observed in [3], a notable feature of transaction data is that they are temporal, e.g. transaction products have *lifespan*. The current model of indirect associations does not take into account the lifetime of items, which might lead to some unfair measurement. We explain the incurred problems with the following illustrative examples.

Example 1. *Without considering the lifespan of transaction items, some discovered indirect associations may not be valid.* Figure 1 (a) shows the publication date of a set of web pages. Figure 1 (b) is an example database where each record is a set of pages visited in a web user session. Let the support threshold be 0.4. Based on traditional indirect associations, a pair of two pages is an infrequent itempair if the absolute support of the pair is less than $\lceil 12\times 0.4\rceil$=5, where 12 is the size of the complete database. Since the absolute support of $\{A, E\}$ is 3 ($<$ 5), traditional indirect association will discover indirect associations for

Page	Publication Date
A	May-05
B	May-05
C	Jun-05
D	Jul-05
E	Aug-05
F	Sep-05

(a)

	Session Date	Session ID	Pages
p_1	Jul-05	1	B C
		2	A B D
		3	A B C
		4	B C D
		5	B D
p_2	Aug-05	6	B C E
		7	A C D E
		8	A B C E
		9	A C E
p_3	Sep-05	10	A F
		11	B C E F
		12	A B C F

(b)

Fig. 1. Motivating example.

this pair of items, via some mediators, such as $\{C\}$. However, since page E is published in Aug 05, it is unfair to compute support of itempairs containing page E with respect to the complete database, which contains records starting from Jul 05. Actually, $\{A, E\}$ is frequent with respect to the set of records from Aug 05, e.g. absolute support of $\{A, E\}$ is 3 $\geq \lceil 7\times 0.4 \rceil$=3. Thus, traditional indirect associations discovered for $\{A, E\}$ are not valid.

Example 2. *Without considering the lifespan of transaction items, some valid indirect associations may not be covered.* The traditional indirect association model discovers a pair of items as an indirect association pattern only if there exists an itemset *M* that occurs frequently together with the two items respectively. Since the pair of itemset $\{E, F\}$ in Figure 1 is infrequent, we need to search whether there exists a mediator itemset M such that E and F are indirectly associated via M. Consider the itemset $\{B, C\}$. Since the absolute support values of $\{E, B, C\}$ and $\{F, B, C\}$ are 3 and 2 respectively, both of which are less than $\lceil 12\times 0.4 \rceil$=5, $\{B, C\}$ will not be considered as a candidate mediator of $\{E, F\}$. However, since page E was published in Aug 05, itemset $\{E, B, C\}$ is frequent w.r.t. the set of records from Aug 05, so does itemset $\{F, B, C\}$. Thus, $\{B, C\}$ should be considered as a candidate mediator while traditional indirect association misses it.

Therefore, considering the lifespan of items, the current indirect association model is not able to discover the complete set of valid indirect association patterns. In this paper, we incorporate time in the current model of indirect associations to discover Informally, we discover a pair of items, x and y, as an indirect association pattern via a mediator M only if 1) x and y are infrequent in their *maximal common existing period*; 2) the occurrence of x (resp. y) depends on M in their maximal common existing period as well. Particularly, we call such type of patterns as *temporal indirect associations*. Temporal indirect associations are useful in the applications of traditional indirect associations, such as competitive product analysis [6] and Web usage mining [5], when the lifespan of items are taken into account.

The main contributions of this paper are summarized as follows.

– We proposed the notion of temporal indirect association considering lifespan of items.
– We designed a novel algorithm to discover indirect association patterns and extended the framework of the algorithm to discover temporal indirect association patterns.
– We implemented the developed algorithms and conducted extensive experiments to evaluate the performance of the algorithms.

2 Problem Statement

Considering the time factor, each transaction item is associated with a lifetime. Similar to the definition in [4], we associate each item with a starting time but no ending time as most applications are interested in existing items. Thus, we define a temporal transaction database as follows.

Definition 1 (Temporal Transaction Database). *Let $P =< p_1, \cdots, p_n >$ be a sequence of continuous time periods such that each period is a particular time granularity, e.g. month, quarter, year etc. $\forall\ 1 \leq i \leq j \leq n$, p_i occurs before p_j, denoted as $p_i \leq p_j$. Given a temporal item x, its starting period is denoted as $S(x)$. Given a temporal itemset X, $S(X) = \max(\{S(x)\})$, where $x \in X$. Let I be a set of temporal items s.t. $\forall x \in I, S(x) \leq p_n$. Let T be a temporal transaction, $T \subseteq I$. The occurring period of T is denoted as $O(T)$. Then, D=$\{T|p_1 \leq O(T) \leq p_n\}$ is temporal transaction database on I over P.*

For example, Figure 1 (b) is a temporal transaction database *D* over three periods, $P =< p_1, p_2, p_3 >$, in accordance with the "month" granularity. *I* =$\{$*A, B, C, D, E, F*$\}$, where each item is associated with a starting period. For example, $S(F) = p_3$. Each transaction in *D* is also associated with an occurring period. For example, for the 8^{th} transaction $T = \{A, B, C, E\}$, $O(T) = p_2$.

For the purpose of incorporating lifespan of items, the measures involved in traditional indirect association, *support* and *dependence* [6], need to be extended to be temporal. We now define the temporal measures as follows.

Definition 2 (Temporal Support). *Let D be a temporal transaction database on I over $P =< p_1, \cdots, p_n >$. Let X be a set of temporal items, $X \subseteq I$. The temporal support of X with respect to the subset of D from the period p_i, denoted as $TSup(X, p_i)$, is defined as:*

$$TSup(X, p_i) = \frac{|\{T|X \subseteq T, O(T) \geq p_i, T \in D\}|}{|\{T|O(T) \geq p_i, T \in D\}|}$$

Then the temporal support of X, denoted as $TSup(X)$, can be computed as $TSup(X, S(X))$.

That is, the temporal support of an itemset X is the ratio of the number of transactions that support X to the number of transactions that occur from the starting period of X. For example, consider the temporal transaction database in Figure 1. Let $X = \{B, C, E\}$. Then, $S(X) = p_2$ (because of E). $TSup(X) = 3/7$ since it is supported by three transactions while there are seven transactions starting from p_2.

Definition 3 (Temporal Dependence). *Let D be a temporal transaction database on I over $P = < p_1, \cdots, p_n >$. Let X, Y be two temporal itemsets, $X \subseteq I$, $Y \subseteq I$. The temporal dependence between X and Y, denoted as $TDep(X, Y)$, is defined as:*

$$TDep(X, Y) = \frac{TSup(X \cup Y)}{\sqrt{TSup(X, S(X \cup Y))TSup(Y, S(X \cup Y))}}$$

Since the correlation between two attributes makes sense only when both attributes exist, we calculate the probability of X and Y (in the denominator) with respect to the subset of D from the period where $X \cup Y$ starts. Similar to the traditional definition of dependence in [6], the value of temporal dependence ranges from 0 to 1. The higher the value of temporal dependence, the more positive correlation between the two itemsets. For example, consider the two temporal itemsets $X = \{B, C\}$ and $Y = \{E\}$ in Figure 1. As computed above, $S(X \cup Y) = p_2$, $TSup(X \cup Y) = 3/7$. Since $TSup(X, p_2)$ is $4/7$ and $TSup(Y, p_2)$ is $5/7$, the $TDep(X, Y)$=$\frac{3/7}{\sqrt{4/7 \times 5/7}} = 0.67$.

Based on the temporal support and temporal dependence extended above, the temporal indirect association can be defined as follows.

Definition 4 (Temporal Indirect Association). *A temporal itempair $\{x, y\}$ is a temporal indirect association pattern via a temporal mediator M, denoted as $< x, y|M >$, if the following conditions are satisfied:*

1. $TSup(\{x, y\}) < t_s$ $(Itempair\ Support\ Condition)$.
2. $TSup(\{x\} \cup M) \geq t_f$,$TSup(\{y\} \cup M) \geq t_f$ *(Mediator Support Condition).*
3. $TDep(\{x\}, M) \geq t_d$, $TDep(\{y\}, M) \geq t_d$ *(Mediator Dependence Condition).*

where t_s, t_f, t_d are user defined itempair support threshold, mediator support threshold and mediator dependence threshold respectively.

For example, consider the pair of temporal items $\{E, F\}$ in Figure 1. Let user defined thresholds t_s, t_f, t_d be 0.4, 0.4 and 0.6 respectively. Since $TSup(\{E, F\})$ $= 1/3 < 0.4$, $\{E, F\}$ is an infrequent itempair. Consider $\{B, C\}$ as a candidate mediator. $TSup(\{E, B, C\}) = 3/7 \geq 0.4$, $TSup(\{F, B, C\}) = 2/3 \geq 0.4$. Meanwhile, $TDep(\{E\}, \{B, C\}) = 0.67 \geq 0.6$ and $TDep(\{F\}, \{B, C\}) = 0.82 \geq 0.6$. Thus, $< E, F|\{B, C\} >$ is a temporal indirect association pattern.

TID	Items
1	A B C D E
2	A B E
3	B C E
4	A B D
5	B C D

(a)

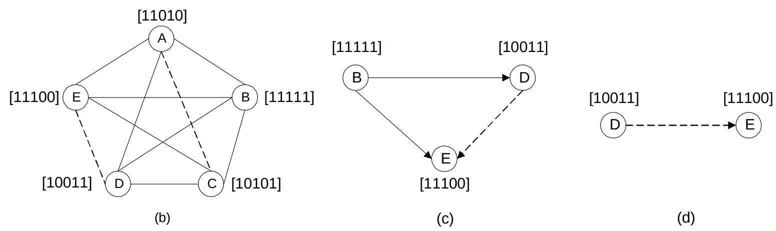


Fig. 2. Indirect association mining.

Problem Statement. Let D be a temporal transaction database over a sequence of time periods $P =< p_1, \cdots, p_n >$. Given user defined thresholds t_s, t_f and t_d, the problem of **temporal indirect association mining** is to discover the complete set of patterns *s.t.* each pattern $< x, y|M >$ satisfies the conditions: 1) $TSup(\{x, y\}) < t_s$; 2) $TSup(\{x\} \cup M) \geq t_f$, $TSup(\{y\} \cup M) \geq t_f$; 3) $TDep(\{x\}, M) \geq t_d$, $TDep(\{y\}, M) \geq t_d$.

3 Algorithm

In this section, we discuss the algorithm for temporal indirect association mining. We first present a novel algorithm for indirect association mining and then extend it to support temporal transaction database.

3.1 Indirect Association Mining

An algorithm called *HI-Mine* was proposed in [7] to use the *divide-and-conquer* strategy to discover mediators. However, *HI-Mine* generates a complete set of mediators for each item x although some of the mediators are useless, e.g. there exists no item y such that $\{x, y\}$ is infrequent and y depends on these mediators as well. Our algorithm addresses this problem by generating a mediator only if there exists an infrequent itempair such that both items depend on it.

Basically, we first construct a *frequency graph* which is used to find *infrequent itempairs* and items that are *possible mediators* of each infrequent itempair. For each infrequent itempair, we then construct a *mediator graph* with these possible mediator items. Then, the complete set of mediators for the infrequent itempair will be generated from the mediator graph.

We use a vertical bitmap representation for the database. For example, consider the transaction database in Figure 2 (a). The bitmap for item A is $[11010]$. Then a *frequency graph* can be defined as follows (For the clarity of exposition, we assume $t_s = t_f$ in the following. The algorithm in Figure 3 explains the situation when $t_s \neq t_f$. Let t_s and t_f be absolute support threshold).

Definition 5 (Frequency Graph). *Given a database D on itemset I, and the user defined mediator (itempair) support threshold t_f, a frequency graph, denoted as $FG = (N, E)$, can be constructed such that N is a set of nodes representing frequent items $\{x|b(x) \geq t_f, x \in I\}$ and E is a set of edges representing* *itempairs. Each node x is associated with the bitmap $b(x)$. Each edge (x, y) is frequent if $b(x) \cap b(y) \geq t_f$. Otherwise, it is infrequent.*

For example, let the threshold t_f be 2. All individual items in the database in Figure 2 (a) are frequent and the constructed *frequency graph* is shown in Figure 2 (b) where infrequent edges are drawn in dashed lines.

Traverse edges in a *frequency graph*. For each infrequent edge, which corresponds to an infrequent itempair, we collect a set of *candidate mediator nodes*.

Definition 6 (Candidate Mediator Node). *Given a frequency graph $FG = (N, E)$, for an infrequent edge $(x, y) \in E$, its candidate mediator nodes, denoted as $MN(x, y)$, is a set of nodes: $\{n | b(n) \cap b(x) \geq t_f, b(n) \cap b(y) \geq t_f, n \in N\}$.*

For example, for the infrequent edge (A, C) in Figure 2 (b), $MN(A, C) = \{B, D, E\}$. Then, a *mediator graph* for an infrequent edge can be constructed with the set of candidate mediator nodes.

Definition 7 (Mediator Graph). *Given a frequency graph FG and an infrequent edge (x, y), a mediator graph created for (x, y) is a directed graph, denoted as $MG(x, y) = (N, E)$, where N is a set of nodes such that $N = MN(x, y)$ and E is a set of directed edges. Each node n is associated with a bitmap $b(n)$ as in FG. Each edge $(m \rightarrow n)$, originating from m if m precedes n according to lexicographical order, is frequent if $b(m) \cap b(n) \geq t_f$.*

For example, the mediator graph constructed for infrequent edge (A, C) is shown in Figure 2 (c). Likewise, infrequent edges are shown in dashed lines.

From the mediator graph $MG(A, C)$, we now present how to compute the set of mediators for infrequent itempair $\{A, C\}$. Let the threshold of support be 0.4 and threshold of dependence 0.6. We first consider the candidate mediator node B. $support(\{A, B\}) = 3/5$ because $b(A) \cap b(B) = 3$. $dependence(A, B)$ $\frac{support(\{A,B\})}{\sqrt{support(A) \times support(B)}} = \frac{3}{\sqrt{3 \times 5}} = 0.77$. The support and the dependence between C and B can be calculated similarly and we discover an indirect association pattern $< A, C | \{B\} >$.

The remaining nodes in the mediator graph that have frequent edges originating from node B consist of $B's\ conditional\ mediator\ base$, from which we construct $B's\ conditional\ mediator\ graph$. For each node n in the conditional mediator graph of node B, its bitmap is updated by joining with the bitmap of node B. After that, each edge $(m \rightarrow n)$ is frequent if $b(m) \cap b(n) \geq t_f$. For example, Figure 2 (d) shows the conditional mediator graph of node B. Then, we compute the mediators involving B, such as $\{BD\}$ and $\{BE\}$, for itempair $\{A, C\}$. Similarly, the support and dependence between A and $\{BD\}$ can be calculated by joining $b(A)$ with $b(D)$ (Note that $b(D)$ represents the support of $\{BD\}$ now) while the support and dependence between C and $\{B, D\}$ can be

(a) MG-Growth

Input: Database D, t_s, t_f and t_d
Output: The complete set of indirect associations S
Description:

Scan D to find $F_1 = \{x|Sup(x) \geq t_f\}$.
Construct the frequency graph FG with F_1.
for each edge (x, y) in FG **do**
  if $Sup(x, y) < t_s$ **then**
    Construct mediator graph $MG(x, y)$
    if $MG(x, y) \neq \emptyset$ **then**
      MGrowth($MG(x, y), M, 0, C$)
      $S = S \cup C$
    end if
  end if
  return S
end for
function MGrowth($MG(x, y), M, dep, C$)
  for each node n in $MG(x, y)$ **do**
    $M[dep] = n; dep + +$
    if $Sup(n, x) \geq t_f$ && $Dep(n, x) \geq t_d$ && $Sup(n, y) \geq t_f$ && $Dep(n, y) \geq t_d$ **then**
      $C = C \cup \{< x, y|M >\}$
    end if
    Construct conditional mediator graph $MG_n(x, y)$
    if $MG_n(x, y) \neq \emptyset$ **then**
      MGrowth($MG_n(x, y), M, dep, C$)
    end if
    $dep - -$
  end for
end function

(b) TMG-Growth

Input: Temporal transaction database D, t_s, t_f and t_d
Output: The complete set of indirect associations S
Description:

Scan D to find $F_1 = \{x|TSup(x) \geq t_f\}$.
Construct the frequency graph FG with F_1.
for each edge (x, y) $s.t. S(x) = p_i, S(y) = p_j$ in FG **do**
  if $TSup(x, y) < t_s$ **then**
    Construct mediator graphs $\{MG^{p_i}(x, y), \cdots, MG^{p_n}(x, y)\}$
    for each graph $MG^{p_k}(x, y) \neq \emptyset$ **do**
      TMGrowth($MG^{p_k}(x, y), M, 0, C$)
      $S = S \cup C$
    end for
  end if
  return S
end for
function TMGrowth($MG^{p_k}(x, y), M, dep, C$)
  for each node n in $MG^{p_k}(x, y)$ **do**
    if $dep == 0$ && n is non-extendable **then**
      return;
    end if
    $M[dep] = n; dep + +$
    if $TSup(n, x) \geq t_f$ && $TDep(n, y) \geq t_d$ && $TSup(n, y) \geq t_f$ && $TDep(n, y) \geq t_d$ **then**
      $C = C \cup \{< x, y|M >\}$
    end if
    Construct $MG_n^{p_k}(x, y)$
    if $MG_n^{p_k}(x, y) \neq \emptyset$ **then**
      TMGrowth($MG_n^{p_k}(x, y), M, dep, C$)
    end if
    $dep - -$
  end for
end function

Fig. 3. Algorithms of *MG-Growth* and *TMG-Growth*.

computed with $b(C) \cap b(D)$. The complete algorithm, *MG-Growth*, is given in Figure 3 (a).

3.2 Temporal Indirect Association Mining

Based on the measure of *temporal support*, a frequency graph consisting of frequent items can be constructed similarly. For example, let the threshold of temporal support be 0.4. The constructed frequency graph is shown in Figure 4 (b).

Before discussing how to construct a mediator graph for an infrequent itempair, we highlight that the downward closure property does not hold for mediator discovery in temporal indirect association mining, e.g even if B is not a mediator of the infrequent itempair $\{A, C\}$, it is possible that $\{BD\}$ is a mediator of $\{A, C\}$. Hence, in order to discover the complete set of mediators for each

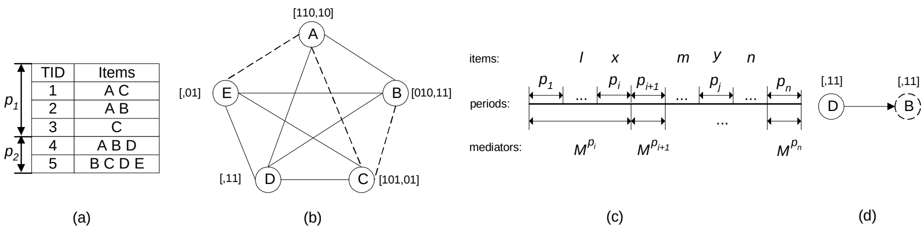


Fig. 4. Temporal transaction database.

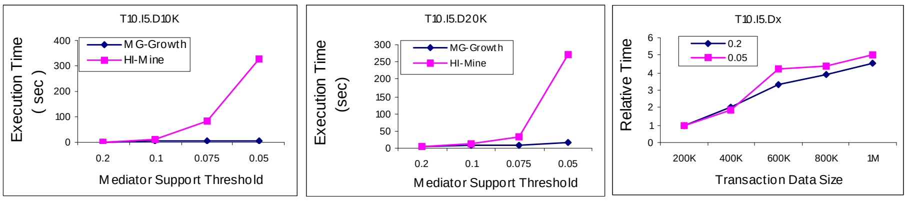
infrequent itempair, we divide the set of mediators according to their lifespan. Given a sequence of periods $P =< p_1, \cdots, p_n >$ as shown in Figure 4 (c), the complete set of mediators M of an infrequent itempair $\{x, y\}$, where $S(x) = p_i$ and $S(y) = p_j$ $(p_i \leq p_j)$, can be divided into $n - i + 1$ subsets as shown in the figure: $M = M^{p_i} \cup M^{p_{i+1}} \cup \cdots \cup M^{p_n}$, where $M^{p_i} = \{X | X \in M, S(X) \leq p_i\}$ and $\forall p_{i+1} \leq p_k \leq p_n, M^{p_k} = \{X | X \in M, S(X) = p_k\}$. When discovering mediators of M^{p_i}, we use the two corresponding subsets of database as counting bases (for computing temporal support and temporal dependence of x and mediators, y and mediators respectively). We create different temporal mediator graphs for discovering different subsets of mediators.

Consider the frequency graph in Figure 4 (b). We now explain how to discover mediators for the infrequent edge (A, C) where $S(A) = S(C) = p_1$. First, we construct the mediator graph for mining M^{p_1}, which involves item B only. Since edge (B, C) is infrequent, there is no candidate mediator nodes and the graph is empty. Then, we construct the mediator graph for mining M^{p_2}, which involves items D and B because the edge (B, C) turns to be frequent with respect to the subset of database from p_2. Note that, D is an extendable mediator node while B is non-extendable[3]. The constructed mediator graph is shown in Figure 4 (d), where non-extendable nodes are depicted in dashed lines. From this graph, we recursively examine whether $\{D\}$ and $\{D, B\}$ are mediators of $\{A, C\}$. The algorithm for mining temporal indirect associations is shown in Figure 3 (b).

4 Performance Evaluation

In this section, we evaluate the performance of developed algorithms. All experiments are conducted on a 2GHz P4 machine with 512M main memory, which runs Microsoft Windows XP. All the algorithms are implemented in C++. In order to obtain comparable experimental results, the method we employed to generate synthetic datasets is similar to the one used in prior works [7]. Without loss of generality, we use the notation $Tx.Iy.Dz$ to represent a data set where the number of transactions is z, the average size of transaction is x and

[3] See the definitions of extendable and non-extendable mediator nodes in our online version [2].



(a) (b) (c)

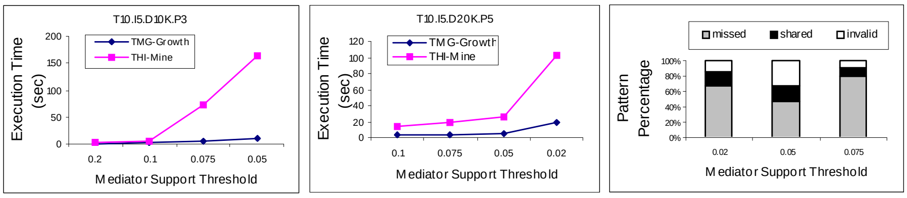
Fig. 5. Experimental Results I.

the average size of potentially large itemsets is y. Additionally, we use the notation $Tx.Iy.Dz.Pn$ to represent a temporal transaction database which is over a sequence of n periods.

Comparison of *MG-Growth* and *HI-Mine*. we compare the performance of *MG-Growth* with *HI-Mine*, which is the clear winner of the other existing algorithms [7]. We ran experiments on two datasets: $T10.I5.D10K$ and $T10.I5.D20K$. The threshold of t_s and t_f are set as the same. The threshold of t_d is set as 0.1. The results are shown in Figure 5. *MG-Growth* is more efficient than *HI-Mine*, especially when t_f (t_s) is small. This is because when the threshold is small, there are more frequent individual items. Consequently, *HI-Mine* needs to discover all the set the mediators for more items no matter whether these mediators are useful or not. On the contrary, *MG-Growth* discovers a mediator only if it is depended on by an infrequent itempair. Thus, the performance of *MG-Growth* will not deteriorate significantly with the decrease of mediator (itempair) support threshold.

We further examine the scale-up feature of *MG-Growth*. Figure 5 (c) shows the results with the variation of data size from $200K$ to $1M$. The scale-up performance under two different thresholds of t_f are studied. The execution times are normalized with respect to the execution time for the data set of $200K$. We observed that the run time of *MG-Growth* increases slightly with the growth of data size, which demonstrated the good scalability of *MG-Growth*.

Comparison of *TMG-Growth* and *THI-Mine*. In order to evaluate the performance of the temporal version of *MG-Growth*, *TMG-Growth*, we also extend the *HI-Mine* to support temporal transaction database [2]. Correspondingly, we denote the temporal version of *HI-Mine* as *THI-Mine*. We compare the performance of *TMG-Growth* and *THI-Mine* with respect to two datasets: $T10.I5.D10K.P3$ and $T10.I5.D20K.P5$. Figures 6 (a) and (b) present the results respectively. Obviously, the temporal version of *MG-Growth* outperforms the temporal version of *HI-Mine* as well. When the number of periods increases, the gap between the two algorithms is apparent even if the mediator support threshold is large.



(a) (b) (c)

Fig. 6. Experimental Results II.

We evaluate the quality of temporal indirect association patterns by comparing the results of the traditional model and the temporal model on the same temporal transaction database. Figure 6 (c) shows the results with respect to the variation of t_f threshold, where black blocks depict the percentage of patterns shared by two models, white blocks depict the percentage of patterns missed by the traditional model and the gray blocks depict the percentage of invalid patterns. It can be observed that the set of temporal indirect association patterns is significantly different from the results of the traditional model.

5 Conclusions

In this paper, we take into account the lifespan of items to explore a new model of temporal indirect association. We first develop an algorithm *MG-Growth* for indirect association mining. Under *MG-Growth*, a set of mediators are generated only if both items in an infrequent itempair depend on them. Then, we extend the framework of *MG-Growth* so that mediators starting from different periods are discovered separately. Our experimental results showed that *MG-Growth* outperforms the existing algorithm significantly and its extended version discovers the temporal indirect association pattern efficiently.

References

1. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proc. of ACM SIGMOD*, 1993.
2. L. Chen, S. Bhowmick, and J. Li. Temporal indirect association mining. In *http://www.cais.ntu.edu.sg/∼sourav/papers/TIA_PAKDD-05.pdf*.
3. X. Chen, I. Petrounias, and H. Heathfield. Discovery of association rules in temporal databases. In *Proc. of IADT*, 1998.
4. C. Lee, M. Chen, and C. Lin. Progressive partition miner: an efficient algorithm for mining general temporal association rules. In *IEEE TKDE vol.15, no. 4*, 2003.
5. P. Tan and V. Kumar. Mining indirect associations in web data. In *Proc. of WebKDD*, 2001.
6. P. Tan, V. Kumar, and J. Srivastava. Indirect association: mining higher order dependencies in data. In *Proc. of PKDD*, 2000.
7. Q. Wan and A. An. Efficient mining of indirect associations using hi-mine. In *Proc. of Canadian Conference on AI*, 2003.