Transitional Patterns and Their Significant Milestones

Qian Wan and Aijun An
Department of Computer Science and Engineering
York University, Toronto, Ontario, Canada
{qwan, aan}@cse.yorku.ca

Abstract

Mining frequent patterns in transaction databases has been studied extensively in data mining research. However, most of the existing frequent pattern mining algorithms do not consider the time stamps associated with the transactions. In this paper, we extend the existing frequent pattern mining framework to take into account the time stamp of each transaction and discover patterns whose frequency dramatically changes over time. We define a new type of patterns, called transitional patterns, to capture the dynamic behavior of frequent patterns in a transaction database. Transitional patterns include both positive and negative transitional patterns. Their frequencies increase/decrease dramatically at some time points of a transaction database. We introduce the concept of significant milestones for a transitional pattern, which are time points at which the frequency of the pattern changes most significantly. Moreover, we develop an algorithm to mine from a transaction database the set of transitional patterns along with their significant milestones. Our experimental studies on real-world databases illustrate that mining positive and negative transitional patterns is highly promising as a practical and useful approach to discovering novel and interesting knowledge from large databases.

1 Introduction

A transaction database usually consists of a set of time-stamped transactions. Mining frequent itemsets or patterns from a transaction database is one of the fundamental and essential operations in many data mining applications, such as discovering association rules, strong rules, correlations, and many other important discovery tasks. The problem of mining frequent itemsets is formulated as finding all the itemsets from a transaction database that satisfy a user specified support threshold.

Since it was first introduced in 1993 [1], the problem of frequent itemset mining has been studied extensively by many researchers. As a result, a large number of algorithms have been developed in order to efficiently solve the problem [2, 6]. In practice, the number of frequent patterns generated from a data set can often become excessively large, and most of them are useless or simply redundant. Thus, there has been recent interest in discovering a class of new patterns, including maximal frequent itemsets [3, 4], closed frequent itemset [9, 12] , emerging patterns [5, 8], and indirect associations [10, 11].

Despite the abundance of previous work, most of the existing frequent pattern mining algorithms do not consider the time stamps associated with the transactions. Therefore, the dynamic behavior of the discovered patterns are often undetected. In this paper, we extend the traditional frequent pattern mining framework to take into account the time stamp of each transaction, i.e., the time when the transaction occurs. We define a new type of patterns, called transitional patterns, to represent patterns whose frequency dramatically changes over time. Transitional patterns include both positive and negative transitional patterns (to be defined in Section 2.2). The frequency of a positive transitional pattern increases dramatically at some time point of a transaction database, while that of a negative transitional pattern decreases dramatically at some point of time. We illustrate transitional patterns using an example as follows.

Consider an example database TDB as shown in Table 1, which has 16 transactions of 8 items. Let's focus on two patterns, P_1P_2 and P_1P_3. Without considering the time information of these transactions, P_1P_2 and P_1P_3 have the same significance in the traditional frequent pattern framework since they have the same frequency 62.50%. However, interesting differences between these two patterns can be found after we consider the time information of each transaction in the database, as shown in the third column of Table 1. For simplicity, suppose TDB contains all the transactions from November 2005 to February 2007, one transaction per month. We can easily see that before (and including) May 2006, pattern P_1P_2 appears every month; but after May 2006, P_1P_2 only occurs 3 times in 9 transactions, which is equivalent to a frequency of 33.33%. That is to

Table 1. An example dataset *TDB*

TID	List of itemIDs	Time stamp	Time point
001	P_1, P_2, P_3, P_5	Nov. 2005	6.25%
002	P_1, P_2	Dec. 2005	12.5%
003	P_1, P_2, P_3, P_8	Jan. 2006	18.75%
004	P_1, P_2, P_5	Feb. 2006	25%
005	P_1, P_2, P_4	Mar. 2006	31.25%
006	P_1, P_2, P_4, P_5, P_6	Apr. 2006	37.5%
007	P_1, P_2, P_3, P_4, P_6	May. 2006	43.75%
008	P_1, P_4, P_6	Jun. 2006	50%
009	P_4, P_5, P_6	Jul. 2006	56.25%
010	P_1, P_2, P_3, P_4, P_5, P_6	Aug. 2006	62.5%
011	P_1, P_3, P_4, P_6	Sep. 2006	68.75%
012	P_1, P_3, P_5	Oct. 2006	75%
013	P_1, P_2, P_3, P_6, P_7	Nov. 2006	81.25%
014	P_1, P_3, P_4, P_5	Dec. 2006	87.5%
015	P_1, P_3, P_4	Jan. 2007	93.75%
016	P_1, P_2, P_3, P_5	Feb. 2007	100%

say that the frequency or support of pattern P_1P_2 decreases significantly after May 2006. On the other hand, after July 2006, the frequency of pattern P_1P_3 increases significantly from 33.33% to 100%.

The above observations have shown that frequent patterns discovered by standard frequent pattern mining algorithms may be different in terms of their distributions in a transaction database. However, such patterns cannot be distinguished with the standard algorithms. The objective of the research presented in this paper is to distinguish such frequent patterns, discover frequent patterns whose frequency changes significantly over time and identify the time points for such significant changes.

Transitional patterns have a wide range of potential applications. For example, in the market basket scenario, transitional patterns allow business owners to identify those products or combinations of products that have recently become more and more popular (or not as popular as before) so that they can adjust their marketing strategy or optimize product placement in retail environments. In medical domains, a significant increase in the occurrence of certain symptom in a group of patients with the same disease may indicate a side effect of a new drug. Finding the time point when this symptom starts to occur more often can help to identify the drug that causes the problem.

2 Transitional patterns

2.1 Preliminaries - frequent patterns

Let $\mathcal{I} = \{i_1, i_2, \ldots, i_m\}$ be a set of *m items*. A subset $X \subseteq \mathcal{I}$ is called an *itemset*. A *k*-itemset is an itemset that contains *k* items. Let $\mathcal{D} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n\}$ be a set of *n* transactions, called a *transaction database*, where each transaction $\mathcal{T}_j$ ($j \in \{1, 2, \ldots, n\}$) is a set of items such that $\mathcal{T}_j \subseteq \mathcal{I}$. Each transaction is associated with a unique identifier, called its *TID*. A transaction $\mathcal{T}_j$ contains an itemset *X* if and only if $X \subseteq \mathcal{T}_j$.

The count of an itemset X in $\mathcal{D}$, denoted as $count_{\mathcal{D}}(X)$, is the number of transactions in $\mathcal{D}$ containing X. An itemset *X* in a transaction database $\mathcal{D}$ has a *support*, denoted as $sup_{\mathcal{D}}(X)$, which is the ratio of transactions in $\mathcal{D}$ containing *X*. That is,

$$sup_{\mathcal{D}}(X) = \frac{count_{\mathcal{D}}(X)}{||\mathcal{D}||} \quad (1)$$

where $||\mathcal{D}||$ is the total number of transactions in $\mathcal{D}$.

Given a transaction database $\mathcal{D}$ and a user-specified minimum support threshold min_sup, an itemset X is called a frequent itemset or frequent pattern if $sup_{\mathcal{D}}(X) \geq min\_sup$. Accordingly, X is called an infrequent itemset or infrequent pattern if $sup_{\mathcal{D}}(X) < min\_sup$.

2.2 Positive and negative transitional patterns

In order to provide formal definitions of transitional patterns, we first introduce the concept of *time points*. Suppose that the transactions in a transaction database $\mathcal{D}$ are ordered according to their time stamps. A time point, denoted by τ, represents a position in the transaction database $\mathcal{D}$ that separates $\mathcal{D}$ into two disjoint parts, $\mathcal{D}_\tau^-$ and $\mathcal{D}_\tau^+$. We use $\mathcal{D}_\tau^-$ to represent the set of transactions in $\mathcal{D}$ that occur before τ, and $\mathcal{D}_\tau^+$ to represent the set of transactions in $\mathcal{D}$ that occur after τ. Thus, $\mathcal{D} = \mathcal{D}_\tau^- \cup \mathcal{D}_\tau^+$, and τ can be represented by a percentage to indicate a position in $\mathcal{D}$ as follows:

$$\tau = \frac{||\mathcal{D}_\tau^-||}{||\mathcal{D}||} \times 100\% \quad (2)$$

For example, in the example database TDB in Table 1, when τ = 25%, TDB_τ^- contains the first 4 transactions and TDB_τ^+ contains the last 12 transactions. Given a dataset, a time point corresponds to a time stamp. Thus, the number of possible time points is the number of different time stamps in the dataset, which can be equal to or less than the number of transactions in the dataset, assuming that some transactions may occur at the same time. The time points for the example dataset TDB is shown in Table 1.

It's easy to see that the value of τ is between 0% and 100%. However, in practice, the range of τ, denoted as $\top_{\mathcal{D}}$, can be specified by the user according to their own interest. For instance, in order to find interesting patterns in the example database TDB which occur during the year 2006, $\top_{TDB}$ should be set to [12.50% ... 87.50%], since 12.50% is the starting time point for 2006 and 87.50% is the ending point of 2006.

Given a time point τ in $\top_{\mathcal{D}}$, the supports of a pattern X in $\mathcal{D}_\tau^-$ and $\mathcal{D}_\tau^+$ are denoted as $sup_\tau^-(X)$ and $sup_\tau^+(X)$, respectively. That is,

$$sup_\tau^-(X) = sup_{\mathcal{D}_\tau^-}(X) = \frac{count_{\mathcal{D}_\tau^-}(X)}{||\mathcal{D}_\tau^-||} \quad (3)$$

$$sup^+_\tau(X) = sup_{\mathcal{D}^+_\tau}(X) = \frac{count_{\mathcal{D}^+_\tau}(X)}{||\mathcal{D}^+_\tau||} \quad (4)$$

Definition 2.1 *The transitional ratio of pattern X at time point τ is defined as:*

$$tran_\tau(X) = \frac{sup^+_\tau(X) \; - \; sup^-_\tau(X)}{MAX(sup^+_\tau(X), \; sup^-_\tau(X))} \quad (5)$$

where X must exist in the database $\mathcal{D}$ so that the denominator cannot be zero.

Definition 2.2 *A pattern X is a Transitional Pattern (TP) in $\mathcal{D}$ with respect to a time period $\top_\mathcal{D}$, if there exists a time point τ in $\top_\mathcal{D}$ such that:*

(a) $sup^-_\tau(X) \geq t_s$ and $sup^+_\tau(X) \geq t_s$;
(b) $|tran_\tau(X)| \geq t_t$.

where t_s and t_t are called pattern support threshold and transitional pattern threshold, respectively. Moreover, X is called a Positive Transitional Pattern (PTP) when $tran_\tau(X) > 0$; and X is called a Negative Transitional Pattern (NTP) when $tran_\tau(X) < 0$.

For example, if $t_s = 0.05$ and $t_t = 0.5$, pattern P_1P_3 in the example database TDB is a positive transitional pattern because there exists a time point (such as 37.5% corresponding to the end of April 2006) where the transitional ratio of the pattern is greater than 0.5 and the pattern is frequent on both corresponding splits of the datasets. Similarly, P_1P_2 is a negative transitional pattern in TDB.

Note that a pattern can be both a positive transitional pattern and a negative transitional pattern in the same transaction database if there exist two time points τ_1 and τ_2 so that conditions (a) and (b) are satisfied on both τ_1 and τ_2, $tran_{\tau_1}(X) > 0$ and $tran_{\tau_2}(X) < 0$. For example, in the example database TDB, pattern P_4P_6 is both a positive transitional pattern and a negative transitional pattern because its transitional ratio at time point 37.50% is 66.67% and the one at time point 62.50% is -66.67%, and condition (a) is also satisfied on both time points.

The reason we have condition (a) for a transitional pattern is that, if we don't have this condition, any pattern that does not occur at the beginning of the transaction database has a transitional ratio equal to 1 when the pattern first occurs in the database (or any pattern that does not occur at the end of the transaction database has a transitional ratio equal to -1 after its last occurrence in the database). However, such a pattern may be just a sporadic pattern that occurs occasionally in the database, which is not interesting at all. In other words, we are only interested in frequent patterns whose frequency changes dramatically during the time period $\top_\mathcal{D}$ in the transaction database.

2.3 Significant milestones

There may be multiple time points at which a transitional pattern satisfies conditions (a) and (b) in Definition 2.2. People are usually interested in the time points where the frequency of a transitional pattern changes the most significantly. Below we define the concept of *significant milestones* to represent such points. The significant milestones can be classified into frequency-ascending milestones and frequency-descending milestones.

Table 2. Example patterns in *TDB* (%)

τ	$tran_\tau(P_1)$	$tran_\tau(P_1P_2)$	$tran_\tau(P_1P_3)$	$tran_\tau(P_4P_6)$
25	-8.33	-50	25	100
31.25	-9.09	-54.55	45	100
37.50	-10	-60	58.33	66.67
43.75	-11.11	-66.67	44.9	35.71
50	-12.5	-57.14	57.14	0
56.25	11.11	-44.9	66.67	-35.71
62.50	10	-58.33	60	-66.67
68.75	9.09	-45	54.55	-100
75	8.33	-25	50	-100

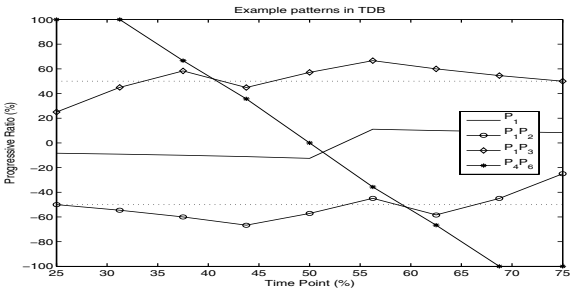

Figure 1. Transitional ratios in TDB

Definition 2.3 *The significant frequency-ascending milestone of a positive transitional pattern X within a time period $\top_\mathcal{D}$ is defined as a tuple, $\langle \mathcal{M}^+, tran_{\mathcal{M}^+}(X)\rangle$, where $\mathcal{M}^+$ is a time point in $\top_\mathcal{D}$ such that:*

1. *$sup^-_{\mathcal{M}^+}(X) \geq t_s$;*
2. *$\forall \tau \in \top_\mathcal{D}$, $tran_{\mathcal{M}^+}(X) \geq tran_\tau(X)$.*

Table 2 lists the transitional ratios of four patterns for all the valid time points between 25% and 75% in the example database TDB. Figure 1 illustrates how the transitional ratios of these four patterns change along the time points. Assuming that the support threshold is 5% and the transitional pattern threshold is 50%, P_1P_3 and P_4P_6 are positive transitional patterns. The significant milestone for P_1P_3 is $\langle$56.25%, +66.67$\rangle$, and the significant milestone for P_4P_6 is $\langle$37.50%, +66.67$\rangle$. Note that even though the transitional ratio of P_4P_6 is 1 at time points 25% and 31.25%, they are not considered to be milestones because they do not satisfy condition 1 in Definition 2.3 due to the fact that P_4P_6 does not occur before time point 31.25%.

Similarly, the significant frequency-descending milestone for a negative transitional pattern is defined below.

Definition 2.4 *The significant frequency-descending milestone of a negative transitional pattern Y within a time period $\top_\mathcal{D}$ is defined as a tuple, $\langle \mathcal{M}^-, tran_{\mathcal{M}^-}(Y)\rangle$, where $\mathcal{M}^-$ is a time point in $\top_\mathcal{D}$ such that:*

1. $sup^{+}_{\mathcal{M}^-}(Y) \geq t_s$;
2. $\forall\, \tau \in \top_{\mathcal{D}}$, $tran_{\mathcal{M}^-}(Y) \leq tran_{\tau}(Y)$.

To give an example, patterns P_1P_2 and P_4P_6 in Table 2 are negative transitional patterns. Their significant frequency-descending milestones are $\langle$43.75%, -66.67$\rangle$ and $\langle$62.50%, -66.67$\rangle$, respectively.

Note that a transitional pattern may have both significant frequency-ascending and significant frequency-descending milestones if it is both a positive and a negative transitional pattern. Also, a positive (or negative) transitional pattern may have more than one significant frequency-ascending (or frequency-descending) milestones.

3 Mining transitional patterns and their significant milestones

In this section, we present an algorithm, called *TP-mine*, for mining the set of positive and negative transitional patterns and their significant milestones with respect to a pattern support threshold and a transitional pattern threshold. The algorithm is given as follows.

1: Extract frequent patterns, P_1, P_2, ..., P_n, and their supports using a frequent pattern generation algorithm with *min_sup* = t_s.
2: Scan the transactions from the first transaction to the transaction right before $\top_{\mathcal{D}}$ to compute the support counts of all the n frequent patterns on this part of the database.
3: $\mathcal{S}_{PTP} \leftarrow \emptyset$, $\mathcal{S}_{NTP} \leftarrow \emptyset$
4: **for all** i = 1 to n **do**
5: $\quad MaxTran(P_i) = 0$, $MinTran(P_i) = 0$
6: $\quad S_{FAM}(P_i) = \emptyset$, $S_{FDM}(P_i) = \emptyset$
7: **end for**
8: **for all** $\tau \in \top_{\mathcal{D}}$ **do**
9: $\quad$ **for** i = 1 to n **do**
10: $\quad\quad$ **if** $sup^{-}_{\tau}(P_i) \geq t_s$ **and** $sup^{+}_{\tau}(P_i) \geq t_s$ **then**
11: $\quad\quad\quad$ **if** $tran_{\tau}(P_i) \geq t_t$ **then**
12: $\quad\quad\quad\quad$ **if** $P_i \notin S_{PTP}$ **then**
13: $\quad\quad\quad\quad\quad$ Add P_i to $\mathcal{S}_{PTP}$
14: $\quad\quad\quad\quad$ **end if**
15: $\quad\quad\quad\quad$ **if** $tran_{\tau}(P_i) > MaxTran(P_i)$ **then**
16: $\quad\quad\quad\quad\quad$ $S_{FAM}(P_i) = \{\langle \tau, tran_{\tau}(P_i)\rangle\}$
17: $\quad\quad\quad\quad\quad$ $MaxTran(P_i) = tran_{\tau}(P_i)$
18: $\quad\quad\quad\quad$ **else if** $tran_{\tau}(P_i) = MaxTran(P_i)$ **then**
19: $\quad\quad\quad\quad\quad$ Add $\langle \tau, tran_{\tau}(P_i)\rangle$ to $S_{FAM}(P_i)$
20: $\quad\quad\quad\quad$ **end if**
21: $\quad\quad\quad$ **else if** $tran_{\tau}(P_i) \leq -t_t$ **then**
22: $\quad\quad\quad\quad$ **if** $P_i \notin S_{NTP}$ **then**
23: $\quad\quad\quad\quad\quad$ Add P_i to $\mathcal{S}_{NTP}$
24: $\quad\quad\quad\quad$ **end if**
25: $\quad\quad\quad\quad$ **if** $tran_{\tau}(P_i) < MinTran(P_i)$ **then**
26: $\quad\quad\quad\quad\quad$ $S_{FDM}(P_i) = \{\langle \tau, tran_{\tau}(P_i)\rangle\}$
27: $\quad\quad\quad\quad\quad$ $MinTran(P_i) = tran_{\tau}(P_i)$
28: $\quad\quad\quad\quad$ **else if** $tran_{\tau}(P_i) = MinTran(P_i)$ **then**
29: $\quad\quad\quad\quad\quad$ Add $\langle \tau, tran_{\tau}(P_i)\rangle$ to $S_{FDM}(P_i)$
30: $\quad\quad\quad\quad$ **end if**
31: $\quad\quad\quad$ **end if**
32: $\quad\quad$ **end if**
33: $\quad$ **end for**
34: **end for**
35: **return** S_{PTP} and $S_{FAM}(P_i)$ for each $P_i \in S_{PTP}$
36: **return** S_{NTP} and $S_{FDM}(P_i)$ for each $P_i \in S_{NTP}$

There are two major phases in this algorithm. During the first phase (Step 1), all frequent itemsets along with their supports are initially derived using a standard frequent pattern generation algorithm, such as *Apriori* [1] or *FP-growth* [6], with t_s as the minimum support threshold. In the second phase (starting from Step 2 to the end), the algorithm finds all the transitional patterns and their significant milestones based on the set of frequent itemsets.

In Step 2, the support counts of all the frequent patterns on the set from the first transaction to the transaction right before the time period $\top_{\mathcal{D}}$ are collected. They are used later in computing $sup^{-}_{\tau}(P_i)$, where P_i is a frequent pattern. After the initializations from step 3 to step 7, the algorithm continues to scan the database $\mathcal{D}$ to find the time points within time period $\top_{\mathcal{D}}$. At each time point τ during the scan, it checks the frequent patterns one by one. For each frequent pattern P_i, it calculates the support of P_i on $\mathcal{D}^{-}_{\tau}$, i.e., $sup^{-}_{\tau}(P_i)$, and the support of P_i on $\mathcal{D}^{-}_{\tau}$, i.e., $sup^{+}_{\tau}(P_i)$. If both of them are greater than t_s, the algorithm then checks the transitional ratio of P_i. If the ratio is greater than t_t, then P_i is a positive transitional pattern and is added into the set $\mathcal{S}_{PTP}$. Then, the algorithm checks whether the transitional ratio of P_i is greater than the current maximal transitional ratio of P_i. If yes, the set of significant frequency-ascending milestones of P_i, i.e., $S_{FAM}(P_i)$, is set to contain $\langle \tau, tran_{\tau}(P_i)\rangle$ as its single element. If not but it is equal to the current maximal transitional ratio of P_i, $\langle \tau, tran_{\tau}(P_i)\rangle$ is added into $S_{FAM}(P_i)$. Similarly, Steps 21-30 are for finding the set of negative transitional patterns and their significant frequency-descending milestones.

If we do not consider the step for generating frequent patterns (i.e., Step 1), the *TP-mine* algorithm scans the database only once to find all the transitional patterns and their significant milestones with respect to a pattern support threshold and a transitional pattern threshold. Suppose the number of frequent patterns generated from Step 1 is *n*, the time complexity of the *TP-mine* algorithm from Step 2 to Step 36 is $\mathcal{O}(\|\mathcal{D}\| + n \times \|\top_{\mathcal{D}}\|)$, where $\|\mathcal{D}\|$ is the number of transactions in $\mathcal{D}$ and $\|\top_{\mathcal{D}}\|$ is the number of time points in $\top_{\mathcal{D}}$.

4 Experimental Studies

To demonstrate the utility of transitional patterns and the efficiency of the *TP-mine* algorithm, we have performed two sets of experiments using datasets from two real-world domains: retail market basket and web log data. Table 3 summaries the parameters of each dataset along with the threshold values used in our experiments.

Table 3. Database characteristics

Database	# Items	# Trans	# FP	# *PTP*	# *NTP*	$\top_{\mathcal{D}}$
Retail	16,470	88,163	580	22	49	[25%, 75%]
LiveLink	38,679	30,586	125	22	22	
t_s = 5‰ and t_t = 50%						

Table 4. Top 10 PTPs in Retail dataset

#	*PTP*	sup_- (‰)	sup_+ (‰)	$\langle \mathcal{M}^+, tran_{\mathcal{M}^+} \rangle$ (%)	sup (‰)
1	{12925}	5.08	32.95	⟨58.52, +84.59⟩	16.64
2	{14098}	5.03	29.72	⟨61.08, +83.07⟩	14.64
3	{39, 12925}	5.07	22.96	⟨68.88, +77.91⟩	10.64
4	{413}	6.19	26.39	⟨25.08, +76.53⟩	21.32
5	{48, 12925}	5.01	19.66	⟨70.93, +74.54⟩	9.27
6	{12929}	5.00	18.35	⟨74.41, +72.75⟩	8.42
7	{48, 413}	5.01	16.57	⟨31.94, +69.77⟩	12.87
8	{39, 413}	5.01	16.30	⟨30.81, +69.28⟩	12.82
9	{405}	5.06	15.05	⟨50.85, +66.35⟩	9.97
10	{39, 48, 413}	5.00	14.06	⟨57.39, +64.43⟩	8.86

4.1 Retail market basket data

The Retail dataset was obtained from the Frequent Itemset Mining Dataset Repository (http://fimi.cs.helsinki.fi). It contains the (anonymized) retail market basket data from an anonymous Belgian supermarket store.

Table 4 shows the first 10 positive transitional patterns in Retail. These patterns are ranked by the transitional ratios at their significant frequency-ascending milestones. For positive transitional patterns, the greater the ratio, the higher the rank; while for negative transitional patterns (shown in Table 5), the less the ratio, the higher the rank.

The first *PTP*, product R_{12925}, has a support rank of 72 in the whole Retail dataset, which is a mediocre frequent item. From its significant milestone, we notice that before the time point 58.52%, its frequency is just a little bit greater than the minimum support threshold; but its frequency increases over 6 times after the time point, which is as twice as much of its frequency in the whole Retail dataset. This unusual phenomena might be the result of a special even around the time point, such as a new advertisement or a sale promotion. In order to satisfy customers' increasing demands for product R_{12925}, the store has to take actions to enhance the supply of this product. Moreover, the supplies of products R_{39} and R_{48} need to be enhanced as well because of their co-occurrences with product R_{12925} in the 3^{rd} and 5^{th} positive transitional patterns.

As we can see from the last line of Table 4, there are 3 items R_{39}, R_{48} and R_{413} in the 10^{th} *PTP*. This pattern can be easily ignored by traditional frequent pattern mining framework since its support is relatively low (ranked 200 out of 580). However, according to this significant milestone, these products appear together more frequently after the time point 57.89%. Therefore, putting these products close to each other or starting a package promotion for these products might be very useful in selling more of these products. This idea is also backed up by the 7^{th} and 8^{th} positive transitional patterns.

Table 5. Top 10 NTPs in Retail dataset

#	*NTP*	sup_- (‰)	sup_+ (‰)	$\langle \mathcal{M}^-, tran_{\mathcal{M}^-} \rangle$ (%)	sup (‰)
1	{1327}	31.82	5.00	⟨56.90, -84.29⟩	20.26
2	{39, 1327}	25.51	5.01	⟨39.52, -80.37⟩	13.11
3	{48, 1327}	20.80	5.00	⟨37.84, -75.96⟩	10.98
4	{32, 39, 41}	45.00	13.04	⟨42.93, -71.02⟩	26.76
5	{41, 225}	17.22	5.01	⟨40.44, -70.91⟩	9.95
6	{32, 41}	60.82	17.92	⟨42.73, -70.53⟩	36.25
7	{38, 39, 41}	57.87	17.19	⟨42.81, -70.29⟩	34.61
8	{32, 39, 41, 48}	31.07	9.34	⟨42.93, -69.94⟩	18.67
9	{38, 39, 41, 48}	37.63	11.34	⟨42.78, -69.87⟩	22.58
10	{41, 65}	18.72	5.69	⟨42.97, -69.61⟩	11.29

Table 6. Top 10 PTPs in Livelink dataset

#	*PTP*	sup_- (‰)	sup_+ (‰)	$\langle \mathcal{M}^+, tran_{\mathcal{M}^+} \rangle$ (%)	sup (‰)
1	{15000}	5.03	25.12	⟨44.17, +79.97⟩	16.25
2	{1375}	5.04	22.72	⟨62.87, +77.79⟩	11.61
3	{1859}	5.54	17.92	⟨75.00, +69.10⟩	8.63
4	{8106}	5.03	15.60	⟨71.49, +67.75⟩	8.04
5	{544}	5.05	15.27	⟨56.96, +66.92⟩	9.45
6	{1381}	5.00	15.03	⟨73.24, +66.72⟩	7.68
7	{273}	5.53	16.33	⟨57.96, +66.16⟩	10.07
8	{1509}	5.03	13.92	⟨45.50, +63.87⟩	9.87
9	{545}	5.02	13.80	⟨57.36, +63.66⟩	8.76
10	{544, 545}	5.02	13.77	⟨57.98, +63.55⟩	8.70

The first 10 negative transitional patterns in Retail are listed in Table 5. The frequency of the 6^{th} negative transitional pattern is very high, ranked 20 out of 580 frequent itemsets. Its frequency is much higher, almost twice as much before the time point 42.73%; but decreases significantly afterwards. This could be the main reason why the frequencies of the 4^{th} and 8^{th} *NTP*s decrease after almost the same time since product R_{39} has the highest frequency in the Retail dataset and appears in most of the top *PTP*s. New marketing strategies should be planned for products R_{32} and R_{41}, such as a new advertisement or price dropping, to resume the sales volume for these two products and other associated products.

4.2 Livelink web log data

The Livelink dataset was first used in [7] to discover interesting association rules from Livelink web log data. This data set is not publicly available for proprietary reasons.

The top 10 positive and negative transitional patterns in the Livelink dataset are shown in Table 6 and Table 7, respectively. As we can see from the first row of Table 6, the object L_{15000} is visited most frequently after the time point 44.17%, its frequency increases about 5 times. This shows that users are very interested in the new information in L_{15000} that are updated after the specific time. Therefore, object L_{15000} should be upgraded to a higher level so that it can be more easily accessed by the users.

On the contrary, the frequency of the first negative transi-

Table 7. Top 10 NTPs in Livelink dataset

#	*NTP*	sup_- (‰)	sup_+ (‰)	$\langle \mathcal{M}^-, tran_{\mathcal{M}^-} \rangle$ (%)	sup (‰)
1	{355}	50.31	7.24	⟨40.42, -85.60⟩	24.65
2	{384}	26.56	5.01	⟨52.32, -81.15⟩	16.28
3	{11034}	18.60	5.03	⟨32.35, -72.97⟩	9.42
4	{434}	33.81	9.76	⟨59.47, -71.14⟩	24.06
5	{15001}	17.03	5.04	⟨46.84, -70.39⟩	10.66
6	{15000, 15001}	16.62	5.04	⟨46.81, -69.68⟩	10.46
7	{1735}	22.00	7.75	⟨60.78, -64.76⟩	16.41
8	{396}	14.15	5.00	⟨52.92, -64.66⟩	9.84
9	{225, 396}	13.54	5.07	⟨52.90, -62.56⟩	9.55
10	{1322}	15.69	5.96	⟨41.26, -62.03⟩	9.97

tional pattern decreased significantly from 57.31% to 7.24% after time point 40.42%. It is very obvious that the information is out-of-date or the users are not interested in it any more. Thus, this object should be moved to a corresponding lower level in order to give room to other important objects, such as L_{15000}.

Object L_{15000} is also in the 6^{th} negative transitional pattern and is frequently visited together with L_{15001} by the users before the time point 46.81%. However, after that time, the frequencies of the 5^{th} (L_{15001}) and 6^{th} negative transitional pattern decrease significantly, which means that most of the users who visit L_{15000} do not visit L_{15001} at the same time. Therefore, these two objects should be treated differently. On the other hand, object L_{544} and L_{545} should be in the same category and have links for the user to access from one to the other more easily.

5 Discussion and Conclusions

In this paper, we introduced a novel type of patterns, positive and negative transitional patterns, to represent frequent patterns whose frequency of occurrences changes significantly at some point of time in the transaction database. We also defined the concepts of significant frequency-ascending milestones and significant frequency-descending milestones to capture the time points where the frequency of patterns changes most significantly. Moreover, we develop the *TP-mine* algorithm to mine from a transaction database the set of transitional patterns along with their significant milestones.

To the best of our knowledge, the *emerging patterns* proposed in [5] is the only kind of patterns that is similar to transitional patterns. Emerging patterns are defined as itemsets whose support increase significantly from one dataset to another. When applied to time-stamped datasets, emerging patterns are used to find contrasts between two datasets with different time periods, which is separated by an unchangeable time point. Theoretically, emerging patterns can be considered as positive transitional patterns with time point set to a constant value. As we can see from the above experimental results, the significant milestones of transitional patterns can be at different places in one dataset. Thus, at a specific time point, the transitional ratio of a pattern might not reach its greatest value or even close to 0. Therefore, with a constant time point value, most of the interesting transitional patterns cannot be identified correctly.

In our experimental study, we demonstrated the usefulness of transitional patterns in two real-world domains and showed that what is revealed by the transitional patterns and their significant milestones would not be found by the standard frequent pattern mining framework. As there are concerns about the practical usefulness of data mining techniques, we hope that the research presented in this paper brings a promising avenue to look at the data from a new angle, which allows us to find new, surprising, useful and actionable patterns from data.

References

[1] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *Proc. of the 1993 ACM SIGMOD Int. Conf. on Management of Data*, pages 207–216, 1993.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. of the 20th Int. Conf. on Very Large Data Bases*, pages 487–499, 1994.

[3] R. J. Bayardo. Efficiently mining long patterns from databases. In *Proc. of the Int. ACM SIGMOD Conf.*, pages 85–93, 1998.

[4] D. Burdick, M. Calimlim, and J. Gehrke. Mafia: A maximal frequent itemset algorithm for transactional databases. In *Proc. of the 17th Int. Conf. on Data Engineering*, 2001.

[5] G. Dong and J. Li. Efficient Mining of Emerging Patterns: Discovering Trends and Differences. *Knowledge Discovery and Data Mining*, pages 43–52, 1999.

[6] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proc. of ACM-SIGMOD Int. Conf. on Management of Data*, pages 1–12, 2000.

[7] X. Huang, A. An, N. Cercone, and G. Promhouse. Discovery of interesting association rules from livelink web log data. In *Proc. of IEEE Int. Conf. on Data Mining*, 2002.

[8] J. Li and K. Ramamohanarao and G. Dong. Emerging Patterns and Classification. In *Proc. of the 6th Asian Computing Science Conf. on Advances in Computing Science*, pages 15–32, 2000.

[9] J. Pei, J. Han, and R. Mao. CLOSET: An efficient algorithm for mining frequent closed itemsets. In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 21–30, 2000.

[10] P. Tan, V. Kumar, and J. Srivastava. Indirect association: mining higher order dependencies in data. In *Proc. of the 4th European Conf. on Principles and Practice of Knowledge Discovery in Databases*, pages 632–637, 2000.

[11] Q. Wan and A. An. Efficient Mining of Indirect Associations Using *HI-Mine*. In *Proc. of the 16th Canadian Conf. on Artificial Intelligence*, page 206–221, 2003.

[12] M. J. Zaki and C. Hsiao. Charm: An efficient algorithm for closed itemset mining. In *Proc. of the 2nd SIAM Int. Conf. on Data Mining*, 2000.