Mining Productive Emerging Patterns and Their Application in Trend Prediction

Vincent Mwintieru Nofong

School of Information Technology and Mathematical Science,
University of South Australia, GPO Box 2471, Adelaide, SA 5001
Email: vincent.nofong@mymail.unisa.edu.au

Abstract

Emerging pattern mining is an important data mining task for various decision making. However, it often presents a large number of emerging patterns most of which are not useful as their emergence are due to random occurrence of items. Such emerging patterns would most often be detrimental in decision making where inherent relationships between the items of emerging patterns are relevant. Additionally, most studies on emerging pattern mining focus on mining interesting categories of emerging patterns for classification and seldom discuss their application in trend prediction. To enable mine the set of emerging patterns with inherent item relations for decision making such as trend prediction, we employ a correlation test on the items of emerging patterns and introduce the productive emerging patterns as the set of emerging patterns with inherent item relations. We subsequently propose and develop PEPs, an efficient framework for mining our proposed productive emerging patterns. We further discuss and show the possible application of emerging patterns in trend prediction. Our experimental results shows PEPs is efficient, and the productive emerging pattern set which is smaller than the set of all emerging patterns, shows potentials in trend prediction.

Keywords: Frequent Patterns, Emerging Patterns, Productiveness Measure, Trend Prediction.

1 Introduction

Emerging Patterns (EPs), the set of patterns whose frequencies increase from one dataset to another, are vital in various decision making. In static datasets such as those with classes (male vs. female, cured vs. not cured), emerging patterns can reveal useful and hidden contrast patterns between datasets to support decision making such as classifier construction (Dong and Li 1999, Li et al. 2001), disease likelihood prediction (Li et al. 2003), discovering patterns in gene expression data (Li and Wong 2001), and so on. In sequential datasets, emerging patterns are useful in decision making such as, studying and understanding customers' behaviour (Tsai and Shieh 2009), predicting future purchases (Nofong et al. 2014) and so on.

Though emerging pattern mining is an important data mining task, it is a difficult task as the downward closure property in frequent pattern mining is not applicable in emerging pattern mining (Cheng et al. 2010, Dong and Li 1999, Poezevara et al. 2011). Over the past years however, various studies have been proposed for efficient mining of emerging patterns (Dong and Li 2005, Li et al. 2003, Li and Wong 2001) and interesting emerging patterns (Fan and Ramamohanarao 2003, 2006, 2002, Li et al. 2001, Terlecki and Walczak 2007, Soulet et al. 2004). Though these works have been useful in mining emerging patterns for various decision making, they are faced with the following challenges:

- They often present a too large or a too small number of emerging patterns for decision making. Reporting a large number of emerging patterns makes it difficult to identify the set of useful ones as some might be: i.) redundant, or ii.) emerging due to random occurrence of items. Such redundant emerging patterns, or those due to random occurrence of items, would most often be detrimental in decision making where non-redundancy or inherent relationships between items of an emerging pattern are vital. On the other hand, reporting a small number of emerging patterns may result in missing some useful emerging patterns that are needed in decision making.
- They often focus on mining interesting sets of emerging patterns for classification and seldom discuss their application in trend prediction. Though emerging patterns can reveal useful emerging trends in time-stamped datasets for trend prediction, this useful application of emerging patterns is unexplored as it is hardly mentioned in existing works on emerging pattern mining.
- Though some categories of emerging patterns such as *jumping* emerging patterns (Fan and Ramamohanarao 2006, Terlecki and Walczak 2007) and *essential* emerging patterns (Fan and Ramamohanarao 2002) are very useful in classifier formation, they will not be ideal in trend prediction. This is because, per their definitions, such emerging patterns in time-stamped datasets will more likely be spikes or noise, and not emerging trends.
- Though the emerging patterns reported in (Fan and Ramamohanarao 2003) can be applicable in trend prediction, some useful emerging patterns needed in decision making might be missed. For instance, on a Twitter dataset, (Fan and Ramamohanarao 2003) misses some interesting and useful emerging hashtags such as,

"#tcot[1]#romneyryan2012", "#tcot#Obama", and "#news#Syria". The emergence of these hashtags though formed by items with inherent relationships (correlated items) and reflective of true emerging trends are missed in (Fan and Ramamohanarao 2003) because their rates of emergence are lower than those of their emerging subsets.

Motivated by the importance of emerging patterns in decision making and the aforementioned challenges in their discovery, we address these challenges as follows. We initially employ a correlation test on emerging patterns and introduce the productive emerging patterns as the set of emerging patterns with inherent item relationships. Subsequently, we propose and develop PEPs, an efficient framework for mining the set of productive emerging patterns, and show their possible application in trend prediction.

We make the following contributions to the discovery of emerging patterns.

- We propose and introduce the productive emerging pattern set as the set of emerging patterns with inherent item relationships.
- We propose and develop PEPs, an efficient productive emerging pattern mining framework.
- We show a possible application of emerging patterns in trend prediction.

In addition to these contributions, it is also worth noting that our proposed productive emerging pattern set achieves a major size reduction in the number of reported emerging patterns.

2 Related Works

The concept to emerging pattern mining was introduced by Dong and Li in (Dong and Li 1999) where they proposed an emerging pattern detection technique for static datasets with classes. They referred to an emerging pattern as an itemset whose support increases significantly from one dataset to another. More specifically, they defined an emerging pattern as an itemset whose growth rate is greater than a given threshold. Emerging pattern mining has since been researched on in works such as (Fan and Ramamohanarao 2003, 2006, 2002, Garcia-Borroto et al. 2014, Li et al. 2003, 2001, Terlecki and Walczak 2007, Soulet et al. 2004).

Over the past years however, some researchers argued that the emerging pattern definition proposed in (Dong and Li 1999) often generates too many emerging patterns making it difficult identifying the set of interesting and useful ones for decision making. Various constraints and techniques were thus proposed to enable mine interesting categories of emerging patterns. Such works include, but not limited to: jumping EPs (Fan and Ramamohanarao 2006, Li et al. 2001, Terlecki and Walczak 2007), essential EPs (Fan and Ramamohanarao 2002), and interesting EPs (Fan and Ramamohanarao 2003, Soulet et al. 2004).

Though the above mentioned works have been useful in mining emerging patterns for various decision making, they are faced with several challenges summarized as follows. Firstly, they often report a too large or a too small emerging pattern sets for decision making. Secondly, they focus on mining interesting sets of emerging patterns for classification and seldom discuss their application in trend prediction. Thirdly, most categories of emerging patterns mined in works such as (Fan and Ramamohanarao 2006, 2002, Terlecki and Walczak 2007) though very good in classifier formation, are not applicable in trend prediction. Additionally, it is worth noting that though emerging patterns mined in works such as (Fan and Ramamohanarao 2003) are applicable in trend prediction, it often misses some useful emerging patterns needed in decision making.

Inspired by the importance of emerging patterns, the aforementioned challenges in their discovery, and their possible application in trend prediction, we focus on how to mine the set of emerging patterns with inherent item relationships and their possible application in trend prediction.

[1]The hashtag #tcot, "Top Conservatives On Twitter" provides a way for conservatives and Republicans to locate and follow the tweets of their like-minded brethren.

3 Preliminaries

The problem of frequent pattern mining and its associated notation can be given as follows. Let $\mathrm{I} = \langle i_1, i_2, ..., i_n \rangle$ be a set of literals, called items. Then, a transaction is a nonempty set of items. A pattern S is a set of transactions satisfying some conditions of measures like frequency. A pattern is of length-k if it has k items, for example, $S = \{a, b, c\}$ is a length-3 pattern.

Given a database of n transactions, $\mathbf{D} =< T_1, T_2, T_3, \ldots, T_n >$, where each T_m in $\mathbf{D}$ is identified by m called TID, the *cover* of a pattern S in $\mathbf{D}$, $cov_{\mathbf{D}}(S)$, is the set of *TIDs* of transactions that contain S. That is,

$$cov_{\mathbf{D}}(S) = \{m : T_m \in \mathbf{D} \wedge S \subseteq T_m\} \tag{1}$$

The *support* of a pattern S in $\mathbf{D}$, $sup_{\mathbf{D}}(S)$, is defined as,

$$sup_{\mathbf{D}}(S) = \frac{|cov_{\mathbf{D}}(S)|}{|\mathbf{D}|} \tag{2}$$

where $|cov_{\mathbf{D}}(S)|$ is called the *support count* of S in $\mathbf{D}$.

Frequent pattern mining is the process of discovering all patterns in a database, $\mathbf{D}$, whose frequencies are larger than or equal to a user specified minimum support (η). A pattern S in $\mathbf{D}$ is said to be productive in $\mathbf{D}$ if (Webb 2010): for all S_1, S_2 (such that, $S_1 \subset S,\ S_2 \subset S,\ S_1 \cup S_2 = S,\ S_1 \cap S_2 = \emptyset$), $sup_{\mathbf{D}}(S) > sup_{\mathbf{D}}(S_1) sup_{\mathbf{D}}(S_2)$.

3.1 Emerging Patterns

Given two datasets, $\mathbf{D_i}$ and $\mathbf{D_{i+1}}$, the growth rate of a pattern S, $GR(S)$, from $\mathbf{D_i}$ to $\mathbf{D_{i+1}}$ is defined as (Dong and Li 1999):

$$GR(S) = \frac{sup_{D_{i+1}}(S)}{sup_{D_i}(S)} \tag{3}$$

Based on the growth rate, Dong and Li in (Dong and Li 1999) introduced the concept of emerging pattern mining. Formally, they defined an emerging pattern as follows.

Definition 1 *(Dong and Li 1999) Given $\rho > 1$ as the growth-rate threshold, a pattern S is said to be a* ρ-emerging *(ρ-*EP *or simply* EP*) from $\mathbf{D_i}$ to $\mathbf{D_{i+1}}$ if $GR(S) \geq \rho$.*

For any two datasets, Definition 1 will report all patterns whose growth rates are greater than or equal to the specified growth rate threshold, ρ.

Though Definition 1 has been accepted and used in mining emerging patterns, it has the following challenges. Firstly, a large number of emerging patterns are often reported and this makes it difficult to comprehend and identify the set of useful ones for decision making. Secondly, the emergence threshold ρ largely determines the number of discovered emerging patterns. If ρ is set low, a large set of emerging patterns will be discovered, most of which might be trivial. However, if ρ is set high, some useful emerging patterns needed in decision making will be missed.

Over the past years, some researchers argued that finding all EPs above a minimum growth rate constraint as proposed in (Dong and Li 1999) often generates too many emerging patterns to be analysed. Soulet et. al. in (Soulet et al. 2004) thus proposed a condensed representation approach for mining emerging patterns based on closed patterns. Fan and Ramamohanarao in (Fan and Ramamohanarao 2003), whose work is quite similar to ours however proposed a way of selecting the set of *interesting* emerging patterns. They define an interesting emerging pattern as follows.

Definition 2 *(Fan and Ramamohanarao 2003) Given $\rho > 1$ as the growth rate threshold, a pattern S is an interesting emerging pattern from $\mathbf{D_i}$ to $\mathbf{D_{i+1}}$ if:*

1. *S is frequent in both $\mathbf{D_i}$ and $\mathbf{D_{i+1}}$,*
2. *$GR(S) \geq \rho$,*
3. *$\forall Y \subset S, GR(Y) < GR(S)$, and,*
4. *$|S| = 1$, or $|S| > 1$ and $\forall Y \subset S$ such that $|Y|=|S|$-1, then, $\chi^2[sup_{\mathbf{D_i}}(S), sup_{\mathbf{D_{i+1}}}(S), sup_{\mathbf{D_i}}(Y), sup_{\mathbf{D_{i+1}}}(Y)] \geq \eta$.*

In Definition 2, the authors aimed at identifying the set of emerging patterns that:

- Cover both datasets - Condition 1.
- Have sharp discriminating powers - Condition 2.
- Are not subsumed by their emerging subsets - Condition 3.
- Have significantly different supports from their immediate subsets to ensure the items of an emerging pattern are correlated - Condition 4.

Though Definition 2 will report a set of emerging patterns as interesting, some useful emerging patterns which capture and reflect vital contrasts or emerging trends will be missed for the following reasons:

1. When ρ is set high in Condition 2: Similar as in Definition 1, when ρ is set high, some useful emerging patterns with inherent item relationships whose rates of emergence are lower than ρ will be missed.
2. The subsumption rule in Condition 3: Some useful emerging patterns whose rates of emergence are lower than their emerging subsets will be missed due to the subsumption rule in Condition 3. For instance, in a Twitter dataset[2] for the month of November 2012, when we set $\varepsilon = 0.04\%$, $\rho = 1.0$ and $\eta = 0.0$, some emerging hashtags which have inherent item relationships (correlated items), such as; #tcot#romneyryan2012, #tcot#Obama, and #news#Syria, reflecting important emerging trends from 1^{st} to 2^{nd}, and from 2^{nd} to 3^{rd} of November, were missed by Definition 2. These emerging hashtags were missed as their growth rates are less than those of their emerging subsets, #tcot, #romneyryan2012, #Obama and #news respectively. However, the emergence of these subsets do not indicate the emergence of their supersets. That is, though the emerging hashtag #tcot could be easily associated with #romneyryan2012 in #tcot#romneyryan2012, so cannot be said of the emergence of #Obama in #tcot#Obama. Similarly, the emergence of #news does not in anyway imply that news about Syria, #news#Syria, is also emerging.
3. When $\eta \gg 0.0$ in Condition 4: Though this is aimed at finding emerging patterns with correlated items, some useful emerging patterns with correlated items will be missed when $\eta \gg 0.0$ in Condition 4. This is because some emerging patterns with correlated items might not have significantly different supports from their immediate subsets. Such emerging patterns which could be useful in decision making will thus be missed when $\eta \gg 0.0$ in Condition 4.

[2]Obtained from CNetS (`http://carl.cs.indiana.edu/data/`).

4 Problem Statement and Definitions

With Definitions 1 and 2, though the set of emerging patterns and interesting emerging patterns can be identified, as mentioned in Sections 1, 2 and 3, some of these reported emerging patterns may be emerging due to random occurrence of items or some emerging patterns with inherent item relationships needed in decision making will be missed. To avoid these situations, we begin by defining an emerging pattern as follows.

Definition 3 *Given ε as the minimum support, a pattern S is an emerging pattern from $\mathbf{D_i}$ to $\mathbf{D_{i+1}}$ if it is frequent in both $\mathbf{D_i}$ and $\mathbf{D_{i+1}}$ and $GR(S) > 1.0$.*

For any two datasets, Definition 3 will detect and report all frequent patterns whose growth rates are greater than 1.0. Definition 3 requiring emerging patterns to have a growth rate greater than 1.0 eliminates the situation in Definitions 1 and 2 where ρ largely controls the number of reported emerging patterns. Also, the minimum support threshold like in Definition 2, ensures only frequent patterns that are emerging are reported. However, given two datasets and the same minimum support with $\rho = 1.0$ and $\eta = 0.0$, though Definition 2 will report a smaller set of emerging patterns, it will miss some vital emerging patterns having inherent item relationships. Definition 3 will however report all such emerging patterns.

Though Definition 3 will not miss emerging patterns with inherent item relationships, some reported emerging patterns in Definition 3 might be emerging due to random occurrence of items. In decision making where inherent relationships among items of an emerging pattern are vital, emerging patterns that are emerging due to random occurrence of items could be detrimental. This is because such emerging patterns which do not encode inherent item relationships will more likely be spikes or noise, and not emerging trends.

To enable detect and report only emerging patterns with inherent item relationships for decision making, we test for positive correlations among all items of an emerging pattern. We employ a productiveness measure proposed in (Webb 2010) for this

test and refer to emerging patterns with inherent item relationships as productive emerging patterns. Formally we define a productive emerging pattern as follows.

Definition 4 *An emerging pattern, S, from $\mathbf{D_i}$ to $\mathbf{D_{i+1}}$, is a productive EP if, $\forall S_1, S_2$ (such that, $S_1 \subset S \ \wedge S_2 \subset S \ \wedge S_1 \cup S_2 = S \ \wedge S_1 \cap S_2 = \phi$) then $sup_{D_i}(S) > sup_{D_i}(S_1)sup_{D_i}(S_2)$ and $sup_{D_{i+1}}(S) > sup_{D_{i+1}}(S_1)sup_{D_{i+1}}(S_2)$.*

Definition 4 implies an emerging pattern is productive if and only if every subset that can be formed from it have inherent item relationships in both $\mathbf{D_i}$ and $\mathbf{D_{i+1}}$. This productiveness measure for every subset is to ensure all items of an emerging pattern encode inherent relationships and not due to random occurrences. This measure in Definition 4 covers the case where an emerging pattern has more than two subsets of items that are independent of one another (Webb 2010). Since the supersets of a non-productive pattern will always contain the non-productive pattern, we use this productiveness measure as one of our main pruning strategies in PEPs to avoid reporting emerging patterns with non-productive subsets. In the rest of our work, we represent the set of productive emerging patterns from $\mathbf{D_i}$ to $\mathbf{D_{i+1}}$ as pE_i^{i+1}.

With Definition 4, our emerging pattern mining problem can now be defined as the process of mining all productive emerging patterns from dataset, $\mathbf{D_i}$ to $\mathbf{D_{i+1}}$, given a minimum support ε, and how they can be employed in trend prediction.

5 Productive Emerging Pattern Mining and Their Application in Trend Prediction

In this section, we firstly discuss and introduce PEPs, our Productive Emerging Pattern mining framework. We follow up with a discussion on how the detected productive emerging patterns can be applied in trend prediction.

5.1 Productive Emerging Pattern Mining

To efficiently mine the set of productive emerging patterns, we propose PEPs, an efficient productive emerging pattern mining framework shown in Algorithm 1. PEPs employs the Apriori-like candidate

Algorithm 1: PEPs($D_i, D_{i+1}, \varepsilon$)

Input: D_i, D_{i+1}, minimum support ε
Output: Productive EP set, pE_i^{i+1}

1 Create set $pE_i^{i+1} = \emptyset$
2 ScanData(D_i, ε) to return F_i
3 ScanData(D_{i+1}, ε) to return F_{i+1}
4 Create set L
5 **for *each*** *item* $a_y \in F_i$ **do**
6 **if** $a_y \in F_{i+1}$ **then**
7 Let $(a_y, cov_{D_i}(a_y)) = F_i(a_y)$
8 Let $(a_y, cov_{D_{i+1}}(a_y)) = F_{i+1}(a_y)$
9 Add $(a_y, cov_{D_i}(a_y), cov_{D_{i+1}}(a_y))$ to L
10 Sort L in item descending order
11 MineEPs(L, ε)
12 **return** pE_i^{i+1}

generation technique in (Agrawal and Srikant 1995). However, for any two datasets, PEPs stores the TIDs of each frequent length-1 item in both datasets to avoid repeated scanning of the datasets and for quick implementation.

PEPs employs three major steps in the productive emerging pattern mining process: i.) finding the length-1 frequent items in the two datasets, ii.) identifying the common length-1 frequent items, and iii.) mining the productive emerging patterns from the common length-1 frequent items. We discuss each step in the following sections.

5.1.1 Finding Frequent Length-1 Items

For any two datasets, $\mathbf{D_i}$ and $\mathbf{D_{i+1}}$, this step finds the set of frequent length-1 items in both datasets with regards to the minimum support using Algorithm 2 (in Lines 2 and 3 of Algorithm 1) as follows.

Algorithm 2: ScanData(D_n, ε)

Input: Dataset, D_n, minimum support, ε
Output: Frequent length-1 items, F_n

1 Create HashMap h_n
2 Create set F_n
3 **for *each*** *transaction* $T \in D_n$ **do**
4 **for *each*** *length-1 item* $a_y \in T$ **do**
5 **if** $a_y \notin h_n$ **then**
6 Create set $cov_{D_n}(a_y) = \{TID\}$
7 Add $(a_y, cov_{D_n}(a_y))$ to h_n
8 **else**
9 Let $(a_y, cov_{D_n}(a_y)) = h_n(a_y)$
10 $cov_{D_n}(a_y) = cov_{D_n}(a_y) + TID$
11 Update h_n with $(a_y, cov_{D_n}(a_y))$
12 **for *each*** *item* $a_y \in h_n$ **do**
13 Let $(a_y, cov_{D_n}(a_y)) = h_n(a_y)$
14 **if** $sup_{D_n}(a_y) \geq \varepsilon$ **then**
15 Add $(a_y, cov_{D_n}(a_y))$ to F_n
16 **return** F_n

For any dataset D_n, as shown in Lines 1 and 2 of Algorithm 2, a hashmap h_n, and the set F_n respectively, are created. From Lines 3 to 11 of Algorithm 2, for each item a_y in each transaction T of D_n, if a_y is not contained in h_n, its coverset $cov_{D_n}(a_y)$ is created and the TID of T added to $cov_{D_n}(a_y)$ in Line 6. The tuple $(a_y, cov_{D_n}(a_y))$ is then added to h_n in Line 7 of Algorithm 2. Else, if a_y is already contained in h_n, $(a_y, cov_{D_n}(a_y))$ is obtained from h_n in Line 9 as $h_n(a_y)$ and the TID of T added to $cov_{D_n}(a_y)$ in Line 10. h_n is then updated with $(a_y, cov_{D_n}(a_y))$ in Line 11.

After all items and their coversets in D_n are added to h_n, the set of frequent length-1 items in D_n are obtained from h_n from Lines 12 to 15 as follows. For each item a_y in h_n, $(a_y, cov_{D_n}(a_y))$ is obtained from h_n in Line 13 as $h_n(a_y)$. As shown in Line 14, if a_y is frequent (that is, $sup_{D_n}(a_y) \geq \varepsilon$), the tuple $(a_y, cov_{D_n}(a_y))$ is added to F_n in Line 15 of Algorithm 2. The set F_n, which contains all frequent length-1 items in D_n and their coversets is then returned in Line 16. For the two datasets, $\mathbf{D_i}$ and $\mathbf{D_{i+1}}$, ScanData(D_i, ε) and ScanData(D_{i+1}, ε) in Lines 2 and 3 of Algorithm 1 will return F_i and F_{i+1} respectively. Figure 1 illustrates the outcome of this process, that is, F_1 and F_2 from toy datasets $\mathbf{D_1}$ and $\mathbf{D_2}$ at $\varepsilon = 0.1$. The set of common frequent length-1 items in $\mathbf{D_i}$ and $\mathbf{D_{i+1}}$ are then obtained from F_i and F_{i+1}.

D_1

TID	Transaction
1	{a, e, f}
2	{c, e, f}
3	{b, c, d, e, f}
4	{c, d, f}
5	{b, e, f}
6	{a, b}
7	{b, c, d, e}
8	{c, d, e, f}
9	{b}

D_2

TID	Transaction
1	{b, e}
2	{b, c, d, e}
3	{b, c, d, f}
4	{b, c, d}
5	{b, e}
6	{b, c}
7	{c, e, f}
8	{b, c, d}
9	{c, d, e}
10	{e, f}

F_1

Pattern	cov_{D_1}
{e}	{1,2,3,5,7,8}
{b}	{3,5,6,7,9}
{f}	{1,2,3,4,5,8}
{d}	{3,4,7,8}
{a}	{1,6}
{c}	{2,3,4,7,8}

F_2

Pattern	cov_{D_2}
{f}	{3,7,10}
{d}	{2,3,4,8,9}
{c}	{2,3,4,6,7,8,9}
{e}	{1,2,5,7,9,10}
{b}	{1,2,3,4,5,6,8}

Figure 1: Set of Frequent Length-1 Items, F_1 and F_2 in $\mathbf{D_1}$ and $\mathbf{D_2}$ at $\varepsilon = 0.1$

5.1.2 Identifying Common Length-1 Frequent Items

This step (from Lines 4 to 10 of Algorithm 1) finds the set of common length-1 frequent items in $\mathbf{D_i}$ and $\mathbf{D_{i+1}}$ as follows. As shown in Line 4 of Algorithm 1, the set L to store the common length-1 frequent items and their coversets in $\mathbf{D_i}$ and $\mathbf{D_{i+1}}$ is created. From Lines 5 to 10 of Algorithm 1, the common frequent length-1 items are identified as follows. For each frequent length-1 item, a_y in F_i, if a_y is also in F_{i+1} (that is, frequent in $\mathbf{D_{i+1}}$), a_y and its coversets, $cov_{D_i}(a_y)$ and $cov_{D_{i+1}}(a_y)$, are obtained in Lines 7 and 8 as $F_i(a_y)$ and $F_{i+1}(a_y)$ respectively. The tuple $(a_y, cov_{D_i}(a_y), cov_{D_{i+1}}(a_y))$ is then added to L in Line 9. The set L is then sorted in item descending order in Line 10. The set of productive emerging patterns are then mined from L in Line 11 of Algorithm 1 by calling MineEPs(L, ε). For our running example, Figure 2 illustrates the sorted L obtained from F_1 and F_2 in Figure 1.

L

Pattern	cov_{D_1}	cov_{D_2}
{b}	{3,5,6,7,9}	{1,2,3,4,5,6,8}
{c}	{2,3,4,7,8}	{2,3,4,6,7,8,9}
{d}	{3,4,7,8}	{2,3,4,8,9}
{e}	{1,2,3,5,7,8}	{1,2,5,7,9,10}
{f}	{1,2,3,4,5,8}	{3,7,10}

Figure 2: Sorted L, The Set of Common Frequent Length-1 Items from F_1 and F_2 in Figure 1

5.1.3 Mining Productive Emerging Patterns

This step mines all productive emerging patterns from L by calling MineEPs(L, ε) (Algorithm 3) in Line 11 of Algorithm 1. Algorithm 3 mines the set of productive emerging patterns from L as follows. In Line 3 of Algorithm 3, if there are no items in L, that is $|L| = 0$, the productive emerging pattern mining terminates and the set pE_i^{i+1} returned in Line 4. Else while $|L| > 0$, the productive emerging patterns are mined from L in the nested for-loop (from Lines 6 to 23 of Algorithm 3) as follows.

In the first for-loop within L (from index $k = 0$ to $|L| - 1$), the tuple $(a_k, cov_{D_i}(a_k), cov_{D_{i+1}}(a_k))$ at the k^{th}-index is obtained in Line 8 as $L[k]$. If a_k is a length-1 item, $GR(a_k)$ is evaluated in Line 10. The tuple $(a_k, GR(a_k))$ is added to pE_i^{i+1} in Line 12 if a_k is emerging, that is, $GR(a_k) > 1.0$. While still at the k^{th}-index, the second for-loop within L (from index $l = (k + 1)$ to $|L| - 1$) starts in Line 13 as follows. Each tuple $(a_l, cov_{D_i}(a_l), cov_{D_{i+1}}(a_l))$ in the l^{th}-index is obtained in Line 14 as $L[l]$. In Line 15, if a_k and a_l have common length-$(|a_k| - 1)$ prefixes, that is, $P_{a_k}[0, |a_k|\text{-}1] = P_{a_l}[0, |a_l|\text{-}1]$, a candidate frequent pattern, S, is created in Line 16 as $S = (a_k \cup a_l, cov_{D_i}(a_k) \cap cov_{D_i}(a_l), cov_{D_{i+1}}(a_k) \cap cov_{D_{i+1}}(a_l))$.

If S is frequent and productive in both D_i and D_{i+1}, it is added to TempL in Line 18. This ensures only frequent and productive patterns are kept as they both follow the anti-monotone property. $GR(S)$ is evaluated in Line 19 and S added to pE_i^{i+1} in Line 21 if S is emerging, that is, $GR(S) > 1.0$. For each k^{th}-index in the first for-loop, the second for-loop repeats till all indexes in L are iterated in the second for-loop. When both nested for-loops are complete, L is recreated in Line 22 from TempL and the content of TempL cleared in Line 23. The size of L is checked and the nested looping repeats until $|L| = 0$.

Stage I

L during first nested looping

Pattern	cov_{D_1}	cov_{D_2}
{b}	{3,5,6,7,9}	{1,2,3,4,5,6,8}
{c}	{2,3,4,7,8}	{2,3,4,6,7,8,9}
{d}	{3,4,7,8}	{2,3,4,8,9}
{e}	{1,2,3,5,7,8}	{1,2,5,7,9,10}
{f}	{1,2,3,4,5,8}	{3,7,10}

EPs detected during first nested looping

Productive EPs
{b}
{c}
{d}
{c, d}

Stage II

L during second nested Looping

Pattern	cov_{D_1}	cov_{D_2}
{c, d}	{3,4,7,8}	{2,3,4,8,9}
{e, f}	{1,2,3,5,8}	{7,10}

No EPs detected during second nested looping

Stage III

L = {ϕ} after second nested looping. Productive EP mining process terminates as $|L| = 0$

Figure 3: Productive Emerging Pattern Mining from L (see Figure 2) at $\varepsilon = 0.1$

We illustrate the productive emerging pattern mining process in Figure 3 on L (see Figure 2) obtained from the toy transactional databases in Figure

Algorithm 3: MineEPs(L, ε)

Input: Set L, minimum support, ε.
Output: Productive EPs set, pE_i^{i+1}

```
1  Let P_{c_n}[0, b] be the the length-b prefix of c_n
2  Create set TempL = ∅
3  if |L| = 0 then
4      return pE_i^{i+1}
5  else
6      while |L| > 0 do
7          for k = 0 to |L| − 1 do
8              Let (a_k, cov_{D_i}(a_k), cov_{D_{i+1}}(a_k)) = L[k]
9              if |a_k| = 1 then
10                 Evaluate GR(a_k)
11                 if GR(a_k) > 1.0 then
12                     Add (a_k, GR(a_k)) to pE_i^{i+1}
13             for l = k + 1 to |L| − 1 do
14                 Let (a_l, cov_{D_i}(a_l), cov_{D_{i+1}}(a_l)) = L[l]
15                 if P_{a_k}[0, |a_k|-1] = P_{a_l}[0, |a_l|-1] then
16                     Create S = (a_k ∪ a_l, cov_{D_i}(a_k) ∩ cov_{D_i}(a_l), cov_{D_{i+1}}(a_k) ∩ cov_{D_{i+1}}(a_l))
17                 if S is frequent and productive in both D_i and D_{i+1} then
18                     Add S to TempL
19                     Evaluate GR(S)
20                     if GR(S) > 1.0 then
21                         Add (S, GR(S)) to pE_i^{i+1}
22         L =TempL
23         TempL.clear()
```

1 at $\varepsilon = 0.1$. As seen in Figure 3, three stages (I, II, and III) are involved in mining the productive emerging patterns from L. We discuss the processes at each stage as follows.

1. **Stage I:** This stage shows L before the first nested looping and the detected productive emerging patterns during the first nested looping. During this first nested looping within L, length-1 frequent patterns $\{b\}, \{c\}$ and $\{d\}$ are added to pE_i^{i+1} in Line 12 of Algorithm 3 since they are all emerging. Productive frequent pattern $\{c, d\}$ is also added to pE_i^{i+1} in Line 21 as it is emerging. Though patterns $\{b, c\}, \{b, d\}, \{b, e\}, \{b, f\}, \{c, e\}, \{c, f\}, \{d, e\}$ and $\{d, f\}$ generated in Line 16 during the first nested looping are frequent, they are all pruned in Line 17 for the following reasons:
   - $\{b, e\}, \{b, f\}$ and $\{d, e\}$ are non-productive in both $\mathbf{D_1}$ and $\mathbf{D_2}$.
   - $\{b, c\}$ and $\{b, d\}$ are non-productive in only $\mathbf{D_1}$.
   - $\{c, e\}, \{c, f\}$ and $\{d, f\}$ are non-productive in only $\mathbf{D_2}$.
2. **Stage II:** This stage shows L recreated from TempL after the complete first nested looping. The second nested looping repeats on L in Stage II. No productive emerging patterns are detected in this stage as no candidate length-3 pattern can be formed from $\{c, d\}$ and $\{e, f\}$ since they do not have same common prefixes.
3. **Stage III:** This stage shows L recreated from TempL after the complete second nested looping. L in this stage has no items since no length-3 patterns were generated in Stage II. The productive emerging pattern mining process terminates in this stage since $|L| = 0$.

Patterns $\{b\}, \{c\}, \{d\}$ and $\{c, d\}$ are thus reported in Line 12 of Algorithm 1 as the set of productive emerging patterns detected from $\mathbf{D_1}$ and $\mathbf{D_2}$ at $\varepsilon = 0.1$.

5.2 Employing Detected Productive Emerging Patterns in Trend Prediction

Though most categories of emerging patterns are mined for classification purposes, in this section, we investigate the possible application of our detected emerging patterns in trend prediction.

Since the supports of an emerging pattern with time can be likened to a stochastic process, we cannot directly employ linear regression in modelling and predicting the emergence of an emerging pattern. As a preliminary step towards trend prediction with emerging patterns, we employ intuition in predicting the continuous emergence and future supports of S.

To predict trends based on emerging patterns, for any consecutive datasets $D_i, D_{i+1} \ldots D_n$ with time, for instance, consecutive; daily, monthly or yearly customer purchases. We take any three consecutive datasets, for example, D_i, D_{i+1} and D_{i+2} where,

1. D_i and D_{i+1} are used as the training set. For a given minimum support (ε), we mine the set of productive emerging patterns from D_i to D_{i+1} and productive "decaying patterns" (DPs) from D_i to D_{i+1}. Our decaying patterns (DPs), from D_i to D_{i+1}, are often referred to as emerging patterns, from D_{i+1} to D_i in previous works mining emerging patterns for classification purposes.

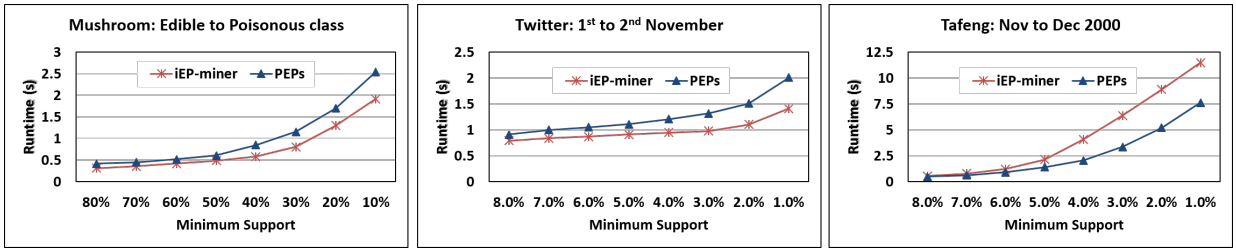

Figure 4: Emerging Pattern Detection Runtime ($\rho = 1.0, \eta = 3.841$)

Table 1: Emerging Patterns in Trend Prediction

Twitter Dataset: Trend Prediction given $\rho = 1.0, \eta = 3.841$ at $\varepsilon = 0.01$						
Days in Nov	Approach	Total EPs	Total DPs	Precision	Recall	F1-mesure
$1^{st}, 2^{nd}$ and 3^{rd}	iEP-miner	18	10	64.29	40.91	50.00
	PEPs	29	10	61.54	54.55	**57.83**
$2^{nd}, 3^{rd}$ and 4^{th}	iEP-miner	17	12	86.21	54.35	66.67
	PEPs	25	19	90.91	86.96	**88.89**
$3^{rd}, 4^{th}$ and 5^{th}	iEP-miner	29	7	33.33	26.67	**29.63**
	PEPs	39	7	26.09	26.67	26.37
$4^{th}, 5^{th}$ and 6^{th}	iEP-miner	9	21	73.33	56.41	63.77
	PEPs	9	36	82.22	94.87	**88.10**
$5^{th}, 6^{th}$ and 7^{th}	iEP-miner	4	26	96.67	72.50	82.86
	PEPs	4	35	97.44	95.00	**96.20**
$6^{th}, 7^{th}$ and 8^{th}	iEP-miner	21	10	87.10	65.85	75.00
	PEPs	30	10	90.00	87.80	**88.89**
$7^{th}, 8^{th}$ and 9^{th}	iEP-miner	27	1	39.29	22.00	28.21
	PEPs	45	1	34.78	32.00	**33.33**
$8^{th}, 9^{th}$ and 10^{th}	iEP-miner	10	21	87.10	56.25	68.35
	PEPs	15	35	92.00	95.83	**93.88**
Tafeng Retail Dataset: Trend Prediction given $\rho = 1.0, \eta = 3.841$ at $\varepsilon = 0.01$						
Months	Approach	Total EPs	Total DPs	Precision	Recall	F1-mesure
Nov, Dec and Jan	iEP-miner	55	79	80.60	48.43	60.50
	PEPs	76	151	82.82	84.30	**83.56**
Dec, Jan and Feb	iEP-miner	82	52	51.49	35.94	42.33
	PEPs	144	79	45.74	53.13	**49.16**

2. D_{i+1} and D_{i+2} are used as our test set. For the same given minimum support (ε), we mine the set of productive emerging patterns from D_{i+1} to D_{i+2} and productive decaying patterns from D_{i+1} to D_{i+2}.
3. For a detected productive emerging pattern, S_1 from the training set, we predict its presence in the test set as a productive emerging pattern, that is, $sup_{D_{i+2}}(S_1) > sup_{D_{i+1}}(S_1)$.
4. For a detected productive decaying pattern, S_2 from the training set, we predict its presence in the test set as a productive decaying, that is, $sup_{D_{i+2}}(S_2) \leq sup_{D_{i+1}}(S_2)$, or being infrequent in D_{i+2}, that is, $sup_{D_{i+2}}(S_2) < \varepsilon$.

6 Empirical Analysis

For our experimental analysis, the following implementations are compared.

1. PEPs: This is an implementation of our proposed productive emerging pattern detection framework. For any two given datasets, PEPs detects and reports all frequent and productive emerging and decaying patterns.
2. iEP-miner: This is our implementation of the method proposed in (Fan and Ramamohanarao 2003). For any two given datasets, iEP-miner detects and reports all interesting emerging and decaying patterns.

We compared the performance of PEPs and iEP-miner on: (i.) runtime and (ii.) trend prediction effectiveness with detected EPs. All methods are implemented in Java and experiments carried on a 64-bit Windows 7 PC (Intel Core i5, CPU 2.50GHz, 4GB Memory). The following datasets were used in our experimental analysis:

1. Mushroom datasets: We obtained this dataset from `http://cgi.csc.liv.ac.uk/~frans/KDD/Software/LUCS-KDD-DN/DataSets`.
2. Twitter Dataset: This dataset consists of daily hashtags of tweets for the month of November 2012. We obtained this data from CNetS (`http://carl.cs.indiana.edu/data/`).
3. Tafeng Retail Dataset: This dataset, obtained from AIIA Lab (`http://aiia.iis.sinica.edu.tw`) comprises of four months of customer transactions from TaFeng Warehouse. That is

customers transactions for the months of November and December 2000, and that of January and February 2001.

6.1 Runtime

Figure 4 shows the runtime of PEPs and iEP-miner. Though PEPs reports higher number of emerging patterns, its performance is comparable to that of iEP-miner which detects fewer number of emerging patterns. As shown in Figure 4, iEP-miner slightly outperforms PEPs at low minimum supports in the mushroom and Twitter dataset. This is because at low minimum supports, more emerging patterns which do not satisfy Conditions 3 and 4 of Definition 2 are pruned. Most of these pruned emerging patterns in iEP-miner are however productive, hence the slight out-performance. However, as can be seen in Figure 4 on the Tafeng retail dataset, PEPs slightly outperforms iEP-miner in the emerging pattern discovery process.

6.2 Decision Making

Table 1 shows a preliminary application of emerging patterns in trend prediction based on our intuition prediction approach described in Section 5.2. We employed the $F1$-measure as the overall goodness measure and evaluate our precision and recall as:

$$Prec = \frac{\#EPs + \#DPs\ correctly\ predicted}{\#EPs + \#DPs\ in\ category} \quad (4)$$

$$Recall = \frac{\#EPs + \#DPs\ correctly\ predicted}{\#EPs + \#DPs\ in\ test\ set} \quad (5)$$

As can be seen in Table 1, productive emerging patterns turn out as the best set for trend prediction as they have higher F1-scores compared to same predictions based on interesting emerging patterns (proposed by (Fan and Ramamohanarao 2003)).

7 Conclusions and Future Works

Productive emerging patterns are emerging patterns whose emergence from one dataset to another are due to inherent item relationships and not due to random occurrence of items. Non-productive emerging patterns, the set of emerging patterns whose emergence are due to random occurrence of items will be detrimental in decision making where inherent relationships between items of emerging patterns are relevant. We make use of a correlation test and introduce the productive emerging pattern set as the set of emerging patterns whose emergence are due to inherent item relationships. We develop PEPs, a productive emerging pattern mining framework and show a potential application of emerging patterns in trend prediction. Our experimental results show that PEPs is efficient, and the productive emerging pattern set which achieves a size reduction in the number of reported emerging patterns shows potential in trend prediction. Our future works are in two areas: i.) trend prediction, which will involve forming a more technical trend prediction model based on our detected productive emerging patterns, and, ii.) classification, where we tend to investigate on the effectiveness of our productive emerging patterns are in classifier formation.

References

Agrawal, R. and Srikant,R. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, pages 3–14, 1995.

Cheng, M. W. K., Choi, B. K. K., and Cheung, W. K. W. Hiding emerging patterns with local recoding generalization. In Zaki, M. J., Yu, J. X., Ravindran, B., and Pudi, V., editors, *Advances in Knowledge Discovery and Data Mining*, volume 6118 of *LNCS*, pages 158–170. Springer Berlin Heidelberg, 2010.

Dong, G. and Li, J. Efficient mining of emerging patterns: Discovering trends and differences. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, pages 43–52, New York, USA, 1999. ACM.

Dong, G. and Li, J. Mining border descriptions of emerging patterns from dataset pairs. *Knowledge and Information Systems*, 8(2):178–202, 2005.

Fan, H. and Ramamohanarao, K. An efficient single-scan algorithm for mining essential jumping emerging patterns for classification. In Chen, M. S., Yu, P. S., and Liu, B., editors, *Advances in Knowledge Discovery and Data Mining*, volume 2336 of *LNCS*, pages 456–462. Springer Berlin Heidelberg, 2002.

Fan, H. and Ramamohanarao, K. Efficiently mining interesting emerging patterns. In Dong, G., Tang, C., and Wang, W., editors, *Advances in Web-Age Information Management*, volume 2762 of *LNCS*, pages 189–201. Springer Berlin Heidelberg, 2003.

Fan, H. and Ramamohanarao, K. Fast discovery and the generalization of strong jumping emerging patterns for building compact and accurate classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 18(6):721–737, 2006.

Garcìa-Borroto, M., Martìnez-Trinidad, J.F., and Carrasco-Ochoa, J. A. A survey of emerging patterns for supervised classification. *Artificial Intelligence Review*, 42(4):705–721, 2014.

Li, J. and Wong, L. Emerging patterns and gene expression data. *Genome Informatics*, 12:3–13, 2001.

Li, J., Dong, G., and Ramamohanarao, K. Making use of the most expressive jumping emerging patterns for classification. *Knowledge and Information Systems*, 3(2):1–29, May 2001.

Li, J., Liu, H., Downing, J. R., Yeoh, A. E. J., and Wong, L. Simple rules underlying gene expression profiles of more than six subtypes of acute lymphoblastic leukemia (all) patients. *Bioinformatics*, 19(1):71–78, 2003.

Li, J., Dong, G., Ramamohanarao, K., and Wong, L. Deeps: A new instance-based lazy discovery and classification system. *Machine Learning*, 54(2):99–124, 2004.

Nofong, V. M., Liu, J. and Li, J. A study on the applications of emerging sequential patterns. In Wang, H. and Sharaf, M. A., editors, *Databases Theory and Applications*, volume 8506 of *LNCS*, pages 62–73. Springer International Publishing, 2014.

Poezevara, G., Cuissart, B., and Crèmilleux, B. Extracting and summarizing the frequent emerging graph patterns from a dataset of graphs. *Journal of Intelligent Information Systems*, 37(3):333–353, 2011.

Soulet, A., Crèmilleux, B. and Rioult, F. Condensed representation of emerging patterns. In Dai, H., Srikant, R., and Zhang, C., editors, *Advances in Knowledge Discovery and Data Mining*, volume 3056 of *LNCS*, pages 127–132. Springer Berlin Heidelberg, 2004.

Terlecki, P. and Walczak, K. Jumping emerging patterns with negation in transaction databases classification and discovery. *Information Sciences*, 177 (24):5675 – 5690, 2007.

Tsai, C.Y. and Shieh, Y. C. A change detection method for sequential patterns. *Decision Support Systems*, 46(2):501 – 511, 2009. ISSN 0167-9236.

Webb. G. I. Self-sufficient itemsets: An approach to screening potentially interesting associations between items. *ACM Transactions on Knowledge Discovery from Data*, 4(1):3:1–3:20, January 2010.