Mining Dependent Items

Hansheng Lei
Dept. of Computer Science
Univ. of Texas Rio Grande Valley
Brownsville, TX 78521
Email: hansheng.lei@utrgv.edu

Yamin Hu and Wenjian Luo
Dept. of Computer Science
Univ. of Science and Tech. of China
Hefei, China 230000
Email: huym@mail.ustc.edu.cn

Cheng Chang Pan
College of Nursing
Nova Southeastern University
Fort Lauderdale, FL 33328
Email: Sam.Pan@nova.edu

***Abstract*—Association Rule (AR) mining has been studied intensively for the past two decades. Essentially, AR models the conditional probabilities of itemsets. However, AR mining generates an overwhelming number of rules which limits its capability in mining real nuggets. We re-examined the problem and propose to start mining on dependent relationships instead of conditional relationships. In contrast to AR mining, dependence mining has received much less attention in the literature. In this paper, a new model, Dependent Pattern (DP) mining is presented. DP has a solid base in classical statistics and at the same time is suitable for large scale computation with the property of downward closure. We validate the model from different perspectives using a variety of datasets. Experimental results demonstrate that DP has remarkable advantages over AR mining and other related methods. This paper serves as a proof of concept. Future work will focus on the theoretical analysis of DP's scalability.**

1. Introduction

Traditional Association Rule (AR) mining is a viable example that statistic problems such as conditional relationships can be exhaustively computed [2]. It opens doors for computation to combine with statistics. However, AR mining often faces numerous obstacles in practice: (a) it generates a huge amount of rules, even more than the number of original transactions. Post-mining on rules is often required [18]; (b) AR does not support other relations, such as negative implication [14], correlation [7] and dependence [9], and (c) the uniform form support threshold and confidence threshold in AR do not reflect the statistical significance of items [15]. For some items, 10% support is high, but for others, even 90% support is low. Setting a universal threshold will eliminate the individual difference between these items.

The motivation for this paper comes from a mining task on a dataset of an education survey [4]. We applied existing AR mining tools but it resulted in an overwhelming number of rules, which contradicts our purpose of mining. We could adjust the support and confidence level to extremely high (e.g. both 95%), but the rules began to lose practical meaning. In examining problems with AR which have also been observed by researchers in the field, we decided to develop a new plausible model that may contribute to the literature.

The confidence in AR mining is merely an estimation of conditional probability: given event A, what is the probability of B. The support threshold is to add a significance level to such probability in its context. While this model is suited for computation using algorithms such as Aprioi and its variants [10], it lacks a deep root in statistical theory. Several specific examples have been given and explained in [15]. One reason is that the AR model is "loose" to some degree, which means the requirements are not strong enough. In contrast to conditional relation, there are stronger relations in statistics such as dependence/independence and correlation. Determining dependence/indepedence is vital in solving many probability problems [3]. If we consider conditional relation one-way, then dependent and correlation relations are two-way (i.e., it implies both $A \rightarrow B$ and $B \rightarrow A$). Therefore, dependent and correlation relations are much stronger. We can expect mining dependent items will generate fewer rules under comparable settings.

There are some studies on both correlation mining [7], [13], [16] and dependent mining [9], [15], [17]. Brin et al. discussed the weakness of AR mining from the perspective of classical statistics [15]. Examples were given to illustrate the weaknesses of AR mining. They proposed to generalize association rules to correlations with significance measured by chi-squared test. Although the generalization is relatively novel, it introduces a new problem: the items in each basket must be made complete by incorporating negative items. For example, if a transaction has only two items, i_1 and i_2, then this transaction must be represented as $\{i_1, i_2, \overline{i_3}, \cdots, \overline{i_k}\}$ in order to calculate the chi-square measure. Making up absent items increases its computation cost.

It has been observed that frequent pattern mining overlooks the interests of some items that are infrequent but interesting [13]. As the first step in AR mining is to locate frequent patterns, some interesting patterns are unfortunately filtered. A frequent pattern does not necessarily mean an interesting pattern. Therefore, mining goals depend on where the interests are and how to model the interests. Sheng et al. proposed to mine mutually dependent patterns. The goal was to define and a m-pattern,i.e., all the items in a subset with m items are mutually dependent. It models the significance of dependency by two conditional probabilities: $P(A|B) > minp$ and $P(B|A) > minp$, where $minp$

TABLE 1. NOTATION

Symbol	Description
I	a set n elements $\{i_1, i_2, ..., i_n\}$
P	dependent pattern all its subsets are dependent.
s	support, i.e., $\frac{\#ofoccurance}{N}$
$s_0(i_k)$	initial support threshold for item $i_k (k = 1, ..., n)$
α	dependence control factor

is a threshold. Essentially, it used double confidence c as selection criteria. While this method can eliminate the need of a support threshold, yet it is still within the framework of support-confidence and the same results can be obtained through post-mining on AR rules.

There is some other remarkable work on dependent item mining. Teng et al. [17] described an algorithm for mining substitution rules, which are derived from *concrete* itemsets (statistically dependent, determined by chi-square test). The algorithm would have to make up negative items for absent items in every transaction. Roy et al. proposed an algorithm to mine the top-k pairs of correlated items [12]. However, the correlation defined was only for pairs of single items. The Pearson's correlation is usually for two variables. Hence, it has limitations when applying to multiple items.

In this paper, we propose a novel method for mining dependent items. Our model is fundamentally different from existing methods in the literature. Starting for the definition of dependency of two single items, we extend it to multiple items. Exploiting the downward closure property of the definition, we present a level-wise algorithm similar as traditional AR, which is easy to implement.

The rest of the paper is organized as follows. In Section 2, we define the concepts and model the mining problem. In Section 3, the dependence mining (DP) algorithm using pseudocode is described. Then, we report our experiments conducted to validate the algorithms in Section 4. Finally, we conclude and describe future work in Section 5.

2. Definitions of Dependent Pattern

Instead of frequent pattern, we define *dependent pattern* to model the dependent relationship in a way suitable for large scale computation. To assist our description, we list the notations in Table 1. A dependent pattern P is defined as follows:

- If P consists of one single item i_k, it is a dependent pattern if and only if $s(i_k) > (1 + \alpha)s_0(i_k)$, where $s_0(i_k)$ is an initial support threshold. Note that the support thresholds $s_0(.)$ are for every $i_k (k = 1, ..., n)$ and can be different.
- If P consists of multiple items, it is a dependent pattern if and only if i) all its subsets are dependent patterns, and ii)any two subsets P_i and P_j, if $P_i \cup P_j = P$, then $s(P) > (1 + \alpha)^2 s(P_i) * s(P_j)$, where α is the dependence control factor.

The initial setting $s_0(i_k)$ for each item can be considered as an initial support threshold. That means, under this model, we can set different thresholds instead of a universal threshold. This is meaningful because not all items are born equal. To our best knowledge, all current mining models in the literature uses universal threshold(s), which overlooks the personality of individuals. In practice, all items are different and it can be problematic to treat them with a same threshold. It is particular the case in the survey data [8]. For example, a question on survey, are you male or female?, the answer (item) has a common sense probability of 50%. If the support turns out to be significantly over 50%, then it is an interesting pattern by itself. In another question, are you freshmen, sophomore, junior or senior?, each possible answer (corresponding to one item) has a common sense probability of 25%. Though, common sense does not serve our best interests.

To measure the degree of interests, a parameter α is introduced to the model. α here is essentially a degree to which the support exceeds common sense. In the applications, we can vary α to mine interesting results. And, the requirements of dependent pattern is much stronger than frequent pattern. Consequently, we can expect the range of rules based on dependent patterns will be much more focused on interesting itemsets.

Note that by definition, the dependent pattern implies mutual dependency among items. When there is one single item, the pattern is essentially a surprise pattern with a support greater than expected. When there are two items, it is the same as dependence in statistics. However, in case of more than two items, there is no equivalent concept from statistics. It is well known than chi-squared test is to decide whether a group of k ($k \geq 2$) variables is k-way independent [6], [11]. If the test rejects the hypothesis of k-way independency, it indicates that some variables among the group are dependent, but not necessarily mutual dependent. Therefore, chi-squared does not apply in our model. However, we can use chi-squared to validate our mining results, i.e., how significant a dependent pattern is.

Moreover, we should be clear that correlation is different from dependent. Correlation is usually linear and it means two variables affect each other. It is a specific type of dependence, as dependence can be non-linear. In this sense, correlation based mining has its limitations and is conceptually different from our mutual dependency. Our mining task here is to mine the dependent patterns. As far as how exactly the items are depending on each other (linear or non-linear), it is beyond the scope of the this paper.

2.1. Properties

According to definition, dependent patter P has the following properties:

- Downward closed. If P is a dependent pattern, then all its subsets are all dependent patterns. In other word, if a subset is not dependent pattern, then none of its supersets is a dependent pattern. We can take advantage of this property to derive a level-wise algorithm, just like the classical *Apriori* [2].

- $s_0(i_k)$ provides a flexible base for the model. It can be a minimum support as traditional AR mining, or common sense (expected) probabilities for answers to survey questions, or they can be determined by sampling historical data. Such a flexibility enables our model to be applicable to a wide range of mining domains.
- α is a proper setting for dependence. According to statistics, if event A is independent of event B, then

$$Prob(P1 + P2) = Prob(P1) * Prob(P2) \quad (1)$$

In other word, $\frac{Prob(P1+P2)}{Prob(P1)*Prob(P2)} > 1$ indicates positive dependence, and $\frac{Prob(P1+P2)}{Prob(P1)*Prob(P2)} < 1$ indicates negative dependence. In practice, since the probabilities are estimated, the equation (1) does not need to hold strictly. Note that the term "$> (1+\alpha)$" in the definition implies positive dependence. We can adjust the term to "$< (1-\alpha)$" to mine negative dependence easily. As a proof of concept, this paper focuses on positive dependence.

2.2. Related work

Ma and Hellerstein presented a model to mine mutually dependent patterns, namely *m-Pattern* [9]. A nonempty itemset P is an m-Pattern with minimum mutual dependence threshold $minp$ ($0 \leq minp \leq 1$)if and only if

$$Prob(P1|P2) \geq minp \quad (2)$$

holds for any nonempty two subsets $P1$ and $P2$.

While m-Pattern seems to be similar to our algorithm DP, its mechanism is fundamentally different. m-Pattern is based on the definition of the dependence between a pair of itemsets: $P1$ and $P2$ are significantly mutual dependent with a minimum dependence threshold $minp$ iff $Prob(P1|P2) \geq minp$ and $Prob(P2|P1) \geq minp$.

m-Pattern models two conditional relationships by postulating that both significance levels must be greater than a threshold. This is an extension on classic associate rule mining. However, combining two conditional relationships does not lead to *dependence* as defined in statistics. Equation 1 is the right statistic measure for independence/dependence. Here is an example to see the difference. Suppose $Prob(P1) = 0.5$, $Prob(P2) = 0.5$, $Prob(P) = Prob(P1 + P2) = 0.25$, and $minp = 0.5$. According to the mutual dependence as defined in m-Pattern,

$$\begin{aligned} Prob(P1|P2) &= Prob(P1 + P2)/Prob(P2) \\ &= 0.25/0.5 = 0.5 \\ &\geq minp = 0.5 \end{aligned}$$

and

$$\begin{aligned} Prob(P2|P1) &= Prob(P2 + P1)/Prob(P1) \\ &= 0.25/0.5 = 0.5 \\ &\geq minp = 0.5 \end{aligned}$$

Therefore, $P1$ and $P2$ are mutually dependent with minimum threshold 0.5. On the other hand, according to statistics, $P1$ and $P2$ are perfectly independent, because

$$\frac{Prob(P)}{(Prob(P1) * Prob(P2))} = \frac{0.25}{0.5 * 0.5} = 1 \quad (3)$$

Starting from equation (2), we can find the reason why the difference can happen.

$$Prob(P2|P1) = \frac{Prob(P1 + P2)}{Prob(P2)} \geq minp \quad (4)$$

Similarly, we can have $\frac{Prob(P2+P1)}{Prob(P1)} \geq minp$. Combining the two, we get:

$$\frac{Prob(P1 + P2)}{\sqrt{Prob(P1) * Prob(P2)}} \geq minp \quad (5)$$

Obviously, equation (5) is different from equation (1). $minp$ does not properly indicate the degree of dependence. Our DP algorithm based on statistic dependence is more robust. Our α in $(1 + \alpha)$ is an appropriate measure of the statistic significance of dependence. A higher value of α indicates a stronger dependent relation.

3. Algorithm for mining dependent patterns

The Dependent Pattern (DP) mining algorithm is easy to implement. The framework can be similar to level-wise Apriori algorithm. The pseudo code is described in algorithm 1 and 2. On level 1 (one item), we can filter items whose support is below the initial support threshold. The initial thresholds are not universal but item dependent, in contrast to Apriori. Level k ($k > 1$) is built on level $k-1$. Only qualified lower level candidates can pass on to a higher level, because the requirement for subset dependence is recursive (i.e., all subsets of qualified pattern must be dependent). Following the level-wise mechanism to construct dependent patterns, we can expect the trend of pattern volumes to increase at the beginning and then decreases after some levels, like a bell shape. However, given the strict requirements imposed by initial settings for support thresholds and dependence control factor α, we can also expect the bell shape is much smoother than Apriori, i.e., less number of patterns on each level.

The time complexity of DP mining is lower than the frequent itemset mining in Aprori because of three features: (a) dependence is a bi-directional requirement, other than one-directional conditional relationship, (b) we have item dependent thresholds $s_0(i_k)$, instead of a universal support threshold, and (c) the pruning power of DP can be controlled by dependence factor α. The stronger pruning power of

TABLE 2. DATASETS USED FOR EXPERIMENTS

Datasets	Description	# of items	# of transactions
DB1	Accidents	292	5000
DB2	Survery	812	1903
DB3	Extended Bakery	50	75000

DP, the less space complexity. We will demonstrate the effectiveness of DP in the experiments.

Algorithm 1 MiningDependentPatternsWithPruning

Input: dataset I, initial thresholds s_0, α
Output: all qualified patterns $\{L_k\}$

1: $C_1 = \{\{a\}|a \in I\}$
2: Compute the qualified set $L_1 = \{v \in C_1|Support(v) > s_0(v)\}$
3: $k = 2$
4: **if** L_{k-1} is empty **then**
5: return $\{L_k\}$
6: **end if**
7: Construct the set C_k based on L_{k-1} by the downward closure property
8: Prune C_k by computing the qualified candidate set $L_k = \{v \in C_k|isDependentPattern(v) = true\}$
9: $k = k + 1$
10: go back to 4

Algorithm 2 isDependentPattern

Input: pattern P(a subset candidate), α
Output: {true, false}

1: **if** all its subset of P are dependent patterns and $support(P) > support(P1) * support(P2) * (1 + \alpha)^2$, where $P1 \cup P2 = P$ **then**
2: return true
3: **else**
4: return false
5: **end if**

TABLE 3. SAMPLE QUESTIONS FROM EDUCAUSE SURVEY (DB3)

Question and answers	Items
Which of the following best describes your class standing during the current academic year?	
(A) Freshman or first-year student	1
(B) Sophomore or second-year student	2
(C) Junior or third-year student	3
(D) Senior or fourth-year student	4
(E) Other type of undergraduate	5
(F) Not an undergraduate student	6
No answer	7
Do you own a printer?	
( )Yes	8
( )No	9
No answer	10
Thinking about the past year, please rate your institutions support for the following activities from a mobile device:	
a. Accessing library resources	
( )Service not offered for mobile device	11
( )Poor	12
( )Fair	13
( )Good	14
( )Excellent	15
No answer	16
b. Registering for courses	
( )Service not offered for mobile device	17
( )Poor	18
( )Fair	19
( )Good	20
( )Excellent	21
No answer	22

4. Experimental results

Three datasets were chosen to evaluate the DP algorithm, with comparisons with classical Apriori and m-Pattern. The summary of the datasets is listed on Table 2. DB1(Accident data) is available from [5]. For the experiments, 292 attributes were selected in DB1. DB3 (Extended Bakery) is commonly used for evaluating mining algorithms, publicly available [1].

DB2 (survey data) is obtained from EDUCAUSE [4]. The survey was administered at a southern state university in collaboration with EDUCAUSE Center for Applied Research (ECAR). ECAR developed the online survey and collected the data remotely. The survey study was primarily conducted to explore undergraduate students attitude toward technology and their academic experiences related to technology use and ownership. Table 3 shows some examples of the questionnaire in the survey. To covert the collected answers into transactions, we let each answer in a question corresponds to an item. Note that survey takers may skip some questions. We view "no answer" as an item. All the answers from one user form one transaction. The Survey has 169 quantitative questions, converted to 812 items for our experiments. There are 1903 survey respondents and thus we have such amount of transactions.

The experiments were designed to evaluate the following three aspects: (a) distribution of of patterns mined using our algorithm, compared to Apriori and m-Pattern, (b) the degree to which dependence setting α affects the distribution of patterns, and (c) the degree to which DP mining works on the Survery dataset and mines meaningful relations, which Apriori fails to do so.

To observe the distribution of mined patterns, we compared DP, Apriori and m-Pattern on DB1. As shown in figure 1, all the three algorithms have a bell shape of pattern distribution over levels. However, DP generates a much smaller number of patterns on each level. This is expected because dependence is a stronger bi-directional relationship rather than unidirectional relationship. While m-Pattern is bi-directional, it produces more patterns than DP. m-Pattern lacks the first level filtering. The only control parameter is $minp$, which is essential a support threshold instead of a dependence degree. DP has two mechanisms to control the generation of patterns: initial thresholds and dependence control factor α.

The α plays a critical role in determining the number of patters. Figure 2 shows the number of total patterns

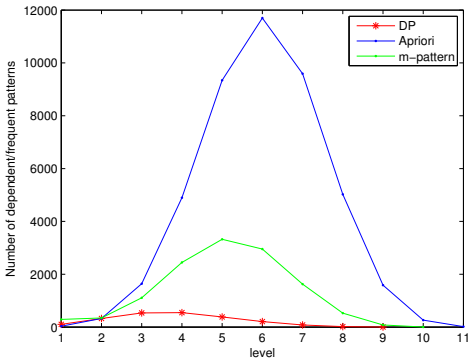

Figure 1. The number of dependent patterns mined on DB1. For DP, initial thresholds are set at 0.04 uniformly and dependence factor $\alpha = 0.2$; for Apriori, support threshold s is set at 0.4; for m-Mattern, $minp$ is set at 0.5.

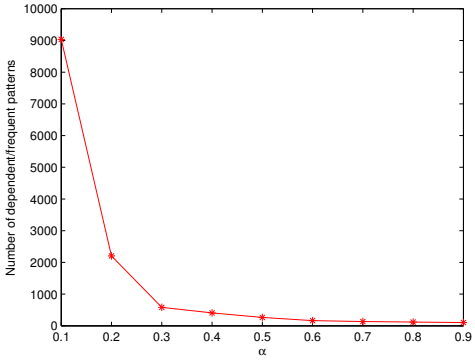

Figure 2. The total number of dependent patterns mined by DP on DB1 while α varies. Initial thresholds are uniformly set as 0.04.

mined by DP vs setting α on DB1. When α increases, the total number of patterns decreases fast. This gives us a powerful tool for mining. We can tighten α to squeeze the total number of patterns. In contrast, Apriori seems weak in controlling the number of patterns.

Table 4 shows some details of the distribution of patterns for DP algorithm, including the maximum, minimum, average and standard deviation of the real supports of the patterns mined. For example, on level 4, DP mined 548 patterns. Among those patterns, the maximum support 0.132, while the minimum support is 0.001. Table 5 shows the results by Apriori. On level 4, Apriroi mined 4896 patterns, with maximum support 0.971 and minimum support 0.40 (which is actually the threshold support). The contrast of the two tables demonstrates that Apriori is entirely controlled by a universal support threshold. The dependence patterns by DP is not restricted by support. On all the levels except level 1, the maximum support in DP is even lower than the minimum support in Apriori. That means, DP mined patterns which are filtered by Apriori in the first place. Overall, the average support of DP is much lower than Apriori, as shown in Figure 4. In Apriori, a large number of items are filtered

TABLE 4. THE PATTERN DISTRIBUTION BY DP ALGORITHM ON DB1. $s_0(\cdot) = 0.04$, $\alpha = 0.2$.

	Cardinality	Max	Min	Mean	Std
1	102	1.000	0.049	0.317	0.291
2	324	0.380	0.004	0.047	0.052
3	535	0.223	0.000	0.021	0.025
4	548	0.132	0.001	0.012	0.014
5	384	0.061	0.001	0.007	0.008
6	209	0.033	0.001	0.005	0.004
7	81	0.014	0.001	0.003	0.002
8	19	0.005	0.001	0.002	0.001
9	2	0.002	0.002	0.002	0.000

TABLE 5. THE PATTERN DISTRIBUTION BY ARIORI ON DB1. $s = 0.4$

	Cardinality	Max	Min	Mean	Std
1	32	1.000	0.411	0.704	0.190
2	327	0.997	0.400	0.598	0.153
3	1642	0.995	0.400	0.544	0.121
4	4896	0.971	0.400	0.510	0.096
5	9343	0.914	0.400	0.486	0.078
6	11700	0.832	0.400	0.468	0.063
7	9592	0.761	0.400	0.455	0.051
8	5023	0.695	0.400	0.444	0.041
9	1586	0.614	0.400	0.435	0.033
10	264	0.541	0.400	0.428	0.024
11	16	0.448	0.403	0.422	0.012

because of the support threshold in the first step. In DP, instead of support, we focus on mining the relations of dependence, even the support is not significantly high. The mean support of m-Pattern is even higher than DP, which indicates that m-Pattern is also support driven, instead of dependence driven.

While the mechanisms of the DP, Aprori and m-Pattern are all different, the mined patterns do have overlaps to a large degree. Table 6 shows overlapping mining results. When $\alpha = 0.2$, the number 2204 on cell (DP, DP) means the total mined patterns by DP is 2204. Cell (DP, Ap)=32, which means the 32 patterns are common (overlapping) among the results by DP and Ap (Apriori). Cell (DP, mP) is 124,

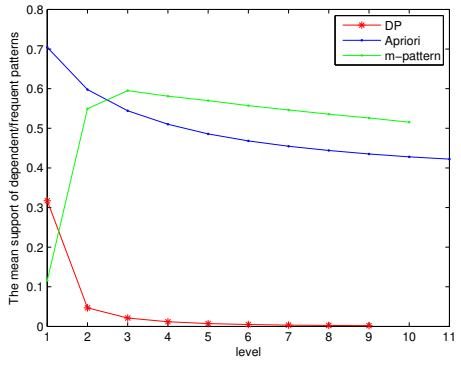

Figure 3. The mean supports of dependent/frequent patterns on DB1. For DP, initial threshold is set at 0.04 and $\alpha = 0.2$; for Apriori, $s = 0.4$; For m-Pattern, $minp = 0.5$.

TABLE 6. THE PATTERN OVERLAPPING BY DP, ARIORI AND M-PATTERN ON DB1. SETTINGS: $s = 0.4$ FOR APRIORI, $minp = 0.5$ FOR M-PATTERN, $s_0(\cdot) = 0.04$ FOR DP.

	$\alpha = 0.2$			$\alpha = 0.4$		
	DP	Ap	mP	DP	Ap	mP
DP	2204	32	124	406	32	110
Ap	32	44421	12351	32	44421	12351
mP	124	12351	12702	110	12351	12702
	$\alpha = 0.6$			$\alpha = 0.8$		
DP	164	32	99	118	32	94
Ap	32	44421	12351	32	44421	12351
mP	99	12351	12702	94	12351	12702

meaning the overlapping between DP and m-Pattern is 124. As α in DP varies, the overlapping persists.

We conclude from the tables that the overlapping of (DP, mP) is larger than (DP, Ap), which indicates that m-Pattern is "closer" to DP and Aprioir to DP. This makes sense because both m-Pattern and DP try to mine dependence, even though through difference mechanisms. Another phenomenon we observe from the table is that the overlapping of (Ap, mP) is much stronger than (DP, Ap) and (DP, mP). For example, when $\alpha = 0.6$, (DP, mP)=99, but (Ap, mP)=12351. This huge difference in the latter indicates that both Ap and mP are support driven and their working mechanisms are rather similar. Finally, the overlapping tells us that each algorithm has its own mining emphasis. It will be difficult to conclude which algorithm is superior to another.

To demonstrate how DP can mine meaningful relations, the Suvery dataset DB2 was used as the testbed. Both DP and Apriori are adjusted to its best status, focusing on what relations can be found out. In DP, we adjust initial thresholds and dependence threshold α. In Apriori, support s and confidence c are set at different values, in order to achieve the best meaningful results.

Table 7 shows sample dependent relations mined by DP. Note that row 1, 2 and 3 altogether indicates a pattern consisting of 3 items which are mutually dependent. Such relation cannot be mined by Apriori with a minimum support (e.g., 80%) because the support threshold filters the 3 items in the first step. When Aprioris sets a lower threshold, the number of associated rules mined will be overwhelmingly huge. In one case, 72735 rules are mined by Apriori with $s = 80\%$ and $c = 80\%$). This makes it prohibitively expensive to locate the meaningful relations.

Table 8 shows the number of association rules mined by Apriori on DB2 by varying support threshold s and confidence threshold c. We can see that the main factor that affects the quantity of rules is indeed the support. The dilemma is that with high s, many potential items are filtered immediately, whereas with low s, too many rules are generated. Even when both s and c are set at 95%, there are still 296 rules mined. Unfortunately, all the rules are meaningless. Note that in Apriori, the rules are between items, instead of variables. When the support is 95%, all the surviving items are those "No answers" survey items(see examples in table 3).

TABLE 7. SOME DEPENDENT RELATIONS MINED BY DP ON THE SURVEY DATASET DB2. $s_0(\cdot) = 0.05$, $\alpha = 0.005$.

Item	Item	Dep.
Technology makes me feel more connected to whats going on at the college. ($\surd$) Strongly Agree	Technology makes me feel connected to professors. ($\surd$) Strongly Agree	0.112
Technology makes me feel more connected to whats going on at the college. ($\surd$) Strongly Agree	When I entered college, I was adequately prepared to use technology needed in my courses. ($\surd$) Strongly Agree	0.103
Technology makes me feel connected to professors. ($\surd$) Strongly Agree	When I entered college, I was adequately prepared to use technology needed in my courses. ($\surd$) Strongly Agree	0.094
Institutions support for checking grade from a mobile device. ($\surd$) Good	Institutions support for ordering transcripts from a mobile device. ($\surd$) Good	0.094

TABLE 8. NUMBER OF ASSOCIATION RULES MINED BY APRIORI ON DB2 WHILE SUPPORT AND CONFIDENCE VARY.

		c			
		0.80	0.85	0.90	0.95
s	0.80	72735	72319	69933	58043
	0.85	15696	15696	15556	13354
	0.90	2943	2943	2943	2891
	0.95	296	296	296	296

Finally, to evaluate the scalability of DP, DB3 was used. Figure 4 shows the runtime of DP while the size of datasets varies. The trend shows that the runtime is linearly increasing, which is a good sign, indicating the scalability of DP is excellent. Regardless, we acknowledge that the size of DB3 (75000) is still a small number. More theoretical analysis and experiments are needed to evaluate the scalability of DP on large dataset in our future work.

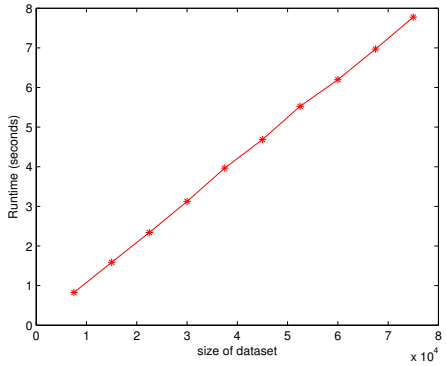

Figure 4. The runtime of DP on DB3. $s_0(\cdot) = 0.04$, and $\alpha = 0.2$.

5. Conclusion and future work

The new DP algorithm is fundamentally different from classical Associate Rule mining. While the m-Pattern algorithm is also for dependence mining, it is conceptually different from DP. This paper serves as a proof of concept. We claim that DP has remarkable advantages that other existing algorithms lack: (a) the initial support thresholds are custom-made for every item, which makes it feasible to filter unwanted items and keep potentially interesting ones, (b) the dependence control factor α is powerful in controlling the mining process starting from level two, and (c) the total number of mined patterns is controllable and thus making mining focused on dependent items, regardless of the support level.

While the proof of concept is encouraging, we are anticipating work ahead to further develop the DP algorithm. In future work, we will continue the theoretical analysis on time complexity and space complexity of DP. Particularly, we will conduct experiments on much larger datasets. We will also investigate whether the idea of FP-growth can be utilized to improve the scalability of DP.

References

[1] Extended bakery dataset. https://wiki.csc.calpoly.edu/datasets/wiki/apriori.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules, 1994.

[3] A. Agresti and M. Kateri. *Categorical data analysis*. Springer, 2011.

[4] W. J. D. D. C. Dahlstrom, E. Ecar study of undergraduate students and information technology. In *Louisville, CO: EDUCAUSE Center for Analysis and Research*, 2013.

[5] K. Geurts, G. Wets, T. Brijs, and K. Vanhoof. Profiling high frequency accident locations using association rules. In *Proceedings of the 82nd Annual Transportation Research Board, Washington DC. (USA), January 12-16*, page 18, 2003.

[6] P. E. Greenwood and M. S. Nikulin. *A guide to chi-squared testing*, volume 280. John Wiley & Sons, 1996.

[7] Y.-K. Lee, W. young Kim, Y. D. Cai, and J. Han. Comine: Efficient mining of correlated patterns. In *In Proc. 2003 Int. Conf. Data Mining*, pages 581–584, 2003.

[8] C.-H. Leu and K.-W. Tsui. Discriminant analysis of survey data. *Journal of statistical planning and inference*, 60(2):273–290, 1997.

[9] S. Ma and J. L. Hellerstein. Mining mutually dependent patterns. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 409–416. IEEE, 2001.

[10] J. Pei, J. Han, R. Mao, et al. Closet: An efficient algorithm for mining frequent closed itemsets. In *ACM SIGMOD workshop on research issues in data mining and knowledge discovery*, volume 4, pages 21–30, 2000.

[11] R. L. Plackett. Karl pearson and the chi-squared test. *International Statistical Review/Revue Internationale de Statistique*, pages 59–72, 1983.

[12] S. Roy and D. Bhattacharyya. Efficient mining of top-k strongly correlated item pairs using one pass technique. In *Advanced Computing and Communications, 2008. ADCOM 2008. 16th International Conference on*, pages 416–421. IEEE, 2008.

[13] B. Saha, M. Lazarescu, and S. Venkatesh. Infrequent item mining in multiple data streams. In *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)*, pages 569–574. IEEE, 2007.

[14] A. Savasere, E. Omiecinski, and S. Navathe. Mining for strong negative rules for statistically dependent items. In *Proceedings of*, volume 1401, pages 442–449, 2002.

[15] C. Silverstein, S. Brin, and R. Motwani. Beyond market baskets: Generalizing association rules to dependence rules. *Data Mining and Knowledge Discovery*, 2(1):39–68, 1998.

[16] G. J. Székely, M. L. Rizzo, N. K. Bakirov, et al. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794, 2007.

[17] W.-G. Teng, M.-J. Hsieh, and M.-S. Chen. On the mining of substitution rules for statistically dependent items. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 442–449. IEEE, 2002.

[18] Y. Zhao. *Post-Mining of Association Rules: Techniques for Effective Knowledge Extraction: Techniques for Effective Knowledge Extraction*. IGI Global, 2009.