IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 1, 2016, accepted September 22, 2016, date of publication November 11, 2016,
date of current version November 28, 2016.

Digital Object Identifier 10.1109/ACCESS.2016.2624418

An Effective Pattern Pruning and Summarization Method Retaining High Quality Patterns With High Area Coverage in Relational Datasets

PEI-YUAN ZHOU[1], GARY C. L. LI[2], (Member, IEEE), AND ANDREW K. C. WONG[2], (Fellow, IEEE)
[1]The Hong Kong Polytechnic University, Hung Hom, Hong Kong
[2]University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada

Corresponding author: P.-Y. Zhou(choupeiyuan@gmail.com)

ABSTRACT Pattern mining has been widely used to uncover interesting patterns from data. However, one of its main problems is that it produces too many patterns and many of them are redundant. To reduce the number of redundant patterns and retain overlapping ones, delta-closed pattern pruning was introduced, yet it can only prune subpatterns if they are covered by superpatterns. Such unduly superpatterns need to be pruned. Furthermore, in order to improve the management and interpretation of patterns, pattern summarization is proposed. It renders a small number of patterns that retain the most crucial information. *RuleCover* algorithm was one of such algorithms. However, it tends to produce over trivial patterns, whereas more interesting and revealing ones may be pruned. To overcome these problems, this paper presents a new algorithm which integrates *delta-closed*, and *RuleCover methods* with our other two new algorithms: 1) *statistically induced pattern pruning* for pruning statistically induced superpatterns by strong subpatterns and 2) *AreaCover* algorithm for pruning overlapping patterns but retain higher order and high quality patterns with large coverage of the data ''area.'' Experimental results show that the proposed algorithms produce very compact yet comprehensive knowledge from patterns discovered from relational data sets.

INDEX TERMS Pattern pruning, pattern summarization, statistical induced pattern.

I. INTRODUCTION

Pattern mining is a commonly used technique to discover patterns or rules inherent in sequences [25], data streams [24] and relational tables [9], [20]. However, it is still very difficult and time-consuming to analyze and comprehend the overwhelming number of patterns generated since a large portion of them are redundant and some are overlapping. Itemset (Pattern) pruning [11] has been proposed to prune the redundant and/or uninteresting patterns while minimizing the information loss yet the outcomes they produced are still not very satisfactory. This paper is focused on pruning redundant statistical significant patterns [22] as well as rendering a small set of crucial summarizing patterns. From here on we will use the term pattern in place of itemset.

Closed Pattern Pruning [6], [7] and *Maximal Pattern Pruning* [3]–[5] are two common pattern pruning techniques. The former is a conservative pruning strategy that retains all information of the pruned patterns. Nevertheless, the number of patterns after such pruning can still be overwhelmingly large. The latter is more aggressive, but it loses a great amount of information. To balance the tradeoff between a smaller number of retaining patterns and the least amount of information loss, *Delta Closed Pruning* (*Delta-Closed*) was proposed [9]. It takes the closed patterns and the maximal patterns as its extreme cases.

Besides pruning subpatterns by the above techniques, there are some statistically significant patterns which might also be redundant due to the way how statistical significance measure is used to define a pattern. In our earlier work [9], when statistical significance is evaluated, it is based on the deviation of the observed frequency of occurrences of a pattern from its identically and independently distributed default model, that is, the expected estimated frequency if the observed pattern is the outcome of such model. If this default model is used for pattern extraction only, there might be instances for a high-order pattern marked as statistically significant yet such outcome could be attributed to the strong statistically significant subpatterns of it. We refer such seemingly statistically significant superpattern as a statistically induced pattern [8]. Hence, whether a pattern is considered as statistical significant or not is based also on its statistical dependence on its statistically significant subpattern(s).

We therefore introduce a conditional statistical residual for such test [8], [18]. If the residual reveal independence, the statistical significance of the superpattern does not depend on that of its subpattern(s) and it should be kept, else it should be pruned. When this notion was applied to pruning sequence patterns using suffix tree data structure, Wong et al. [8], [18] reported a significant reduction of superpatterns which are indeed induced by strong subpatterns. Such approach has not been adopted to prune patterns discovered in relational dataset. In this paper, we consider both delta-closed patterns and non-statistically induced patterns in the pattern pruning process. To our knowledge, this is the first attempt to apply the pruning of statically induced superpatterns to relational datasets and obtain much superior pattern pruning results. Both delta-closed patterns and non-statistically induced patterns have been shown to successfully retain concise representations of discovered frequent and statistically significant patterns without information loss [43].

However, pattern pruning might still leave behind quite a large number of patterns. Hence, in order to further reduce the number of retained patterns as well as minimizing information loss, a pattern summarization strategy is introduced. Liu. et al presented in [44] that numerous research work had been dedicated to frequent pattern summarization which aims to obtain a much smaller set of patterns to represent the complete set of frequent patterns. Wang et al. has given the definition of pattern summarization in [43]. Given a collection of frequent patterns, pattern summarization aims to find a more concise representation such that the original collection of patterns and their support information can be reasonably recovered. In brief, pattern summarization is a more aggressive pattern pruning approach that can provide a concise representation of the discovered patterns.

In the past, RuleCover [13] has been proposed to prune patterns based on the overlapping relation of samples. It attempts to prune patterns which have been covered by other patterns. However, RuleCover often retains low-order patterns which are usually trivial patterns that do not reveal interesting or surprising information. To overcome this problem, we propose a strategy to allow patterns to cover optimally not only samples but also attributes. Therefore, a new method known as AreaCover is proposed. This method does not favour low-order patterns that tend to just cover more samples but rather considers patterns that have both high sample and attribute coverage. It thus gives a fair account to the actual coverage of the patterns in the dataset. Hence, AreaCover algorithm retains patterns with maximal area coverage rather than only sample coverage. Due to the use of a pattern summarization process after pattern pruning and AreaCover instead of using only RuleCover, our combined method is able to improve both summarization accuracy and summarization computational cost.

In summary, the contributions of this paper can be listed as follows: i) we propose a combining *Delta-Closed Pruning* and *Statistically Induced Pruning* (DCSI) algorithm to prune a large number of sub/superpatterns from original collection of frequent patterns while minimizing the information loss; ii) we incorporate AreaCover, extended from the well-known RuleCover algorithm, and produce significant improvement of the data coverage and the pattern quality of the discovered patterns; iii) by combining DCSI pruning and AreaCover, we have developed an effective, flexible and generic framework for pattern post-analysis. The experimental result shows that our integrated methodology renders much impressive results in attaining superior pattern reduction, data coverage and pattern quality when compared with its contemporary. It makes pattern discovery [20] much more useful not only in revealing statistically significant patterns but also in presenting them in a comprehensive yet more unique and compact manageable manner. We believe that it will greatly impact pattern discovery and analysis.

The rest of the paper is organized as follows: Section 2 summarizes related works in the literatures. Section 3 provides the notations and definitions. Section 4 describes the proposed DCSI and AreaCover methodology. Section 5 reports the experimental results and the performance evaluation of the proposed algorithms together with their contemporaries. Section 6 concludes the paper and outlines the future research directions.

II. RELATED WORK

Reducing the number of patterns has been a major theme in pattern mining [40]. Hence, pattern pruning and pattern summarization are two strategies proposed for pattern post-analysis to make patterns more interpretable [39] in support of discovering useful knowledge from data [13].

Pattern pruning is a common practice to reduce the number of patterns in Pattern Mining [10]–[14]. Two common pruning techniques are *Closed Itemset Pruning or Closed Pattern Pruning* (*Closed Pruning*) [6], [7] and *Maximal Itemset Pruning or Maximal Pattern Pruning* (*Maximal Pruning*) [3]–[5].

Given a *minimum support* and a*minimum confidence*, we can find all *Frequent Patterns* (*FPs*) with a higher frequency than the *minimum support.* We denote the patterns as $\{P_1, P_2, \ldots P_n\}$ which are discovered from a relational dataset D. The set of samples matched by a pattern P_i is denoted by $m(i)=\{m(i)\in D|m(i) \supseteq P_n$. Therefore, an *FP* P_i is called *maximal* if it is not a subset of any other *FP's* [6]. A FPP_i is called *closed* if none of its proper superpattern exist [6]. The advantage of *Closed Pruning* is that it is ''lossless'' and the original patterns can hence be fully recovered [3]–[5]. The *Maximal Pruning* is used to significantly reduce the number of patterns regardless of its possible information loss [3]–[5]. In summary, to ensure no information loss as the result of pattern discovery [20], *Closed Pruning* is proven to retain patterns equivalent to the original pattern set [19] while maximal pattern pruning will lose some information (patterns).

Based upon these two techniques, a number of algorithms were proposed [30]. For example, MaxMiner [31], GenMax [32] are maximal frequent itemset mining

algorithms, and CHARM [33], CLOSET+ [34], DCI_CLOSED [35] are several frequent closed itemset mining algorithms existing today. Different pruning strategies were incorporated into the above pattern mining algorithms.

Closed Pruning (CP) can remove redundant patterns effectively. However, often, when the specified closure is too restrictive, the compression rate is fairly low. To attain a good balance of the tradeoff between the smaller number of retaining patterns and the least amount of information loss, normally *Delta-Closed Pruning* is introduced [9] in between, treating *Closed* and *Maximal Pruning* as their extreme cases. In pruning redundant sequence patterns discovered, Wong et al. [8] applied the concept of *Delta-Closed Patterns* (*DCP*s) as well as statistical induced patterns [8], [18] to prune statistically significant sequence patterns and obtained very good results with high quality and representative patterns impacting a number of their later works [35], [36].

In addition, there are also a number of other optimizations of frequent patterns mining algorithms [30]. Pincer-search was proposed in [29], which provides two primary observations: 1) any subset of a frequent itemset is frequent, 2) any superset of an infrequent itemset is infrequent. Bayardo et al. [14] proposed the use of minimum improvement in confidence to prune association rules and suggested the pruning of uninteresting association rules based on certain criteria. Toivonen et al. [12] proposed an algorithm to find a subset of association rules that can cover the entire dataset. Liu et al. [11] measured the significance of rules using a chi-square test for correlation and then pruned the insignificant ones.

In addition, to further reduce the number of discovered patterns, pattern summarization which generates a comprehensive and representative summary for all discovered patterns has also been proposed. It aims at automatically selecting a small subset of patterns that are representative to other patterns [13]. In the literatures, most research works are on pattern pruning rather than pattern summarization, although these two problems are related. Pattern summarization can be considered as a very aggressive pruning method where most patterns, except for a few representative ones are pruned. The RuleCover method [13] was proposed as classical pattern summarization method, to prune a group of patterns sharing the same consequent. However, what have been retained are mostly trivial patterns. Further, in paper [41], the authors proposed the spanning set approach which defines a formal coverage criterion and selected k itemsets to represent discovered itemsets. Yang et al. [42] proposed the profile-based approach to derive itemsets based on frequency criteria from the k clusters which are formed from all frequent itemsets. Wang and Parthasarathy [43] proposed the markov random field approach to improve the estimation of the frequent itemsets. Chunyang et al. [44] proposed an approximate P-RFP mining algorithm, which effectively and efficiently compresses the set of probabilistic frequent patterns.

Hence, generally speaking, coverage criterion and frequency criterion are two key criteria and employed for evaluating the representation of itemsets or patterns [40]. In our proposed algorithm, we pruned patterns first to select the significant frequency patterns, and then summarized patterns based on data coverage.

III. NOTATIONS AND DEFINITION

In pattern mining, patterns with different orders (numbers of the variables they span) are generated. Let an N dimensional dataset D consist of M discrete valued data samples $\mathrm{v}_i = \{v_{i1}, v_{i2}, \ldots v_{iN}\}$ (where $i \epsilon M$) after discretization from the original data. Thus, D can be considered as relational dataset consisting of an attribute set $A = \{A_1, A_2, \ldots A_N\}$ consisting of N mixed-mode attributes (with a mixture of discrete and continuous values but all become categorical attributes) after the continuous values are discretized [45]. Suppose that the observed frequency of occurrences of a compound event is o, and its expected random default frequency of occurrences is e. Then the compound event can be considered as a statistically significant event, referred to as *association pattern* or *pattern* for short [10], if e deviates from o significantly.

Definition 1 (Patterns): A pattern $P_i = \{P_{il}, P_{im}, \ldots P_{in}\}$ (l, m, $n \epsilon N$) should be a subset of associating items of data samples with significant frequency of occurrences. *The* size of P_i (i.e. the number of items P_i contains) is the order of the pattern which should be at least two to make it nontrivial.

Definition 2 (Subpattern and Superpattern): A pattern P_i is a subpattern of another pattern P'_i when$P_i \subseteq P'_i$, and P'_i can be referred to as a superpattern of P_i.

Definition 3 (The Number of Occurrences of a Pattern): The *number of occurrences* of P_i denoted by K_{P_i} is the sum of the number of samples covered by P_i that occurs in the original data D.

Definition 4 (Support and Confidence): The support of a pattern P_i, denoted as $S(P_i)$, is defined as the proportion of samples in the dataset containing the pattern. The confidence of a rule that P_i implies $P_j$is defined as $C(P_i => P_j) = S(P_i \cup P_j)/S(P_i)$.

Definition 5 (Frequent Pattern): A pattern is *frequent* if the *number of occurrences* K_{P_i} of the pattern satisfies $K_{\mathrm{P}_i} \geq min_{occ}$, where min_{occ} specifies the *minimum number of occurrences* required. min_{occ} should be at least 2.

Definition 6 (Delta-Closed Pattern): Delta-Closed Patterns represents closed patterns with delta-tolerance. Given a set of frequent patterns, a delta-closed pattern P_i is one that does not have any delta closed superpattern P_j such that $K_{P_j} \geq \delta \cdot K_{\mathrm{P}_i}$, where δ is the *tolerance* factor and $0 \leq \delta \leq 1$. That is to say, P_i is not delta-closed if it has a delta-closed superpattern P_j such that $K_{P_j} \geq \delta \cdot K_{\mathrm{P}_i}$.

Definition 7 (Standard Residual): Standard residual proposed by Haberman [15] and adopted by Chan and Wong [20], [38] for pattern discovery is used to measure how the frequency K_{P_i} of a pattern P_i deviates from its expected frequency e'_{P_i}. It is defined as: $Z_{P_i} = (K_{P_i} - E_{P_i})/\sqrt{E}_{P_i}$. A pattern P_i is positively or negatively

statistically significant, if Z_{P_i} is greater or smaller than the predefined minimum threshold respectively.

A. DELTA-CLOSED PRUNING

Delta-Closed Pruning was proposed as *Closed Patterns (CPs)* with *delta-tolerance* (δ) [9], which can provide a controllable tight lossy approximation to the *CPs* [8]. To this end, a value of δ is set, and aFPP_i is called delta-closed if there exists no proper superset with $|m(i)| \geq \delta \cdot |m(j)|$. Furthermore, we must observe that among the statistically significant patterns some are considered as redundant because their statistical significance is actually induced by strong significant subpatterns. Since both Closed-Patterns *(CPs)* and their non-closed subpattern share the same level of statistical significance, it is safe to remove the non-closed patterns to finish the first stage of the pruning process [19].

B. RULECOVER SUMMARIZATION

RuleCover is an existing summarization method which evaluates the representation of patterns using a coverage criterion. It is a process that selects a small subset of representative patterns from all patterns to furnish a reasonable and explicit representation to cover as many samples from the data as possible. Thus it attempts to solve the ''too many patterns'' problem. In [13], the RuleCover method was proposed to prune a group of patterns sharing the same consequence. A set of selected rules contained in a set of samples is known as the rule cover, denoted as Δ, of those samples. A greedy algorithm finds the close-to-optimal cover. For each iteration, the algorithm selects the patterns that cover the largest number of samples and stops when all the remaining samples contain the selected patterns.

In order to specify the process of RuleCover Summarization, let us consider a relational dataset D and a set of patterns $\{P_1, P_2, \ldots P_n\}$ discovered in it. Then the set of samples matched by a pattern P_i is denoted by $m(i) = \{m(i) \in D | m(i) \supseteq P_i\}$. The RuleCover result Δ (denoting the remained patterns) is initialized as an empty set. The set u are samples not matched by the patterns in Δ whereas the sets u_i are samples in u not matched by the pattern P_i. Iteratively, all the mined patterns that match the most of the samples in u is moved to Δ. The samples matched by this rule are then removed from u. It is repeated until the patterns in Δ cover at least $\varepsilon \times 100\%$ of the samples where ε called as minimum coverage, is the only parameter specifying the minimum percentage of samples to be covered by the rule cover (Δ) [13].

RuleCover is essentially a sample-matching based method. However, low-order patterns often cover most of the samples. This would lead to a situation where RuleCover usually retains low-order patterns that cover the largest portions of samples, and mostly the lower order patterns are trivial and not that interesting nor informative. Hence, to avoid such situation, we propose AreaCover as an additional pruning phase.

IV. PROPOSED PATTERN PRUNING METHODOLOGY

In this section, we present an algorithm that incorporates the *Statistically Induced Pruning* [8] to relational datasets and then propose a new strategy that combines *Delta-Closed* and *Statistically Induced Pattern Pruning*, referred as DCSI for abbreviation. Furthermore, we integrate this pruning strategy with *AreaCover Pattern Pruning* (AreaCover) to create an effective pattern pruning algorithm to prune patterns in datasets.

A. DELTA-CLOSED AND STATISTICALLY INDUCED (DCSI) PRUNING ALGORITHM

1) STEP ONE: DISCOVERY OF DELTA CLOSED PATTERNS

In order to ensure that the patterns obtained in the pattern mining process is delta-closed, we begin with second-order patterns. We compare each of them with its superpatterns of order three. If it is covered by one of its superpattern of order three, it is not delta-closed and will be pruned. For example, we assume that a subset of the mined patterns are $\{P_1, P_2, P_3, P_4\}$ and P_1={“A”,“C”,“ ”,“D”}, P_2={“A”,“”,“ ”,“D”}, P_3={“B”,“ ”,“E”,“D”}, P_4={“B”,“ ”,“ ”,“D”} where “ ” represents an event not considered as part of the pattern. The orders of these patterns are 3, 2, 3, and 2. Let us suppose that the number of occurrences of P_1, P_2, P_3 and P_4 are supposed to be {100, 150, 180, 200} respectively. We note that P_1 is the superpattern of P_2, and P_3 is the superpatterns of P_4. When we set δ to 0.8, we prune P_4 since $180 \geq 0.8 \times 200$ implying that P_4 is not a delta-closed. However, P_2 is kept as a delta closed pattern since $100 < 0.8 \times 150$.

2) STEP TWO: DISCOVERY OF STATISTICALLY INDUCED PATTERNS

Up to this point, each of delta-closed patterns is obtained based on its deviation from the default independence model given in Definition 4. Then we need to check whether the statistical condition for considering a pattern as statistically significant is merely attributed by having a statistically strong subpattern. This question can be answered by finding whether the strong subpattern is independent with that event outside of it which forms part of its superpattern that is considered as significant based on the independence default model. If so, the superpattern by itself cannot be considered as a statistically significant pattern. It is thus a fake one which is induced by its strong subpattern and is redundant in a sense and should be pruned. Hence, we formulate a conditional standard *residual* [15] also referred to as conditional statistical significance as defined in definition 7 to check whether or not a statistically significant pattern P is induced by its subpattern(s).

Definition 8 (Conditional Statistical Significance): To check whether a pattern P_i is statistically induced by its subpattern, we first identify its proper subpattern P_j and check the*conditional statistical significance* $Z_{P_i|P_j}$ of P_i given P_j [8]

which defined as:

$$Z_{P_i|P_j} = [K_{P_i} - (pr\,(P_i|P_j) \cdot K_{P_j})]/\sqrt{pr\,(P_i|P_j) \cdot K_{P_j}}$$

Where $pr\,(P_i|P_j) \cdot K_{P_j}$ represents the expected frequency of P_i given P_j and $pr\,(P_i|P_j) \cdot K_{P_j} = [pr\,(P_i, P_j)/pr\,(P_j)] \cdot K_{P_j}$.

The *conditional statistical significance* can be used to evaluate how strong a pattern is induced by its strong subpatterns significantly. The fake significant patterns can be captured if their significance are due to their subpattern. Hence, if the value $Z_{P_i|P_j}$ is lower than a threshold, that pattern is considered as statistically induced and redundant. It should be removed. Combining both steps we gives the pseudo code as Fig. 1 of the DCSI algorithm.

Input: p^o (original statistical significant patterns)
$K_{p^{order}}$ (the frequency of occurrence of p_{order})
$order_{max}$ (max order for original patterns)
sig (threshold of the significance value)
δ (delta tolerance factor)
Output: p^o (result after pruning patterns)
Initialize: sig; δ (set by user)
Algorithm:
For iterator i = 3: $order_{max}$
  For each patterns p^i in p^o with order i
    For each patterns p^{i-1} in p^o with order i-1
      If p_{i-1} is the subpattern of p^i
        If $k_{p^{i-1}} \cdot \delta < k_{p^i}$
          delete p^{i-1} from p^o (not delta close)
        Else if $Z_{p_i|p_{i-1}} <$ sig
          delete p^i from p^o (not non-induced)
  End
 End
End
Return p^o

FIGURE 1. The Pseudo code for DCSI pruning algorithm.

It is more intuitive to represent delta-closed and non-statistical induced patterns graphically. In Fig.2 (a), the pattern-induced data of the four discovered frequent patterns are shown as four highlighted blocks. They are

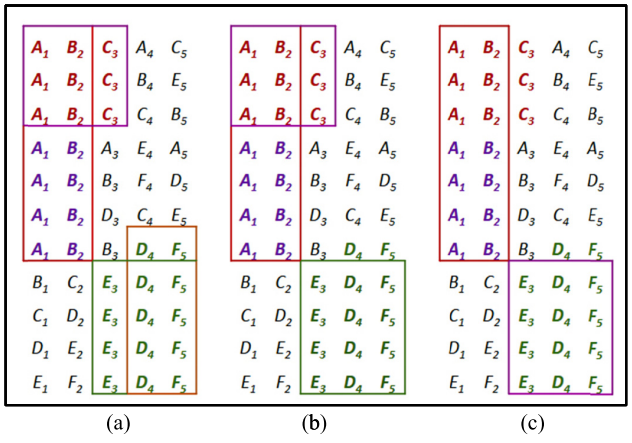


FIGURE 2. Patterns in a relational dataset. (a) Original pattern blocks, (b) Delta-closed patterns, (c) The delta-closed and non-statistical induced patterns.

$\{P_1, P_2, P_3, P_4\}$. $P_1 = \{A_1, B_2, C_3, \text{“”}, \text{“”}\}$, $P_2 = \{A_1, B_2, \text{“”}, \text{“”}, \text{“”}\}$, $P_3 = \{\text{“”}, \text{“”}, E_3, D_4, F_5\}$, and $P_4 = \{\text{“”}, \text{“”}, \text{“”}, D_4, F_5\}$ where “ ” represents an event not considered as part of the pattern. The orders of these patterns P_1, P_2, P_3 and P_4 are 3, 2, 3, and 2 and their number of occurrences of are supposed to be {3, 7, 4, 5} respectively.

We first prune the redundant subpatterns that are not delta-closed patterns. We note that P_2 is the subpattern of P_1, and P_4 is the subpattern of P_3. When we set δ to 0.8, we prune P_4 since $4 \geq 0.8 \times 5$ implying that P_4 is not a delta-closed pattern. However, P_2 is kept as a delta closed pattern since $3 < 0.8 \times 7$. This implies that if the subpattern is significant enough, although it is covered by its superpattern(s), it cannot be removed. After the first step pruning, all data are still be covered by the mined pattern. The new pattern blocks are shown as Fig. 2(b) as we notice that $P_4 = \{\text{“”}, \text{“”}, \text{“”}, D_4, F_5\}$is pruned.

Secondly, three mined patterns $P_1 = \{A_1, B_2, C_3, \text{“”}, \text{“”}\}$, $P_2 = \{A_1, B_2, \text{“”}, \text{“”}, \text{“”}\}$, $P_3 = \{\text{“”}, \text{“”}, E_3, D_4, F_5\}$ are remained after delta-closed pattern pruning. We note that P_1 is the superpattern of P_2. When we set sig = 1.96, we prune P_1 since $Z_{P_1|P_2} = 0 <$sig (1.96) implying that P_1 is induced by the strong statistics of P_2. After the second step pruning, all samples and attributes in the dataset are still be covered by the mined pattern. The new pattern blocks is shown as Fig. 2(c). Note that P_1 is pruned. The total number of patterns has been reduced from four to two without losing any statistical significant information.

B. AREACOVER ALGORITHM

We have also observed that after the removal of the above two types of redundant patterns, there are still patterns which are redundant among them due to the condition of pattern overlapping. Hence, further pattern reduction is still needed. The objective of pattern summarization is to obtain even a small subset of representative patterns. Though RuleCover algorithm has been proposed to address this issue, yet due to its characteristics as presented in the last paragraph of Section 3.B, it often retains mostly the low-order patterns that cover large portions of the samples. In practice, these patterns are usually obvious, trivial and do not embody interesting information. To overcome this problem, a new method known as AreaCover is proposed which allows the remaining discovered patterns to cover not only samples in the dataset but also attributes as well. We use the notations of all variables in Fig. 3 to describe AreaCover algorithm, and the major steps are given in Fig. 4.

AreaCover takes the original set of patterns $\Gamma = \{P_i | i = 1, \ldots n\}$ and the set of samples $I(i) = \{d \in D | d \supseteq P_i\}$ containing those samples matched by each of the patterns in Γ as input. Basically, the major difference of AreaCover from RuleCover is that the $m(i)$ in RuleCover is replaced by $I(i)$ generated from the entire set of patterns Γ since AreaCover considers both the set of matched samples and matched attributes. The AreaCover result is stored in Δ (the set of the retained patterns) which is initialized to an empty set at

Variables:

D = {1,2,…d,…d'}; original relational database

Γ= {P_i|i = 1, ... n}; discovered patterns set

I(i), i = 1,…,n; samples set matched by pattern P_i

ε :minimum coverage

Δ: remained patterns

u : samples are not matched by patterns in Δ

u_i: samples in u that are matched by the pattern P_i

FIGURE 3. Notations for AreaCover.

Initialize: ε = 80% (set by user)
Algorithm:
Δ= ϕ ;
u=I(1,…n);
For i = 1 to n
u_i = I(i)
End
Repeat
Choose $P_i \in \Gamma$ **so that area cover of** $|u_i|$ **is largest**
$\Delta = \Delta \cup P_i$,
$\Gamma = \Gamma \backslash P_i$
For each $P_i \in \Gamma$
$u_i = u_i \backslash I(i)$
if $|u_i|$ =0 **then** $\Gamma = \Gamma \backslash P_i$
End
u=u\I(i)
Until $|u| \leq |u| * (1 - \varepsilon)$
Return the AreaCover Patterns Δ

FIGURE 4. The Pseudo code of AreaCover algorithm.

the onset. The set u is used to store those samples that are not matched by the patterns in Δ whereas the set u_i stores those samples in u that are matched by the pattern P_i. Iteratively, the pattern in Γ that matches the largest area (sample × attribute) in u is moved from Γ to Δ and the samples matched by this pattern are removed from u. This is repeated until the patterns in Δ cover at least $\varepsilon \times 100\%$ of the samples where ε, referred to as the *minimum coverage*, is the only parameter used to specify the minimum percentage of samples covered by the area cover Δ.

Fig. 5 gives an illustration with seven patterns: $\Gamma = \{P_1, P_2, \ldots P_7\}$ and each pattern can cover the sample and attribute sets $I = \{I(1), I(2), \ldots I(7)\}$ where $I(i)$ is covered by P_i respectively. If RuleCover is used, it first selects P_7 since it covers the maximum number of samples*I(7)*. And then it produces either $\Delta = \{P_7, P_6\}$ which are highlighted by red color *or* $\Delta = \{P_7, P_1\}$ depending on the algorithm implementation. Obviously, it misses important patterns such as P_2 and P_3 which cover *I(2) and I(3)* respectively. However, AreaCover selects the patterns according to the percentage of the area coverage in the sequential order as P_3, P_1, P_2, P_7, P_6, $P_4|P_5$ where $P_4|P_5$ means that either P_4 or P_5 is selected since both of their expected covered areas are the same. Here, the top three patterns are highlighted by green color. Note that ε can be used to bound the number of patterns in the

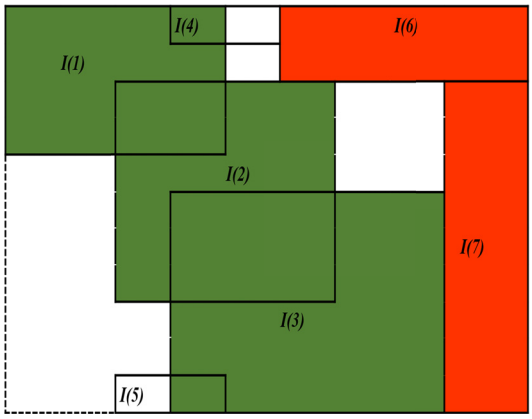


FIGURE 5. An example to compare RuleCover and AreaCover.

area cover Δ. For example, since *I(1), I(2)* and *I(3)* occupies approximately 80% of the total area of *I(1) to I(7)*, $\varepsilon = 80\%$ would produce $\Delta = \{I(3), I(1), I(2)\}$ only. Moreover, the acceptance of $I(7)$ in RuleCover would not dismiss *I(2), I(3)* and *I(5)* if AreaCover is used. Actually these two I sets, $I(7)$ and { *I(2), I(3), I(5)*}, are governed by two different sets of attributes. Hence, AreaCover is less prone to the problem of dismissing patterns within the samples being covered by other patterns since it considers matched samples as well as matched attributes.

In order to describe our work clearly for better understanding, a graphical overview of our method in is given in Fig. 6. The notations used are the same as those used in the definitions in Section 3.

C. DISCUSSION ON PARAMETER SETTING

In the above algorithm, three parameters are specified: i) delta tolerance factor $\boldsymbol{\delta}$, ii) statistical significance threshold *sig* for DCSI algorithm, and iii) minimum coverage for AreaCover algorithm.

First, the parameter δ is the sufficient fraction for a pattern to be considered as being mostly covered by its superpattern. In our opinion, a pattern P_i can be considered as non-delta-closed if it has a delta-closed superpattern P_j covering only 80 percent of its occurrence. For a good practice, $\delta = 0.8$ is a reasonable number.

Second, the statistical significance value Z_{P_i} of the pattern P_i can help us to decide if a null hypothesis can be rejected under the random model assumption [23]. Z_{P_i} is associated with a p-value which is the probability of observing the pattern P_i having at least Z_{P_i} score in discovered patterns by random model. The parameter *sig* is the threshold value of Z_{P_i} score as well. For $Z_{P_i} = 1.96$ (or nearly 2), the corresponding p-value is equal to 0.05. Hence, setting the threshold *sig=1.96* (or nearly 2) requires that the p-value of significant pattern is at most 2.5%. In statistic, it is conventional to set *sig* to be 1.96, corresponding to the p-value of 0.025.

In AreaCover algorithm, the parameter ε is used to specify the desirable minimum coverage. It is the only parameter

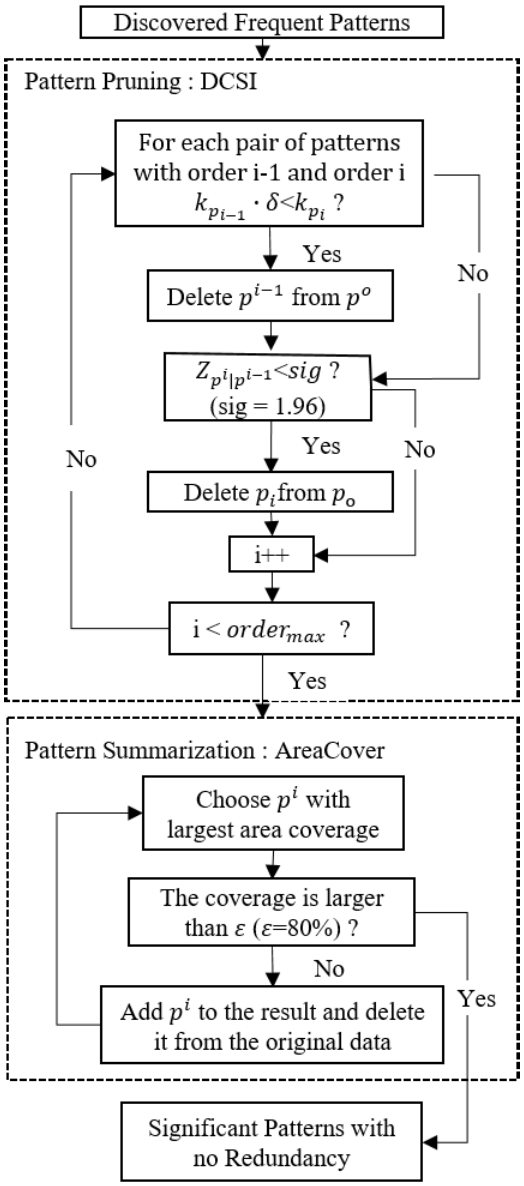


FIGURE 6. Overview of the proposed algorithms.

specifying the minimum percentage of samples covered by the RuleCover and AreaCover. In our opinion, when the remained patterns can cover 80% samples, the patterns are adequately representing the dataset. Hence, $\varepsilon = 0.8$ is a reasonable number.

D. COMPLEXITY ANALYSIS

In this section, we will show the results of the complexity analysis for DCSI and AreaCover, and compare them with existing algorithms.

Suppose that we have n frequent patterns. Delta-Closed patterns can be extracted from the frequent patterns by following procedures.

1. Sort the frequent patterns in descending order according to their length. The run-time complexity for this operation is $O(n)$.
2. Find proper superpattern for each frequent pattern in the current set of delta-closed patterns. The run-time complexity is $O(n)$. Hence the total runtime for Delta Closed method is $O(n^2)$.

After finding Delta-Closed patterns, $\sqrt{n}$ patterns are remained from original n patterns, and the average time of procedures of detection statistical induced patterns is $O(n)$. When combining the above two methods into DCSI, $O(n^2+n)$ is needed for finishing all stage of DCSI. The best case is $O(n^2)$ which is same with running Delta-Closed alone.

The run-time complexity of RuleCover reported in [13] is polynomial with respect to $\bigcup_{i=1}^{n} m(i)$. (n is the number of patterns). However for AreaCover, we add one more statement "*if* $|u_i| = 0$ *then* $\Gamma = \Gamma P_i$" after discovering the coverage pattern. The statement removes patterns P_i from Γ if the number of the uncovered matched samples $|u_i|$ is 0. This operation is only of constant time. However it could significantly speed up the AreaCover algorithm depending on how the patterns overlap. This effect is most significant in the managing patterns of lower levels (i.e. the first few runs). If there are a few large patterns in a given level, the speed of AreaCover can be significantly increased.

V. EXPERIMENTAL RESULT

This section reports three sets of experiments using eight benchmark datasets obtained from UCI repository [2] and one large dataset C-Cube [21]. To evaluate the performance of the proposed pruning algorithm, we compare the pruning result: a) between using delta-closed pruning alone and using the proposed DCSI in section 5.A and b) between using RuleCover and the proposed AreaCover in section 5.B. Finally, in section 5.C, we evaluate the pruning performance of experiments using delta-closed and statistical induced pruning with and without AreaCover.

TABLE 1. Datasets description with number of patterns previously mined with no pruning.

Dataset	Samples #	Attribute #	Patterns #
Iris	150	4	45
liver-disorder	345	6	56
Wine	178	14	2880
Glass Identification	214	10	1687
Breast Cancer	699	10	1732
Car	1728	6	50
Tic-Tac-Toe Endgame	958	10	380
Letter Recognition	4747	17	715

Before our experiments, some pre-processing and pattern mining [20] are first used to mine patterns from the datasets. Table 1 is a summarization of eight datasets obtained from UCI repository [2] and the number of patterns mined with no pruning. The datasets contain continuous data, discretized data and mixed-mode data. In addition, another large dataset for machine learning and pattern recognition, known as cursive character challenge (C-Cube) [21] with 34-dimensional

TABLE 2. Comparison results between delta-closed pruning and DCSI pruning. (a) Pruning result of delta-closed pattern pruning. (b) Pruning result for delta-closed and statistically induced (DCSI) pattern pruning.

(a)

Delta-Closed	Patterns #				Samples #		
Dataset	All Patterns	After pruned	Pruned (Sub)patterns	Pruning Rate	Original Data	After Pruned	Covered Rate
Iris	49	41	8	16.33%	120	120	100%
liver-disorder	56	38	18	32.14%	277	249	90%
Wine	2881	1785	1096	38.04%	143	143	100%
Glass	1688	747	941	55.75%	172	172	100%
Breast Cancer	1733	1296	437	25.22%	560	560	100%
Car	51	45	6	11.76%	1383	1298	94%
Tic-Tac-Toe	381	297	84	22.05%	767	767	100%
Letter	716	525	191	26.68%	3798	3794	100%
C-Cube	17959	13309	4650	25.89%	19133	19133	100%
Average	**2834.89**	**2009.22**	**825.67**	**28.21%**	**2928.11**	**2915.11**	**98.22%**

(b)

DCSI	Patterns #					Samples #		
Dataset	All Patterns	After Pruning	Pruned Subpatterns	Pruned Superpatterns	Pruning Rate	Original Data	After Pruned	Covered Rate
Iris	49	38	8	3	22.45%	120	120	100%
liver-disorder	56	32	18	6	42.86%	277	255	92%
Wine	2881	236	1096	1549	91.81%	143	142	99%
Glass	1688	138	941	609	91.82%	172	170	99%
Breast Cancer	1733	256	437	1040	85.23%	560	559	100%
Car	51	26	6	19	49.02%	1383	1298	94%
Tic-Tac-Toe	381	199	84	98	47.77%	767	767	100%
Letter	716	411	191	114	42.60%	3798	3794	100%
C-Cube	17959	10121	4650	3188	43.64%	19133	19133	100%
Average	**2834.89**	**1273**	**825.67**	**736.22**	**57.47%**	**2928.1**	**2915.3**	**98%**

feature vectors is used for evaluating the performance of the algorithms. There are total 19133 records with 34 features, and for each record a letter (a-z) is used as a label. Hence, the size of the dataset is 19133×35.

Data are first transformed into discrete events, and then mined or discovered [10], [13], [20]. The first two columns of Table 1 show the number of samples and attributes of each dataset, and the third columns shows the number of patterns mined.

A. COMPARISON BETWEEN DELTA-CLOSED AND STATISTICALLY INDUCED PRUNING

Table 2(a) and Table 2(b) summarize the pruning results on 8 real-world datasets when setting $\delta = 0.8$ for Delta-Closed algorithm and the significance threshold to 1.96 for DCSI respectively. Both tables consist of two sections: *patterns* to represent the number of patterns mined and pruned, and *induced data* to represent the number of samples covered by patterns respectively. Table 2(a) shows the number of pruned subpatterns in the third column, and Table 2(b) shows the total number of pruned subpatterns as well as superpatterns in two sections. Finally, the pruning rate in the tables is defined as

$$(all\ patterns - patterns\ pruned)/(all\ discovered\ patterns)$$

In Tables 2(b), the highest pruning rate is 92% for the Wine and Glass datasets using DCSI , which is higher than the pruning rate 38% and 56% respectively using Delta-Closed alone. On the average, DCSI reduces 57.47% of the pre-pruned patterns, much higher than 29% of that obtained by Delta-Closed alone. Although more than half of patterns are pruned using DCSI, 98% of the samples are still covered with a coverage rate only slightly lower than the 98.22% obtained by Delta-Closed.

The experimental results of individual datasets are summarized as below:

1. Iris: Although pruning is unnecessary for Iris dataset with small number of patterns, it is still good to see the reduction of 22% patterns using DCSI. The pruning rate is higher than 16% when pruning subpatterns alone. Although more patterns are pruned, 100% samples are still covered by the retained patterns both for DCSI and Delta-Closed.
2. Liver-disorder: Only 32% patterns are pruned using Delta-Closed, but 43% of sub/superpatterns are pruned by DCSI.
3. Wine: The number of patterns reduced by DCSI is more than twice of that reduced by Delta-Closed, with the pruning rate of 92% against 38% respectively.

TABLE 3. (a) Pattern summarization with RuleCover algorithm. (b) Pattern summarization with AreaCover algorithm.

(a)

RuleCover	Patterns			Sample Coverage			Attribute Coverage		
Dataset	All Patterns	After pruning	Pruning Rate	Original Data	After pruning	Covered Rate	Original Attribute	After Pruning	Covered Rate
Iris	49	4	91.84%	120	115	95.83%	5	2	40.00%
liver-disorder	56	7	87.50%	277	222	80.14%	6	5	83.33%
Wine	2881	1	99.97%	143	131	91.61%	14	2	14.29%
Glass	1688	2	99.88%	172	157	91.28%	10	3	30.00%
Breast Cancer	1733	2	99.88%	560	519	92.68%	10	3	30.00%
Car	51	5	90.20%	1383	1177	85.10%	7	4	57.14%
Tic-Tac-Toe	381	4	98.95%	767	637	83.05%	10	4	40.00%
Letter	716	5	99.30%	3798	3225	84.91%	17	6	35.29%
C-Cube	17959	5	99.97%	19133	16069	83.99%	35	10	28.57%
Average	2834	3.89	**96.39%**	2928	2427	**87.62%**	12.67	4.33	**39.85%**

(b)

AreaCover	Patterns			Sample Coverage			Attribute Coverage		
Dataset	All Patterns	After pruning	Pruning Rate	Original Data	After Pruning	Covered Rate	Original Attribute	After Pruning	Covered Rate
Iris	49	5	89.80%	120	101	84.17%	5	5	100.00%
liver-disorder	56	11	80.36%	277	229	82.67%	6	6	100.00%
Wine	2881	5	99.83%	143	137	95.80%	14	12	85.71%
Glass	1688	4	99.76%	172	146	84.88%	10	10	100.00%
Breast Cancer	1733	3	99.83%	560	452	80.71%	10	10	100.00%
Car	51	8	84.31%	1383	1272	91.97%	7	7	100.00%
Tic-Tac-Toe	381	6	98.43%	767	690	89.96%	10	9	90.00%
Letter	716	11	98.46%	3798	3372	88.78%	17	16	94.12%
C-Cube	17959	14	99.92%	19133	18421	96.28%	35	32	91.43%
Average	2834	7.4	**94.52%**	2928	2725	**88.36%**	12.67	11.89	**95.7%**

4. Glass: DCSI reduces more than 90% of the patterns while 99% of the samples are covered by the retained patterns.
5. Breast Cancer: Delta-Closed reduces 25% of the frequent patterns while DCSI reduces 85%, three times more than those reduced by of Delta-Closed.
6. Car: DCSI can reduce 49% patterns which is fourfold more than those by Delta-Closed.
7. Tic-Tac: Delta-Closed reduces 22% of patterns, while, DCSI reduces 48% patterns. Besides pruning more patterns, the remaining patterns also cover 100% of the samples.
8. Letter: Delta-Closed pruning reduces 27% of the patterns similar to all above experiments, while DCSI reduces 43% of the patterns.
9. C-Cube: Delta-Closed pruning reduce 25.9% of the redundant patterns and cover 100% of the data, while DCSI reduces 43.64% of the patterns and cover 100% of the data.

B. COMPARISON BETWEEN RULECOVER AND AREACOVER

RuleCover [19] and the proposed AreaCover are pattern summarization algorithms which are more aggressive for pruning patterns. In this section, these two coverage-based summarization algorithms are applied to the same eight datasets for evaluation. Tables 3 (a) and (b) show the comparison results when setting 80% as the minimum coverage. The average number of mined patterns is significantly reduced (from 2834 to 4) using RuleCover and (from 2834 to 7) using AreaCover. However, it is easy to use low-order patterns to obtain high coverage when RuleCover is used. The proposed AreaCover takes into consideration not only samples but also the attributes. Hence it allows higher order patterns to cover the data area rather than using the samples alone as shown in Fig 5 when comparing the red blocks alone with both the red and the green blocks of data being covered by the retaining patterns. It also explains that ours retains a higher average of number of patterns that Rule-Cover.

Table 3 shows the experimental result of RuleCover and the combined Rule and Area Cover in greater details. In the Wine dataset in Table 3(a), only one pattern is retained after pruning by RuleCover since 91.6% samples are covered by the two order pattern *{alcohol =[11.03, 14.3], alkalinity_of_ash =[15.5,30]}*. However, in AreaCover (Table 3(b)), we retain five patterns with 92% sample coverage and 86% attribute coverage compared with 92% and 14% for RuleCover respectively. On the average, Table 3 (a) shows that the percentage of sample and attribute coverage are around 88% and 39% respectively for RuleCover, whereas are around 88% and 96% respectively for AreaCover (Table 3(b)). On the whole, AreaCover usually produces good results. It reduces 2834 average number of patterns to 7.4 with average sample coverage as 88.4% and attribute coverage as 95.7%. We hence recognize that AreaCover produces more quality patterns of higher order and a higher coverage percentage on the entire data area rather than just on the samples. The higher percentage of attribute coverage by AreaCover indicates the pattern comprehensiveness and quality. In another words, if we consider the coverage of samples alone, the attribute information

TABLE 4. The number of patterns retained and the coverage of samples and attributes after pattern pruning and pattern summarization.

Dataset	Patterns #	Patterns # After DCSI & AreaCover	Pruning Rate	Coverage of Data	Coverage of Attribute
Iris	49	5	89.80%	84%	100%
Liver- disorder	56	7	87.50%	80%	100%
Wine	2881	8	99.72%	88%	93%
Glass	1688	8	99.53%	81%	100%
Breast Cancer	1733	5	99.71%	85%	100%
Car	51	7	86.27%	91%	86%
Tic-Tac-Toe	381	6	98.43%	90%	90%
Letter	716	11	98.46%	89%	94%
C-Cube	17959	15	99.92%	96.2%	91.43%
Average	2834.89	8	**95.91%**	**87.13%**	**94.94%**

may be lost when RuleCover is used. Hence, the use of AreaCover can capture more attribute information, a significant gain of pattern quality. In summarization, AreaCover renders a very small set of summarizing patterns which give adequate cover of the data and attributes area associating with them.

C. EXPERIMENTS FOR THE INTEGRATED ALGORITHM

With quality patterns retained after our DCSI Pruning, we are still able to examine minor variations of patterns. Hence after pattern pruning, we integrated DCSI and AreaCover to obtain even a much more compact pattern sets with maximal data area coverage. We refer it as summarizing patterns. In this section, we present the experimental results obtained by our proposed method that integrates DCSI and AreaCover. The proposed method consists of two stages: 1) the use of DCSI to prune redundant patterns with less information loss; and 2) the use of AreaCover to further reduce the number of patterns to obtain a much small number of patterns which will still cover a large specified percentage of the entire dataset and/or encompass a specified percentage of the attributes. We refer the second stage as Summarization since it retains a very small set of patterns yet covers a large percentage of the dataset where most of the minor variations are suppressed. Table 4 describes the number of patterns and the percentage of covered data as well as the percentage of attributes being covered after pattern pruning when the integrated algorithm is used. We set $\delta = 0.8$ and the threshold of the as 1.96 for DCSI and set the minimum coverage $\varepsilon = 0.8$ for AreaCover. The result of the first stage is the same as reported in the experimental results in section 5.A (Table 2(b)). After the second stage, the average of pruning rate for 9 real-world datasets is 95.91%. Moreover, although more than 90% patterns are pruned, the average of coverage rate for the samples is still higher than 87% for the entire data area and 95% for the number of attributes. In summary, for only pruning sub/superpatterns, 58% patterns are pruned (Table 2(b)), and after two-stage pruning, 96% patterns are pruned yet the retaining ones still cover 87% of the dataset.

VI. CONCLUSION

This paper presents a significant pattern pruning and summarization methodology to reduce the overwhelming number of patterns after pattern mining yet ensuring the retaining patterns to have maximal data area coverage minimizing the information loss. First of all, further to Delta-Closed subpattern pruning, we adapt and combine our Statistical Induced Pattern Pruning method used in our sequence pattern discovery algorithm for pruning patterns discovered from relational datasets. Our DCSI Algorithm can prune both the subpatterns covered by their superpatterns, and also superpatterns induced from strong subpatterns. Thus, it significantly impacts the pattern pruning results. Secondly, to ensure the pattern quality in keeping high order patterns and utilizing more attributes in revealing pattern association in the dataset, we develop an AreaCover algorithm to replace the RuleCover Algorithm that usually retains mostly low-order patterns which often are trivial and uninteresting when only the coverage of samples is considered. Finally, when we integrate both algorithms into one which we refer to as pattern summarization, we are able to render much better pruning rate with very large data area coverage. More importantly, we are able to retain significant high quality patterns covering large part of samples and attributes in the relational dataset. Hence, the retaining patterns are much more informative in revealing complex yet significant association in the most comprehensive and condensed form. Its impact in pattern mining is clearly supported by the superior results as given in Table 4. A comparison of the effectiveness of our proposed algorithm with that of the others from a considerable set of experiments with various types of data and problems clearly demonstrates that ours is more realistic and superior.

If we intend to examine minor variations of patterns, we could apply pattern clustering [10] on the pattern set after our DCSI Pruning. Summarization can also be applied to each pattern cluster to render a brief summary of each of them. This will be our next task. By and large, in this paper we show that our method has solved a plaguing problem for years in pattern mining since most of the existing methods still produce overwhelming amounts of patterns. It hence enables the users to have a better grasp of the knowledge embedded in data big and small. The experiments on real-world data clearly show that our proposed algorithm provides a better trade-off between the number of patterns pruned and the amount of information retained – a challenge in the era of big data and patterns.

REFERENCES

[1] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 3rd ed. San Mateo, CA, USA: Morgan Kaufmann, 2006.

[2] A. Asuncion and D. J. Newman, "UCI machine learning repository, school of information and computer science," Dept. Inf. Comput. Sci., Univ. California, Berkeley, CA, USA, Tech. Rep. 2007. [Online]. Available: http://archive.ics.uci.edu/ml/

[3] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proc. 20th Int. Conf. Very Large Data Bases*, Santiago, Chile, 1994, pp. 487–499.

[4] D. Burdick, M. Calimlim, and J. Gehrke, "MAFIA: A maximal frequent itemset algorithm for transactional databases," in *Proc. 17th Int Conf. Data Eng.*, Apr. 2001, pp. 443–452.

[5] K. Gouda and M. J. Zaki, "Efficiently mining maximal frequent itemsets," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2001, pp. 163–170.

[6] J. Pei, J. Han, and R. Mao, "Closet: An efficient algorithm for mining frequent closed itemsets," in *Proc. SIGMOD Int. Workshop Data Mining Knowl. Discovery*, 2000, pp. 21–30.

[7] M. J. Zaki and C. J. Hsiao, "Efficient algorithms for mining closed itemsets and their lattice structure," *IEEE Trans. Knowl. Data Eng.*, vol. 1, no. 4, pp. 462–478, Apr. 2005.

[8] A. K. C. Wong, D. Zhuang, G. C. L. Li, and E.-S. A. Lee, "Discovery of delta closed patterns and non-induced patterns from sequences," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 8, pp. 1408–1421, Aug. 2012.

[9] J. Cheng, Y. Ke, and W. Ng, "δ-Tolerance closed frequent itemsets," in *Proc. 6th Int. Conf. Data Mining*, 2006, pp. 139–148.

[10] A. K. C. Wong and G. C. L. Li, "Simultaneous pattern and data clustering for pattern cluster analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 7, pp. 911–923, Jul. 2008.

[11] B. Liu, W. Hsu, and Y. Ma, "Pruning and summarizing the discovered associations," in *Proc. 5th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 1999, pp. 125–134.

[12] H. Toivonen, M. Klemettinen, P. Ronkainen, K. Hätönen, and H. Mannila, "Pruning and grouping discovered association rules," in *Proc. MLnet Workshop Statist. Mach. Learn. Discovery Databases*, 1995, pp. 47–52.

[13] C. L. Li, "Association pattern analysis for pattern pruning, clustering and summarization," M.S. thesis, Elect. Comput. Eng., Univ. Waterloo, Waterloo, ON, Canada, 2008.

[14] R. J. Bayardo, R. Agrawal, and D. Gunopulos, "Constraint-based rule mining in large dense databases," in *Proc. 15th IEEE Int. Conf. Data Eng.*, Mar. 1999, pp. 188–197.

[15] S. Haberman, "The analysis of residuals in cross-classified tables," *Biometrics*, vol. 29, no. 1, pp. 205–220. 1973.

[16] A. K. C. Wong, T. S. Liu, and C. C. Wang, "Statistical analysis of residue variability in cytochrome *C*," *J. Molecular Biol.*, vol. 102, no. 2, pp. 287–295, Apr. 1976.

[17] A. K. C. Wong and T. S. Liu, "Typicality, diversity and feature patterns of an ensemble," *IEEE Trans. Comput.*, vol. C-24, no. 2, pp. 158–181, Feb. 1975.

[18] A. K. C. Wong, D. Zhuang, G. C. L. Li, and E.-S. A. Lee, "Discovery of non-induced patterns from sequences," in *Proc. 5th IAPR Int. Conf. Pattern Recognit. Bioinformat.*, 2010, pp. 149–160.

[19] J. Li, G. Liu, and L. Wong, "Mining statistically important equivalence classes and delta-discriminative emerging patterns," in *Proc. 13th ACM SIGKDD Intl. Conf. Knowl. Discovery Data Mining*, 2007, pp. 430–439.

[20] A. K. C. Wong and Y. Wang, "High order pattern discovery from discrete-valued data," *IEEE Trans. Knowl. Data Eng.*, vol. 9, no. 6, pp. 877–893, Nov/Dec. 1997.

[21] F. Camastra, M. Spinetti, and A. Vinciarelli, "Offline cursive character challenge: A new benchmark for machine learning and pattern recognition algorithms," in *Proc. 18th Int. Conf. Pattern Recognit.*, Aug. 2006, pp. 913–916.

[22] W. Wang and Y. Jiong, "Statistically significant patterns," in *Proc. Mining Sequential Patterns Large Data Sets*, 2005, pp. 63–112.

[23] G. Cumming, *Understanding The New Statistics: Effect Sizes Confidence Intervals, and Meta-Analysis*. Evanston, IL, USA: Routledge, 2012, pp. 27–28.

[24] J. Cheng, Y. Ke, and W. Ng, "A survey on algorithms for mining frequent itemsets over data streams," *Knowl. Inf. Syst.*, vol. 16, no. 1, pp. 1–27, 2008.

[25] Y.-C. Chen, J. T.-Y. Weng, and L. Hui, "A novel algorithm for mining closed temporal patterns from interval-based data," *Knowl. Inf. Syst.*, vol. 46, no. 1, pp. 151–183, 2015,

[26] P. Papapetrou, G. Kollios, S. Sclaroff, and D. Gunopulos, "Mining frequent arrangements of temporal intervals," *Knowl. Inf. Syst.*, 2009, pp. 133–171.

[27] W. H. Au, K. C. C. Chan, and A. K. C. Wong, "A fuzzy approach to partitioning continuous attributes for classification," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 5, pp. 715–719, May 2006.

[28] G. P. K. Wu, K. C. C. Chan, and A. K. C. Wong, "Unsupervised fuzzy pattern discovery in gene expression data," *BMC Bioinformat.*, vol. 12, no. 5, p. 1, Jul. 2011.

[29] D.-I. Lin and Z. Kedem, "Pincer-search: A new algorithm for discovering the maximum frequent set," in *Proc. EDBT Conf.*, 1998, pp. 103–119.

[30] C. C. Aggarwal, M. A. Bhuiyan, and M. Al Hasan, *Frequent Pattern Mining Algorithms: A Survey, Frequent Pattern Mining*. Springer International Publishing, 2014, pp. 19–64.

[31] R. J. Bayardo,Jr., "Efficiently mining long patterns from databases," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 1998, pp. 85–93.

[32] K. Gouda and M. Zaki, "Genmax: An efficient algorithm for mining maximal frequent itemsets," *Data Mining Knowl. Discovery*, vol. 11, no. 3, pp. 223–242, 2005.

[33] M. J. Zaki and C. J. Hsiao, "CHARM: An efficient algorithm for closed association rule mining," Dept. Comput. Sci., Rensselaer Polytechnic Inst., vol. 10. Tech. Rep. 99, 1999.

[34] J. Wang, J. Han, and J. Pei, "CLOSET+: Searching for the best strategies for mining frequent closed itemsets," in *Proc. ACM KDD Conf.*, 2003, pp. 236–245.

[35] C. Lucchese, S. Orlando, and R. Perego, "Fast and memory efficient mining of frequent closed itemsets," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 1, pp. 21–36, Jan. 2006.

[36] A. K. C. Wong and E.-S. A. Lee, "Aligning and clustering patterns to reveal the protein functionality of sequences," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 1, no. 3, pp. 548–560, May/Jun. 2014.

[37] E. A. Lee, F. J. Whelan, D. M. E. Bowdish, and A. K. C. Wong, "Partitioning and correlating subgroup characteristics from aligned pattern clusters," *Bioinformatics*, to be published, doi: 10.1093/bioinformatics/btw2193.

[38] K. C. C. Chan and A. K. C. Wong, "APACS: A system for the automatic analysis and classification of conceptual patterns," *Comput. Intell.*, vol. 6, no. 3, pp. 119–131, Aug. 1990.

[39] X. Yan, H. Cheng, J. W. Han, and D. Xin, "Summarizing itemset patterns: A profile-based approach," in *Proc. 11th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2005, pp. 314–323.

[40] J. Ruoming, M. Abu-Ata, Y. Xiang, and N. Ruan, "Effective and efficient itemset pattern summarization: Regression-based approaches," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008. pp. 399–407.

[41] F. Afrati, A. Gionis, and H. Mannila, "Approximating a collection of frequent sets," in *Proc. KDD*, 2004, pp. 12–19.

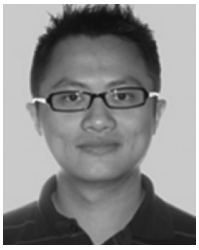
[42] X. Yan, H. Cheng, J. Han, and D. Xin, "Summarizing itemset patterns: A profile-based approach," in *Proc. KDD*, 2005, pp. 314–323.

[43] C. Wang and S. Parthasarathy, "Summarizing itemset patterns using probabilistic models," in *Proc. KDD*, 2006, pp. 730–735.

[44] L. Chunyang, L. Chen, and C. Zhang, "Summarizing probabilistic frequent patterns: A fast approach," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 527–535.

[45] A. K. C. Wong, B. Wu, G. P. K. Wu, and K. C. C. Chan, "Pattern discovery for large mixed-mode database," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2010, pp. 859–868.

PEI-YUAN ZHOU received the B.Sc. degree from the faculty of Information Technology, Macau University of Science and Technology, Macau, in 2010, and the M.Sc. degree from the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, in 2011. She is currently pursuing the Ph.D. degree with the Department of Computing, The Hong Kong Polytechnic University. Her research interests include multivariate time series mining and pattern analysis.



GARY C. L. LI (M'04) received the B.A. degree (Hons.) and the M.Phil. degree in computing from The Hong Kong Polytechnic University, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo. In 2006, he invented a patented technology Pattern Clustering and Data Grouping. He co-founded Pattern Intelligence Inc., a company that developed Production Intelligence in Oil Industry. Pattern Intelligence was sold to Pattern Discovery Technologies Inc., in 2009. He served as a Consultant for numerous high-tech commercialization projects and Research and Development projects in production intelligences (oil sands and bitumen extraction), search industry (text, images and videos), finance, education, and governments. He is currently a Senior Data Scientist with Amazon. He has authored articles in top journals and international conferences in the area of data mining and knowledge discovery. His research interests include knowledge discovery, data mining, and pattern recognition.

ANDREW K. C. WONG (F'04) received the B.Sc. degree (Hons.) and the M.Sc. degree from The Hong Kong University, and the Ph.D. degree from Carnegie Mellon University. He was the Founding Director of the Pattern Analysis and Machine Intelligence Laboratory with UW, and the Distinguished Chair Professor with The Hong Kong Polytechnic University from 2000 to 2003. He is currently a Distinguished Professor Emeritus (Systems Design Engineering at the University of Waterloo). He has authored extensively over 300 papers/chapters. He holds five patents. His research areas cover machine intelligence, computer vision, intelligence robotics, pattern recognition, data mining, and bioinformatics. He has been invited as a keynote/plenary speaker for IEEE international conferences and the IEEE Distinguished Speaker Program. He served as the General Chair of the ISTED Conference of Robotics and Control, Hawaii, USA, in 1996, and the General Chair of the IEEE/ RSJ International Intelligence Robotic Systems Conference, Victoria, BC, Canada, in 1998. He is a Co-Founder of Virtek Vision International Corporation, a publicly traded company in TSX in laser vision technology. He was the President of the organization from 1986 to 1993 and the Chairman from 1993 to 1997. In 1997, he co-founded Pattern Discovery Software Systems Ltd., and has served as the Chairman till 2013.