Inferring Implicit Rules by Learning Explicit and Hidden Item Dependency

Shoujin Wang, *Student Member, IEEE*, and Longbing Cao, *Senior Member, IEEE*

***Abstract*—Revealing complex relations between entities (e.g., items within or between transactions) is of great significance for business optimization, prediction, and decision making. Such relations include not only co-occurrence-based explicit relations but also nonco-occurrence-based implicit ones. Explicit relations have been substantially studied by rule mining-based approaches, including association rule mining and causal rule discovery. In contrast, implicit relations have received much less attention but could be more actionable. In this paper, we focus on the implicit relations between items which rarely or never co-occur while each of them co-occurs with other identical items (link items) with a high probability. A framework integrates both explicit and hidden item dependencies and a corresponding efficient algorithm IRRMiner captures such implicit relations with implicit rule inference. Experimental results show that IRRMiner not only infers implicit rules of various sizes consisting of both frequent and infrequent items effectively, it also runs at least four times faster than IARMiner, a typical indirect association rule mining algorithm which can only mine size-2 indirect association rules between frequent items. IRRMiner is applied to make recommendations and shows that the identified implicit rules can increase recommendation reliability.**

***Index Terms*—Hidden dependency, implicit rules, pattern relation analysis, rule inference.**

I. Introduction

A. Target Problem and Motivation

RULE mining is an important task and a key issue in knowledge discovery and data mining [1], [2]. The resultant rules are sometimes quite useful for business optimization, prediction and decision making [3]. Classic rule mining methods, e.g., association rule mining [4] and causal rule mining [5], are essentially based on explicit co-occurrences only, and focus on explicit and dependent relations (e.g., associations and causal relationships) while ignoring more implicit relations [6]–[8]. For example, association rule mining relies on the *support-confidence* framework to select those items which co-occur frequently to generate rules to infer the explicit and straightforward associations between them, while causal discovery discovers the direct cause-effect relations between two items (variables) by showing how the variations of one item cause changes to the other [9].

TABLE I
Instance of the ERD Dataset

Items / Transactions	*pizza*	*napkins*	*coke*	*sprite*
t_1	0	1	0	1
t_2	1	1	1	0
t_3	0	1	0	0
t_4	0	1	1	0
t_5	1	1	0	1
t_6	1	1	1	1

However, it is not trivial to capture *implicit relations* [10], [11] by analyzing rule relations (also called pattern relation analysis) [8] to make the identified rules actionable [12]. In this paper, *implicit relations* [8], [13] refer to the connections between several items which do not co-occur frequently but have a high probability of co-occurring with the third-party identical items. Here, the third-party items are called *link itemset* as they serve as bridges to connect those rarely or never co-occurring items. Such implicit relations cannot be identified by association rule mining or causal rule discovery without pattern relation analysis. In some cases, implicit relations are even more valuable for discovering novel and unexpected rules to support business events, compared to straightforward associations or causal relations. By taking the third-party items (i.e., link items) into account, implicit rules are also more informative than explicit ones, which only focus on their main aspects (e.g., antecedent and consequent).

Researchers have realized the significance of implicit relations between items and have proposed indirect association mining [14], [15]. However, it is built on association rule mining [16], which only makes it applicable for frequent items while ignoring infrequent ones. Furthermore, existing indirect association mining only focuses on pairwise relations (e.g., the relation between sprite and coke) while ignoring the complex relations among multiple items (e.g., the relation among sprite, coke, and pepsi).

Taking the ERD data[1] as an example (Table I), the values 0 and 1 in the first and second rows of column 1 indicate that *pizza* is not bought in transaction t_1 but in t_2. It is easy to infer the implicit rule $coke \oplus sprite | pizza$, which indicates that either *coke* or *sprite*, but not both, is quite likely to be bought

Manuscript received May 15, 2017; accepted September 17, 2017. This paper was recommended by Associate Editor E. Chen *(Corresponding author: Longbing Cao.)*

The authors are with the Advanced Analytics Institute, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: shoujin.wang@student.uts.edu.au; longbing.cao@uts.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2017.2768547

[1] An electronic retail transaction dataset from a Chinese E-commerce platform.

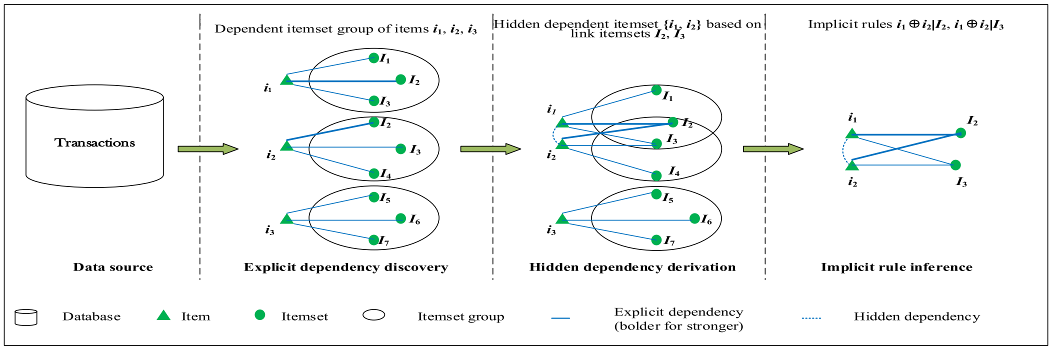

Dependent itemset group of items i_1, i_2, i_3
Hidden dependent itemset $\{i_1, i_2\}$ based on link itemsets I_2, I_3
Implicit rules $i_1 \oplus i_2|I_2$, $i_1 \oplus i_2|I_3$
Transactions
i_1
i_2
i_3
I_1
I_2
I_3
I_4
I_5
I_6
I_7
Data source
Explicit dependency discovery
Hidden dependency derivation
Implicit rule inference
Database
Item
Itemset
Itemset group
Explicit dependency (bolder for stronger)
Hidden dependency

Fig. 1. Implicit rule inference framework which combines both explicit and hidden dependency.

with *pizza*. This rule reveals the shopping preferences that *coke* and *sprite* may not be usually bought together since they share the same function, whereas they are quite likely to be bought together with the same third-party goods like *pizza*. This kind of implicit connection between *coke* and *sprite* is conditional on the link itemset *pizza*. It can not only help with increasing profit through competitive product analysis [17] (*coke* and *sprite* are competitive products) but can also contribute to precise recommendations by reducing redundant items (*coke* and *sprite* are likely to be redundant if recommended to one consumer at the same time). Such implicit relations cannot be identified by traditional explicit co-occurrence-based rule mining approaches like association rule mining [18] or causal rule discovery [19] because of their extremely low co-occurrences and hidden relations between the items involved.

In fact, although some items are implicitly related, it is possible to identify such relationships. For example, a person may buy *pizza* and *coke* for a lunch, but try *pizza* and *sprite* next time. In reality, such partial replacement in product combinations is quite popular in areas, such as commerce and medical services. Capturing such implicit and complex relations and then inferring implicit rules helps businesses to deeply understand customer consuming behaviors, which provides more solid support for business optimization, prediction, and decision making [20], [21].

B. Our Design and Main Contributions

The above observation shows the importance of counting implicit relations between items and the feasibility of identifying implicit rules composed of infrequently or even never co-occurring items. Here, we propose a novel framework to mine implicit rules.

A "three-step" framework is proposed to mine implicit rule $x \oplus y|Z$ (meaning items x and y are implicitly related with Z as the link itemset), which is illustrated in Fig. 1 and explained as follows.

1) Identify all dependent itemsets of each item in the transactional dataset.
2) For given items x and y, if they share at least one identical dependent itemset Z, itemset $\{x, y\}$ is chosen as a hidden dependent itemset.
3) Compute the implicit relation strength (IRS) between x and y; if it is larger than a predefined threshold, itemset $\{x, y\}$ is selected as an implicitly related itemset. Based on this, an implicit rule $x \oplus y|Z$ is inferred.

The main contributions of this paper are as follows.

1) A novel *implicit rule inference* framework is proposed to infer implicit relation-based rules (IRRs), which follows a three-step strategy. We call the implicit IRR *implicit rules* for simplification in this paper.
2) An *implicit rule inference* algorithm, *IRRMiner*, is proposed, by which those items which rarely or never co-occur but are implicitly closely related to each other are detected.
3) IRRMiner is used in recommendation as a case study, which enables the recommendations to be more reliable and precise in real-world business.

II. Related Work

Rule mining is an important research issue and has been widely explored. Much progress has been achieved in various subfields in this field. Here, we briefly review some of the work which is closely related to our proposed implicit rule inference: association rule mining, correlation rule mining, causal discovery, and indirect association mining.

A. Association Rule Mining

Well-known algorithms focusing on association rule mining include AIS [22], Apriori [18], FP-Tree [23] and the linear prefix tree-based algorithm [24]. All these methods focus on the improvement of algorithm efficiency and much progress has been achieved by utilizing more effective candidate generation methods and pruning strategies. However, they are all based on the *support-confidence* framework and target explicit "co-occurrence" based associations. As a result, they only capture the explicit and straightforward relations while ignoring implicit relations. They filter out infrequent items which may be of significance and simply focus on the main aspects (antecedent and consequent) while ignoring the influence of other related aspects (e.g., link items).

B. Correlation Rule Mining

Correlation rule mining is another important branch in the rule mining area. It tries to mine those statistically correlated items driven by frameworks different from the "support-confidence" one. Specifically, some measures to describe the correlations between distinct items are used as the selection criteria. In particular, *lift* and χ^2 *correlation* were introduced in 1997 to mine correlation rules [25]. Approximate similarity measures have to be proposed to explore efficient computation due to the lack of the downward closure property for lift and χ^2. Another two interesting measures for correlation rule mining, *all confidence* and *bond*, which were introduced in 2003 [26], follow the downward closure properties and can be used to mine correlation rules efficiently. Some other related work includes mining both associated and correlated rules [27], the efficient mining of correlation rules with multiple all-confidence thresholds and so on [28]. The progress achieved in correlation rule mining can leverage some of the drawbacks of association rule mining illustrated above by including infrequent items. However, correlation rules still only focus on explicit relations while ignoring implicit ones. In addition, similar to association rules, correlation rules do not take the influence of related aspects (e.g., link items) into account when capturing the relations between the main aspects.

C. Causal Discovery

The three main approaches for causal discovery are: 1) graphical causal modeling; 2) constraint-based approaches; and 3) association rule-based approaches. Causal relationships are represented by Bayesian networks or similar probabilistic graphical models in graphical causal modeling [29]. These methods cannot handle high dimensional data and often incur high computational cost. To be more efficient, constraint-based approaches learn several fixed local causal structures, e.g., CCC [30] and CCU [31] rather than the complete Bayesian network [32]. Such methods cannot identify causal relations, which cannot be represented by these fixed structures and cannot identify combined causal factors. Association rule mining has recently been combined with a cohort study to discover causal association rules and has proven to be quite effective [33]. However, similar to association rule mining, these approaches tend to target direct and explicit causal relations, while more implicit and indirect relations are ignored.

D. Indirect Association Mining

Identifying implicit rules has rarely been explored to the best of our knowledge. The most related work is indirect association rule mining. The concept of indirect association rules infers the relation between two items which are not associated with each other directly but both of them are associated with identical third-party itemsets, called mediators. Hamano and Sato [14] proposed a framework to mine indirect association rules to analyze targeting consumers and competitors. Specifically, given an item pair which co-occurs infrequently and a mediator itemset, a dependency constraint μ is used to ensure the strong direct associations between each item from the item pair and the mediator itemset, and then the indirect association between the pair of items is derived based on the strong direct associations. Another algorithm called IPMA [15] uses a new dependency measure, called critical relative support, to replace μ while retaining the same other procedures as the Hamano and Sato framework. Other variants of indirect association mining include mining temporal indirect associations [34], mining direct and indirect fuzzy sequential patterns [35], and mining indirect associations in Web data for Web recommendation [36]. Although different algorithms and techniques as well as their variants are developed to capture indirect associations, essentially, their frameworks are the same as the typical one proposed in [14]. All of these algorithms take a step forward in capturing implicit relations; however, they basically extend the association rule mining framework by simply putting two association rules together to derive an indirect one [16], which limits them to frequent items while filtering out infrequent ones. As a result, they can only discover the indirect associations between two frequent items. However, implicit relations may also exist between several infrequent items and between some infrequent items and frequent ones. Such two kinds of implicit relations involving infrequent items are ignored by all the existing methods. This is caused by frequency-based association rule mining, which is the foundation of current indirect association mining frameworks. Another drawback of existing broadly used frameworks is that they only focus on the relations between *two* items (e.g., i_1, i_2), which makes them inapplicable for itemsets with more than two items (e.g., $i_1, i_2, i_3 \ldots$). In addition, most of the existing work in this area only focus on rule mining algorithm development while ignore the application mechanisms [37] of the resultant rules.

In summary, existing work cannot effectively derive implicit rules from analyzing the explicit and implicit combinations of items and consider nonoccurring and novel items. In addition, more recent work highlights the need of discovering actionable knowledge from data [12], [38] that can support decision-making. However, quite limited empirical studies have been conducted to show why the above resultant rules are useful and how they benefit real-world businesses. To address this issue, pattern relation analysis [8] is undertaken in this paper to learn the explicit and implicit item dependency and infer implicit rules.

III. Framework for Implicit Rule Inference

We first give the precondition and then illustrate the three-step framework discussed in the introduction. The three steps refer to explicit dependency discovery, hidden dependency derivation, and implicit rule inference by integrating both explicit and hidden item dependency.

A. Precondition

To identify the implicit rules in which items infrequently or never co-occur, the first step is to remove those item combinations of frequently co-occurring items like {*pizza*, *coke*}, i.e., itemsets with high frequency (e.g., {*pizza*, *coke*}). This is because these items are explicitly associated and can be easily and efficiently mined using frequent pattern mining techniques like Apriori, which is out of the scope of this paper. Note that frequent items are still kept to form implicit rules. For example, both *coke* and *sprite* are frequent, but they rarely co-occur within one transaction, i.e., Sup(*coke*, *sprite*) is low, and they constitute a typical implicit rule. The precondition for a given itemset to be implicitly related is that its support is not larger than a minimum threshold. This precondition greatly benefits our proposed algorithm by pruning those frequently co-occurring items thus reducing the search space.

Precondition 1: It is possible for a given itemset $I = \{i_1 \cdots i_j\}$ to be an implicitly related itemset only if it meets the following precondition:

$$\mathrm{Sup}(I) \leq \mathrm{minsup} \tag{1}$$

where $\mathrm{Sup}(I)$ is the support of itemset I and minsup is the predefined minimal support.

B. Explicit Item Dependency Discovery

Given a transactional dataset as shown in Table I, each row indicates a transaction, such as t_1, t_2, and all the transactions constitute the transactional set T, $T = \{t_1, t_2 \ldots t_{|T|}\}$. Each column indicates an item like pizza and coke while the value 1 means that an item occurs in the corresponding transaction, otherwise the value is 0. All the items in the transactional table constitute the full itemset U, $U = \{i_1, i_2 \ldots i_{|U|}\}$. Each transaction t is a subset of U, $t \subseteq U$. For example, the first transaction t_1 includes two distinct items: 1) napkins and 2) sprite.

With the transactional information, the explicit dependency between an item and itemsets is identified using point-wise mutual information (PMI) [39] for its strong ability to capture both nonlinear and linear dependencies [40], [41]. The PMI between item i and itemset I is calculated as

$$\mathrm{PMI}(i, I) = \log \frac{p(i, I)}{p(i)p(I)} \tag{2}$$

where $p(i)$ and $p(I)$ are the marginal probabilities of i and I, respectively, while $p(i, I)$ is their joint probability.

Definition 1 (Dependent Itemset): An itemset I is defined as a *dependent itemset* of a given item $i (i \notin I)$ (denoted as $S_i = I$) if the PMI between them is positive, that is, $\mathrm{PMI}(i, I) > 0$. It is denoted as $S_{i1}, S_{i2}, \ldots$ if item i has more than one dependent itemset.

Note that generally $\mathrm{PMI}(i, I) \in (-\infty, \min[-\log p(i), -\log p(I)]]$, however, $\mathrm{PMI}(i, I) \in (0, \min[-\log p(i), -\log p(I)]]$ in our algorithm to ensure the positive dependency between i and I.

Based on the dependent itemset concept, the dependent itemset group is defined as follows.

Definition 2 (Dependent Itemset Group): For a given item i, all its dependent itemsets $(S_{i1}, S_{i2} \ldots)$ constitute its dependent itemset group, denoted as $\boldsymbol{A}_i$

$$\boldsymbol{A}_i = \{S_{i1}, S_{i2} \cdots\}. \tag{3}$$

Example 1: Taking item coke in Table I as an example, its dependent itemset group $\boldsymbol{A}_{coke} = \{S_{coke1}, S_{coke2}\} = \{\{pizza\}, \{pizza, napkins\}\}$.

C. Hidden Item Dependency Derivation

Given an itemset $I = \{i_1, i_2 \ldots i_j\}$, the dependent itemset group $\boldsymbol{A}_{i_1}, \boldsymbol{A}_{i_2} \ldots \boldsymbol{A}_{i_j}$ of each item from I is, respectively, identified. $\boldsymbol{A}_{i_j} = \{S_{i_j 1}, S_{i_j 2} \ldots S_{i_j k}\}$, where $S_{i_j k}$ is the kth dependent itemset of i_j.

Definition 3 (Link Itemset and Link Itemset Group): Given an itemset $I = \{i_1, i_2 \ldots i_j\}$ and the dependent itemset group $\boldsymbol{A}_{i_j}$ of each item i_j from I, the link itemset group of I is defined as the intersection set of all dependent itemset groups of items within I, denoted as $\boldsymbol{G}_I$. Each element of $\boldsymbol{G}_I$ is defined as a link itemset of I, denoted as H_I. It is denoted as $H_{I1}, H_{I2} \ldots$ when itemset I has more than one link itemset

$$\boldsymbol{G}_I = \boldsymbol{A}_{i_1} \cap \boldsymbol{A}_{i_2} \ldots \boldsymbol{A}_{i_j} = \{H_{I1}, H_{I2} \ldots\}. \tag{4}$$

Definition 4 (Hidden Dependent Itemset): Given an itemset I, it is defined as hidden dependent if its link itemset group is not empty. Formally

$$\boldsymbol{G}_I \neq \emptyset. \tag{5}$$

Example 2: Let us take itemset $\{coke, sprite\}$ from Table I as an example. The dependent itemset groups of items *coke* and *sprite* are $\boldsymbol{A}_{coke} = \{\{pizza\}, \{pizza, napkins\}\}$ and $\boldsymbol{A}_{sprite} = \{\{pizza\}, \{pizza, napkins\}\}$, respectively. Hence, the link itemset group of $\{coke, sprite\}$ is $\boldsymbol{G}_{\{coke, sprite\}} = \boldsymbol{A}_{coke} \bigcap \boldsymbol{A}_{sprite} = \{\{pizza\}, \{pizza, napkins\}\} \neq \emptyset$. Accordingly, $\{coke, sprite\}$ is a hidden dependent itemset with two link itemsets $\{pizza\}$ and $\{pizza, napkins\}$.

D. Implicit Rule Inference

Given a hidden dependent itemset, we first compute its IRS and then select those itemsets whose IRS is larger than a minimum threshold as implicitly related itemsets. Lastly, we infer implicit rules based on these itemsets.

Given a hidden dependent itemset $I = \{i_1, i_2 \ldots i_j\}$ together with its link itemset group $\boldsymbol{G}_I (\boldsymbol{G}_I \neq \emptyset)$, its IRS is calculated under the intuition that if I has more link itemsets and the items $(i_1, i_2 \ldots i_j)$ within I have stronger dependencies on them, these items are more strongly implicitly connected. As a result, the IRS of I is larger.

Definition 5 [Conditional IRS (CIRS)]: Given a hidden dependent itemset $I = \{i_1, i_2 \ldots i_j\}$ and a link itemset H_I, its IRS conditional on H_I is computed as

$$\mathrm{CIRS}(I|H_I) = \min\bigl(\mathrm{PMI}(i_1, H_I) \cdots \mathrm{PMI}\bigl(i_j, H_I\bigr)\bigr) \tag{6}$$

where $\mathrm{CIRS}(I|H_I) \in (0, \min(-\log p(i_1), \ldots - \log p(i_j), -\log(H_I))$.

Definition 6 (IRS): Given an hidden dependent itemset $I = \{i_1, i_2 \ldots i_j\}$ and its link itemset group $\boldsymbol{G}_I$, its IRS is computed by summing its CIRS on all link itemsets. Formally

$$\mathrm{IRS}(I) = \sum_{H_I \in \boldsymbol{G}_I} \mathrm{CIRS}(I|H_I) \tag{7}$$

where $\mathrm{IRS}(I) \in (0, \sum_{H_I \in \boldsymbol{G}_I} \min(-\log p(i_1), \ldots - \log p(i_j), -\log(H_I))$. The larger $\mathrm{IRS}(I)$ is, the stronger the implicit relation that exists between the items within I.

Definition 7 (Implicitly Related Itemset): An implicitly related itemset candidate is implicitly related if its IRS is larger than the minimum threshold. Formally

$$\mathrm{IRS}(I) \geq \mathrm{minIRS} \tag{8}$$

where minIRS is a predefined threshold to ensure that strong enough implicit relations exist between the items within I.

Definition 8 (Implicit Rules): Given an implicitly related itemset $I = \{i_1, i_2 \cdots i_j\}$ and its link itemset group $\boldsymbol{G_I} = \{H_{I1} \ldots H_{In}\} (n = |\boldsymbol{G_I}|)$, n implicit rules are inferred, which constitute an implicit rule cluster $\boldsymbol{R}_0$

$$\mathbf{R}_0 \begin{cases} r_{01}: i_1 \oplus i_2 \oplus \cdots i_j | H_{I1} \\ r_{02}: i_1 \oplus i_2 \oplus \cdots i_j | H_{I2} \\ \cdots \\ r_{0n}: i_1 \oplus i_2 \oplus \cdots i_j | H_{In}. \end{cases} \tag{9}$$

All the rules from $\boldsymbol{R}_0$ share the same implicitly related itemset $(\{i_1, i_2, \ldots, i_j\})$ but take different link itemsets (e.g., H_{I1} or H_{I2}) as their conditions. The first rule $i_1 \oplus i_2 \oplus \cdots \oplus i_j | H_{I1}$ implies that once H_{I1} has been bought, there is great probability that one out of $i_1, i_2 \ldots i_j$ will be bought.

Example 3: Following the implicitly related itemset candidate $\{coke, sprite\}$ in Example 2, its conditional IRS (CIRS) conditional on its link itemset $\{pizza\}$ is $\mathrm{CIRS}(\{coke, sprite\}|\{pizza\}) = \min(\mathrm{PMI}(coke, \{\mathrm{pizza}\}), \mathrm{PMI}(sprite, \{pizza\})) = 0.12$. Similarly, $\mathrm{CIRS}(\{coke, sprite\}|\{pizza, napkins\}) = 0.12$. Accordingly, $\mathrm{IRS}(\{coke, sprite\}) = 0.24$. If we set $\mathrm{minIRS} = 0.1$, $\{coke, sprite\}$ is an implicitly related itemset. Based on this, two implicit rules $coke \oplus sprite|pizza$ and $coke \oplus sprite|\{pizza, napkins\}$ are derived. In reality, such rules indicate that *coke* and *sprite* are rarely bought together, whereas they are much more likely to be bought together with other itemsets $\{pizza\}$ or $\{pizza, napkins\}$. This can be seen from the transactions in Table I. These observations are consistent with customer shopping behaviors, whereby one may prefer to buy a basket of products with different functions rather than the same function.

Algorithm 1 Implicit Rule Inference

```
Require: T: binary transaction matrix; minsup: support
    threshold; minIRS: IRS threshold; MaxSize: the maximum
    size of implicitly related itemsets;
Ensure: P: implicitly related itemsets; H: link itemsets;
 1: Mine dependent itemset group $A_i$ of each item $i$ according
    to Definition 2 and store items with nonempty $A_i$ in D;
 2: while L=2 do
 3:   Generate all possible size-L itemsets on D and store
      them in $Q\{L\}$;
 4:   for each itemset $I \in Q\{L\}$ do
 5:     if $\boldsymbol{G_I} \neq \emptyset$ then
 6:       Select $I$ as size-L hidden dependent itemset and
          store it in $C\{L\}$;
 7:     end if
 8:   end for
 9:   for each itemset $I_c$ in $C\{L\}$ do
10:     if $IRS(I_c) \geq minIRS$ then
11:       Store $I_c$ in $CC\{L\}$;
12:       if $Sup(I_c) \leq minsup$ then
13:         Select $I_c$ as implicitly related itemset and store
            it in $P\{L\}$, and store all its link itemsets in $H\{L\}$;
14:       end if
15:     end if
16:   end for
17: end while
18: while L=3 :MaxSize do
19:   for $m = 1 : Size(CC\{L-1\})$ do
20:     for $n = m + 1 : Size(CC\{L-1\})$ do
21:       if $CC\{L-1\}(m)([1:L-2]) == CC\{L-1\}(n)([1:L-2])$ and $CC\{L-1\}(m)(L-1) \neq CCL-1(n)(L-1)$ then
22:         Store $[CC\{L-1\}(m), CC\{L-1\}(n)(L-1)]$ into
            $Q\{L\}$ as size-L itemset ;
23:       end if
24:     end for
25:   end for
26:   for each size-L itemset $I$ in $Q\{L\}$ do
27:     if all size-(L-1) sub-itemsets of $I$ are in $CC\{L-1\}$
        then
28:       Execute the same operations from Lines 4 to 16;
29:     end if
30:   end for
31: end while
```

IV. IRRMiner Algorithm

Following the framework illustrated in the previous section, the *IRRMiner* algorithm is developed to mine implicit rules. The following anti-monotonous Property 1 is used to generate size-L implicit itemset candidates from size-(L-1) ones directly to reduce the search space. The size of an implicit rule is defined in (10) in Section V. Applying this property from lines 16 to 22 in Algorithm 1 guarantees to find all rules satisfying the given constraints efficiently. Next, we first give and prove such a property theoretically and then describe the implicit rule inference algorithm below.

Property 1: Given a candidate itemset $I = \{i_1 \cdots i_j\}$ and its IRS ($\mathrm{IRS}(I)$), any subset $I'(I' \subseteq I, |I'| \geq 2)$ must not have a lower IRS, namely $\mathrm{IRS}(I') \geq \mathrm{IRS}(I)$.

Proof: Given a candidate itemset $I = \{i_1, i_2 \ldots i_l \ldots i_j\}(|I| \geq 3)$ and one of its subset $I' = \{i_1, i_2 \ldots i_l\}(l < j, |I'| \geq 2)$, according to Definition 3, their link itemset groups are $\boldsymbol{G_I} = \boldsymbol{A}_{i_1} \cap \boldsymbol{A}_{i_2} \ldots \boldsymbol{A}_{i_l} \ldots \boldsymbol{A}_{i_j}$ and $\boldsymbol{G'_I} = \boldsymbol{A}_{i_1} \cap \boldsymbol{A}_{i_2} \ldots \boldsymbol{A}_{i_l}$, respectively. So $\boldsymbol{G_I} = \boldsymbol{G'_I} \cap \boldsymbol{A}_{i_{l+1}} \ldots \boldsymbol{A}_{i_j} \subseteq \boldsymbol{G'_I}$. Assume $\boldsymbol{G_I} = \{H_1, H_2 \cdots H_k\}$ while $\boldsymbol{G'_I} = \{H_1, H_2 \cdots H_k \cdots H_m\}(m > k)$, based on Definitions 5 and 6, $\mathrm{IRS}(I) = \mathrm{CIRS}(I|H_1) + \mathrm{CIRS}(I|H_2) + \cdots \mathrm{CIRS}(I|H_k)$ while $\mathrm{IRS}(I') = \mathrm{CIRS}(I'|H_1) + \mathrm{CIRS}(I'|H_2) + \cdots \mathrm{CIRS}(I'|H_k) + \cdots + \mathrm{CIRS}(I'|H_m)$. Now we compare $\mathrm{CIRS}(I|H_1)$ and $\mathrm{CIRS}(I'|H_1)$, $\mathrm{CIRS}(I|H_1) = \min\{\mathrm{PMI}(i_1, H_1)$, $\mathrm{PMI}(i_2, H_1) \cdots \mathrm{PMI}(i_l, H_1)$, $\mathrm{PMI}(i_{l+1}, H_1) \cdots \mathrm{PMI}(i_j, H_1)\}$ and $\mathrm{CIRS}(I'|H_1) = \min\{\mathrm{PMI}(i_1, H_1)$, $\mathrm{PMI}(i_2, H_1) \cdots \mathrm{PMI}(i_l, H_1)\}$, it is clear that $\mathrm{CIRS}(I|H_1) = \min\{\min\{\mathrm{PMI}(i_1, H_1)$, $\mathrm{PMI}(i_2, H_1) \cdots \mathrm{PMI}(i_l, H_1)\}$, min

$\{\mathrm{PMI}\ (i_{1+1}, H_1) \cdots \mathrm{PMI}(i_j, H_1)\}\} = \min\{\mathrm{CIRS}(I'|H_1), \min\{\mathrm{PMI}(i_{1+1}, H_1) \cdots \mathrm{PMI}(i_j, H_1)\}\} \leq \mathrm{CIRS}(I'|H_1)$. Similarly, $\mathrm{CIRS}(I|H_2) \leq \mathrm{CIRS}(I'|H_2), \ldots, \mathrm{CIRS}(I|H_k) \leq \mathrm{CIRS}(I'|H_k)$, hence $\mathrm{IRS}(I) \leq \mathrm{CIRS}(I'|H_1) + \mathrm{CIRS}(I'|H_2) + \cdots \mathrm{CIRS}(I'|H_k)$. Recall that $\mathrm{CIRS}(I|H_l) > 0$ as illustrated in Definition 5, it is easy to conclude that $\mathrm{IRS}(I) \leq \mathrm{CIRS}(I'|H_1) + \mathrm{CIRS}(I'|H_2) + \cdots \mathrm{CIRS}(I'|H_k) + \cdots + \mathrm{CIRS}(I'|H_m) = \mathrm{IRS}(I')$. Hence, Property 1 is proved. ■

Combining Property 1 and Definition 7, it is easy to conclude that any subset $I'(I' \subseteq I, |I'| \geq 2)$ of an implicitly related itemset I is also an implicitly related itemset if I' meets the precondition $\mathrm{Sup}(I') \leq \mathrm{minsup}$. Such conclusion not only reduces the search space in the candidate generation process but also helps to identify whether a size-L candidate itemset is implicitly related or not by checking all its size-(L-1) subsets. This contributes a lot to the space and time efficiency of the whole *IRRMiner* algorithm.

Algorithm 1 is divided into three parts: 1) discover dependent itemset groups (line 1); 2) mine size-2 implicit rules and prepare to mine implicit rules of larger sizes (lines 2–17); and 3) discover implicit rules with a size larger than 2 (lines 18–31). Specifically, all dependent itemsets are identified and possible size-2 itemsets are generated on those items whose dependent itemset group is not empty (lines 1–3). Then the link itemset group, IRS and support of these generated itemsets are checked one by one to filter out those nonpotential implicitly related itemsets step by step while keeping implicitly related itemsets together with their corresponding link itemsets as the output (lines 4–17). When the size of an itemset grows larger than 2, the anti-monotonous Property 1 is utilized to generate larger candidate itemsets more efficiently based on the pattern growth method (lines 19–25) and to conduct prefiltering on these itemsets (lines 26–27). Finally, the implicitly related itemsets and their link itemsets are achieved by undertaking the same filtering operations (line 28) as those used in the mining of size-2 implicit rules.

V. Experiments and Evaluation

A. Experiment Set Up

No existing work can exactly mine our proposed implicit rules, to the best of our knowledge, and only a typical indirect association rule mining algorithm (IARMiner) [14] can partially discover rules similar to ours. The rules mined by IARMiner are in the form of $(M; \{x, y\})$, where itemset M is the mediator itemset for connecting items x and y. These rules can be transferred to our implicit rules, such as $x \oplus y|M$. To evaluate our proposed algorithm, we compare our proposed IRRMiner with the representative indirect association rule mining algorithm IARMiner on four real-world transactional datasets: ERD, Bookcross,[2] MovieLens_1 and MovieLens_2. Such comparison has some limitations due to the nonexactly identical goals of the compared algorithms. Specifically, we can only make a comparison on the capability of mining size-2 rules instead of larger ones (rules of size-3 and size-4) between IRRMiner and IARMiner as IARMiner can only mine size-2 rules; the mined rules may not always be completely identical as the constraints used in IARMiner and IRRMiner are not completely the same. However, empirical results show that most of the resultant size-2 rules from both algorithms are the same. MovieLens_1 and MovieLens_2 are extracted from the MovieLens 10M[3] dataset by including different parts of transactions. A detailed description of these datasets is given in Table II. Items, books, and movies in the experimental datasets are called items uniformly in this paper to simplify the terms. Note that all the four transactional datasets are transferred into 0-1 encodings as Table I and the density shown in Table II is quantified by $Density = (\#entries\ valued\ 1/\#entries)$, for instance, the density of transaction Table I is $(15/24) = 62.5\%$.

Both IRRMiner and IARMiner have two key parameters: minsup (called t_r in [14]) is shared by the two algorithms while minIRS and t_μ are used in IRRMiner and IARMiner, respectively. To be specific, minsup is a frequency-constraint to ensure the implicitly related or indirect associated items do not co-occur frequently [e.g., $sup(x, y) \leq$ minsup]. minIRS is used to guarantee strong IRS between implicitly related items [e.g., $\mathrm{IRS}(x, y) \geq$ minIRS] in IRRMiner while t_μ is to ensure strong dependency between each of the indirectly associated items and the corresponding mediate itemset M [e.g., $\mu(x, M) \geq t\_\mu$]. In addition, two extra parameters t_f and t_m are also used in IARMiner, where t_f is to make sure each item in an indirectly associated item pair is frequent [e.g., $sup(x) \geq t\_f$] and t_m is to guarantee that it co-occurs with the mediate itemset frequently [e.g., $sup(x, M) \geq t\_m$]. In all the experiments, we keep the common parameter minsup(t_r) identical for both algorithms to ensure fair comparisons while empirically tuning other noncommon parameters.

In order to show the capacity of IRRMiner to cover infrequent items and to mine implicit rules larger than size-2, we conduct comparisons between IARMiner and IRRMiner in terms of rule coverage, rule size, and rule number in the following Parts B and C, respectively. To test the efficiency of our proposed IRRMiner, we compare the run time of IRRMiner and that of IARMiner in the following Part D. A data factor test is conducted in Part E to test the outcome difference of our proposed IRAMiner on datasets with different characteristics.

B. Rule Coverage Comparison

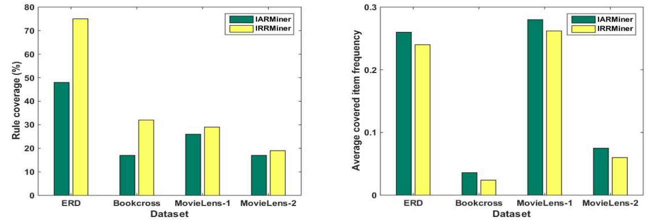
Nearly all the indirect rule mining approaches including IARMiner can only mine size-2 rules, to make a fair comparison, we also limit the size of rules from IRRMiner to 2 when comparing rule coverage. The coverage of the size-2 rules resulting from IARMiner and IRRMiner together with the average frequency of their covered items are given in Fig. 2(a) and (b), respectively. Here, *coverage* is defined as the ratio of items covered by all the mined rules with respect to the whole item population, while the frequency of a certain item is its occurrence times divided by the total number of transactions in a dataset. On one hand, Fig. 2(a) shows that the coverage of IRRMiner is obviously larger

[2] Available on http://grouplens.org/datasets/book-crossing/.

[3] Available on http://grouplens.org/datasets/movielens/.

TABLE II
STATISTICS OF EXPERIMENTAL DATASETS

Statistics	ERD	Bookcross	MovieLens_1	MovieLens_2
No. of Transactions	4626	10938	69878	69878
No. of Items	72	162	200	200
Avg. Items per Transaction	12.78	3.58	34.05	3.43
Avg. Frequency per Item	820.81	241.3	11895	1199
Density	20.56%	2.21%	17.02%	1.72%



(a) (b)

Fig. 2. Rule coverage and average covered item frequency on different datasets.

than IARMiner on all the four datasets, which means our proposed algorithm can discover implicit relations between more items than IARMiner. On the other hand, Fig. 2(b) illustrates that the average frequency of the covered items by IRRMiner is clearly lower than that by IARMiner, which means our algorithm can discover implicit relations between more infrequent items. Combining these two figures, it is easy to conclude that IRRMiner discovers not only the implicit rules between frequent items as the existing indirect association mining approaches do, but also implicit rules between infrequent items. The reason behind this is easy to find by looking at the algorithm design, which is different from most indirect association mining algorithms, which are limited to frequent items only, due to their base (frequent association mining). However, our algorithm goes beyond such a base, and it is not necessary for the items to be frequent.

C. Rule Size and Number Comparison

To make a fair comparison, the corresponding constraints in IARMiner and IRRMiner are set to be equivalent to each other. Specifically, the common parameter minsup(t_r) is set to 10%, 1%, 15%, and 1.5% empirically on ERD, Bookcross, MovieLens_1, and MovieLens_2, respectively, both in IARMiner and IRRMiner algorithms. Both minIRS and t_μ are set to 0 as it is only in this case that these two constraints are essentially equivalent to each other as proved below.

Proof: Suppose implicitly related itemset $I = \{x, y\}$, itemset M is a link itemset to connect x and y. According to (2) and (6)–(8), minIRS $= 0$ indicates IRS$(x, y) \geq 0$, which means there exists at least one link itemset H_I which makes CIRS$(I|H_I) \geq 0$. Suppose CIRS$(I|M) \geq 0$, according to (6), PMI$(x, M) \geq 0$ and PMI$(y, M) \geq 0$, namely $p(x, M) \geq p(x)p(M)$ and $p(y, M) \geq p(y)p(M)$. On the other hand, according to the definition of $\mu(x, M)$ in [14], $t\_\mu = 0$ means $\mu(x, M) = ([p(Mx) - p(M)p(x)]/[p(Mx)(1 - p(x))]) \geq 0$, which also indicates $p(x, M) \geq p(x)p(M)$, for the same reason, $p(y, M) \geq p(y)p(M)$. So the key constraints IRS$(x, y) \geq$ minIRS in IRRMiner and $\mu(x, M) \geq t\_\mu$ in IARMiner are actually the same when their thresholds are set to 0. ■

The size of an implicit rule "$r : i_1 \oplus i_2 \oplus \cdots i_j | H_I(I = \{i_1, i_2 \cdots i_j\})$" is defined as the size of itemset I, which measures how many items are implicitly related conditioned on H_I. Formally

$$\text{Size}(r) = \left|\left\{i_1, i_2 \cdots i_j\right\}\right|. \tag{10}$$

Table III shows the number of indirect association rules mined by IARMiner (IAR. for short) and implicit rules mined by IRRMiner (IRR). Two main conclusions can be drawn from this.

1) IRRMiner can mine more size-2 rules than IARMiner (i.e., 4243 versus 2226 on the ERD dataset). By checking the rules more deeply, we find the rules mined by IARMiner are a subset of rules from IRRMiner. This is because IRRMiner targets not only the implicit rules between frequent items but also the infrequent ones, as stated in the introduction. Therefore, IRRMiner can cover more items and generate more rules, which is consistent with the results of the coverage comparison in Part B.
2) IRRMiner can mine implicit rules with a size larger than 2 while IARMiner cannot. Theoretically, IRRMiner can output rules as large as the number of items in the transaction as long as the dataset supports such rules. Note that rules with a size larger than 4 are not shown due to space limitations. Thanks to the design behind IRRMiner, which considers the implicit relations among all possible items rather than just the indirect association between a pair of frequent items as IARMiner does, the complex implicit relations among multiple items discovered by IRRMiner are more general and more consistent with the real-world cases compared to the indirect association rules of size-2. In summary, IRRMiner goes far beyond IARMiner by returning more rules of larger sizes. Implicit rules of a large size reveal much more hidden information between multiple items than size-2 rules which only reflect pairwise relations between every two items. Taking a sample from the ERD dataset as an example, the triad implicit relation among three books *An Introduction to Secondary Data Analysis*, *Python Data Science Handbook*, and *Microsoft Excel 2013 Data Analysis* is richer than the pair-wised indirect association between any two of them.

Essentially, the indirect association rule mining framework is a special case of our proposed framework. When we only focus on the implicit relations between frequent items and limit

TABLE III
NUMBER OF MINED RULES BY IARMINER (IAR.) AND IRRMINER (IAR.)

		Size 2	Size 3	Size 4
ERD	IAR.	2226	-	-
	IRR.	4243	50405	336216
Bookcross	IAR.	82	-	-
	IRR.	156	221	342
MovieLens_1	IAR.	8495	-	-
	IRR.	15219	203656	1926018
MovieLens_2	IAR.	908	-	-
	IRR.	1689	20792	109800

the rule size to 2, our framework is simplified to the existing indirect association mining one, and can mine the same rules as indirect association mining does.

D. Run Time Comparison

To evaluate the efficiency of the developed IRRMiner algorithm, two sets of experiments are organized. One is to compare the run time of IARMiner and IRRMiner to mine the same size-2 rules, in which both the rules and rule numbers resultant from both algorithms are exactly the same. The other is to compare the run time of both algorithms to mine the same number of size-2 rules, where the rules may not be completely identical.

Recall that IRRMiner can be simplified to indirect association mining algorithms and can mine the same rules as them, as discussed in the last paragraph in Part C. We add an extra frequency constraint on IRRMiner and limit the rule size to 2 to make sure it only discovers the same rules or the same number of rules as IARMiner. We keep the values of corresponding parameters the same in both algorithms for a fair comparison, namely minIRS and minsup in IRRMiner are equal to t_μ and minsup in IARMiner, respectively. In addition, we keep both minIRS and t_μ unchanged (minIRS $= t\_\mu = 0$, according to the proof in Part C) while change minsup in two algorithms synchronously in the first set of experiments; in the second set of experiments, the minsup in IARMiner and IRRMiner are kept equal and unchanged (empirically 10% in ERD, 1% in Bookcross, 15% in MovieLens_1, and 1.5% in MovieLens_2 for both algorithms) while minIRS in IRRMiner and t_μ in IARMiner are adjusted accordingly to mine the same number of rules. The results of these two sets of experiments are given in Tables IV and V, respectively, in which the symbol "*" represents the time spent by IARMiner under certain t_μ values. Note that in the second set of experiments, to achieve the identical number of rules, t_μ in IARMiner does not necessarily need to be equal to minIRS in IRRMiner.

It is clear that, in mining either the same implicit size-2 rules in Table IV or the identical number of size-2 implicit rules in Table V, our proposed IRRMiner is much more efficient than IARMiner. The run time is reduced by around 80% on the ERD dataset and around 90% on the other three datasets by IRRMiner, compared to IARMiner.

One main contribution to efficiency improvement is the first step (explicit dependency discovery) in our proposed IRRMiner, which only exists in our algorithm. It checks whether an item has dependent itemsets; items without dependent itemsets are not considered in the subsequent steps. Many items without dependent itemsets but are frequent are filtered out in an early stage. However, such items cannot be removed in the beginning of IARMiner which uses a support threshold to filter out nonfrequent items in its first step. This partly explains why IRRMiner is clearly more efficient than IARMiner.

The time complexity analysis of Algorithm 1 is detailed in this paragraph. Assume the total number of items in the transactional matrix is N. Given an implicit rule mining task, the total process of the IRRMiner algorithm is divided into two stages: 1) the preparation stage illustrated by lines 1 and 2) the implicit rule mining stage described from line 2 to the end of the algorithm. Please note that the preparation stage is a preprocessing operation, once it is ready, various sizes of implicit rules under different minIRS values can be mined without the need to conduct the preparation stage again, which means it is not necessary to run line 1 every time when we mine implicit rules. Hence, the run time of IRRMiner mainly depends on the second stage. Specifically, suppose a percentage of α of all the N items have dependent itemsets, so $C_{\alpha N}^2$ possible size-2 itemsets will be generated in line 2, which result in $C_{\alpha N}^2 = ([(\alpha N) * (\alpha N - 1)]/2)$ times of computation from lines 3 to 7, accordingly, the time complexity of these lines is $O((\alpha N)^2)$. Meanwhile, at most $C_{\alpha N}^2$ hidden dependent itemsets will be selected in line 5, which results in the maximum computational times being also $C_{\alpha N}^2$ from lines 8–15. The time complexity of lines 8–15 is $O((\alpha N)^2)$ too. Overall, the time complexity of the implicit rule mining stage is $O((\alpha N)^2)$ in mining size-2 implicit rules. Similarly, the time complexity of mining size-3, size-4, ..., etc. implicit rules are $O((\alpha N)^4)$, $O((\alpha N)^8), \ldots$, etc. For the preparation stage, when we set the maximum length of link itemset to 1, 2, 3, etc., the time complexity is $O((\alpha N))$, $O((\alpha N)^2)$, $O((\alpha N)^4)$, etc. In IARMiner, the time complexity is $O((\beta N)^3)$, $O((\beta N)^4)$, $O((\beta N)^6)$, etc., when the maximum length of the mediate itemset is set to 1, 2, 3, etc., where parameter β is the percentage of frequent items with respect to all items. To summarize, when mining size-2 rules, the time complexity of IRRMiner is $O((\alpha N)^2)$, compared to $O((\beta N)^3)$, $O((\beta N)^4)$, $O((\beta N)^6)$, etc., of IARMiner if we do not consider the preparation stage of IRRMiner. Even if we take the preparation into consideration, the time complexity of IRRMiner is $O((\alpha N)^2)$, $O((\alpha N)^2)$, $O((\alpha N)^4)$ compared to $O((\beta N)^3)$, $O((\beta N)^4)$, $O((\beta N)^6)$ of IARMiner, respectively, by setting the maximum length of the dependent itemsets in IRRMiner to the same as that of the mediate itemset in IARMiner. Accordingly, IRRMiner reduces the time complexity of $O((N))$ and $O(N^2)$, respectively, when we set the maximum length of link itemset to 1 and larger than 1 with the consideration of preparation. This explains why IRRMiner is always much more efficient than IARMiner, especially in a dataset with a large number of items. This is consistent with the empirical results in Tables IV and V.

E. Data Factor Test

To test the performance of the proposed algorithm on datasets of different characteristics, we conduct data factor test.

TABLE IV
RUN TIME (IN SECOND) UNDER DIFFERENT *minsup*(*m.s.*)

ERD			Bookcross			MovieLens_1			MovieLens_2		
	IAR.	IRR.		IAR.	IRR.		IAR.	IRR.		IAR.	IRR.
$m.s.$=6%	9.4	**2.4**	$m.s.$=0.6%	474.5	**5.31**	$m.s.$=13%	1130	**24.5**	$m.s.$=1.3%	600	**5.74**
$m.s.$=10%	16.6	**2.6**	$m.s.$=1%	511.6	**5.21**	$m.s.$=15%	1162	**24.6**	$m.s.$=1.5%	641	**5.75**
$m.s.$=14%	20.8	**2.8**	$m.s.$=1.4%	518.6	**5.18**	$m.s.$=17%	1192	**25**	$m.s.$=1.7%	700	**5.79**

TABLE V
RUN TIME (IN SECOND) UNDER DIFFERENT *minIRS*(*m.I.*)

	ERD		Bookcross		MovieLens_1		MovieLens_2	
	IAR.	IRR.	IAR.	IRR.	IAR.	IRR.	IAR.	IRR.
$m.I.$=0	16.6*	**2.6**	511.6*	**5.21**	1162*	**24.6**	641*	**5.75**
$m.I.$=0.8	15.5*	**2.4**	510.2*	**5.2**	1080*	**24**	552*	**5.66**
$m.I.$=1.6	13.8*	**2.3**	481.7*	**5.17**	964*	**23.3**	492*	**5.63**

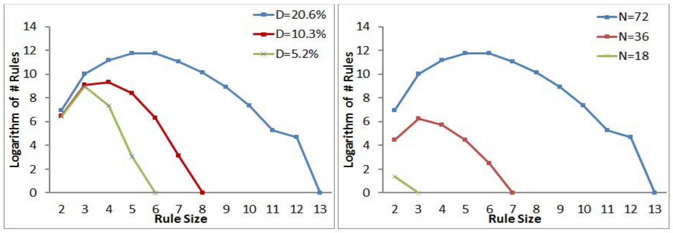
Specifically, two data factors: density (D) and the total number of items (N) are selected. Recall the number of possible size-2, size-3, etc., itemsets are $C^2_{\alpha N}$, $C^3_{\alpha N}$, etc., respectively, as illustrated in the fourth paragraph of Section V-D, it is obvious that α and N can substantially affect the number of mined implicit rules by first deciding the number of possible generated itemsets and candidate itemsets. Furthermore, α is greatly affected by data density, because the items in a dense dataset are more likely to be dependent on each other than those in a sparse one. In other words, data density closely relates to the number of resultant rules via α.

To make a fair comparison, when one data factor is tested, we ensure the other data factor is identical on all the datasets by conducting necessary processing on them. For example, when testing the effect of density, the number of items in the three datasets used in 1) is kept the same.

1) Density Test: We test the effect of dataset density on the number of implicit rules mined by IRRMiner by running it on three real-world datasets with various density degrees but with the same number of items. The number of obtained rules is shown in Fig. 3(a).

It is quite obvious that under the same experimental settings, the number of rules is significantly influenced by the data density. Fig. 3(a) shows that the denser a dataset is, the more rules are obtained. The number of rules of all sizes for the dataset with a density of 20.6% is much larger than that for the dataset with density of 10.3%, the latter is also much larger than that on the dataset with density of 5.2%. In addition, it is much more likely that larger rules will be obtained for denser datasets, for instance, only the densest dataset ($D = 20.6\%$) results in rules of size larger than 9. This is consistent with the statement in the first paragraph of this section that items in a dense dataset are more likely to depend on each other and lead to a higher α for the dataset, and produce more rules.

2) Item Number Test: We test the effect of item number (N) on the number of implicit rules mined by IRRMiner by running it on three real-world datasets with different number of items but with the same density. The number of rules obtained is shown in Fig. 3(b). The results are also consistent with the analysis in the first paragraph in this section. When the data density is unchanged, larger N implies more candidate itemsets generated and also more rules mined.



(a) (b)

Fig. 3. Rule number comparisons with respect to different data density degrees (D) or item numbers (N). (a) Test on data density. (b) Test on item number.

VI. CASE STUDY: IMPLICIT RULES-BASED RECOMMENDATION

IRRMiner can be applied to different cases, such as product promotion and cross-saling by exploring the implicit relations hidden behind various products. Here, we show how the mined implicit rules assist in pattern-based recommendation [42] to increase recommendation reliability. We first analyze the theoretical benefits and then justify them with real-world case studies. Note that the implicit rules used for recommendations are mined on transactional data, hence we can only make recommendations based on transactional information, which is the typical scenario, where pattern-based recommendation is applicable. This is quite different from the well-known content-based or collaborative filtering [43] based recommendations which are built on the rating data.

One of the most important applications of association rules or correlation rules is to increase product sales by recommending some items associated with the items that a customer has just bought. To apply these rules to the recommendation domain, we introduce the concept of recommendation rules in the form of $X \to Y$ to describe the recommendation strategy, whereby itemset Y is recommended to those consumers who have just bought itemset X. In this case, a direct method to evaluate recommendation quality is to check whether the recommended items have actually been bought by the customers or not. The higher possibility of Y to be bought together with X, the more reliable the recommendation rule $X \to Y$. Based on such observation, the recommendation reliability of a typical recommendation rule $X \to Y$ is defined as the percentage

TABLE VI
MEAN RECOMMENDATION RELIABILITY USING IMPLICIT RULES (IM_RULE) AND CORRELATION RULE (CORE_RULE) WITH RESPECT TO DIFFERENT *minIRS*

		$minIRS = 0$	$minIRS = 0.5$	$minIRS = 1$	$minIRS = 1.5$
ERD	Im_Rule	35.54%	37.5%	38.52%	39.17%
	Core_Rule	33.7%*	35.6%*	36.3%*	36.8%*
MovieLens_1	Im_Rule	49.41%	49.97%	51.8%	54.11%
	Core_Rule	43.98%*	44.45%*	45.97%*	47.72%*

TABLE VII
MEAN RECOMMENDATION RELIABILITY OF TOP-K RULES WHEN *minIRS* = 1

		$All\ rules$	$K = 10$	$K = 5$	$K = 3$
ERD	Im_Rule	38.52%	40.58%	42%	43.2%
	Core_Rule	36.3%*	38.33%*	39.7%*	40.9%*
MovieLens_1	Im_Rule	51.8%	62.89%	63.79%	64.21%
	Core_Rule	45.97%*	53.87%*	55.32%*	55.32%*

of transactions with X and Y included with respect to those including X. Formally

$$\text{Reliability}(X \to Y) = \frac{\#\text{transactions including } X \text{ and } Y}{\#\text{transactions including } X}. \tag{11}$$

Usually, different patterns lead to different recommendation strategies and recommendation rules in pattern-based recommendation [44], which result in different levels of recommendation reliability. Note that here, *pattern* is a general concept; implicit rules, association rules, and correlation rules [45] are all specific forms of patterns. To be specific, association rule-based recommendation suggests those explicitly associated items; correlation rule-based recommendation suggests explicitly correlated items; while implicit rule-based recommendation makes recommendations by considering not only explicit dependency but also implicit dependency between items. Since our mined implicit rules are built on the basis of dependency between items, which is similar to correlation rules, we compare implicit relation-based recommendation with correlation rule-based recommendation in terms of reliability. Next, we analyze the recommendation strategies in the form of recommendation rules based on implicit rules and correlation rules [25], respectively, and then compare these two kinds of rule-based recommendations in terms of reliability.

Suppose x and y are two distinct items while Z is an itemset and an implicit rule $x \oplus y|Z$ is mined among them. Accordingly, a recommendation rule $Z \wedge \neg y \to x$ ($Z \wedge y \not\to x$) is derived for recommending a related item x to those who have just bought itemset Z based on the explicit dependency between x and Z, and at the same time the effect of the implicitly related item y of x is considered. Specifically, when a business plan to promote or recommend item x to a customer, if the customer has bought Z but not y (and x), it can be proceeded. However, if the customer has already bought Z and y, the planed recommendation action should be terminated. In this way, the conditions (antecedents of recommendation rules) to recommend an item are more precise, and accordingly, recommendation reliability is improved. On the other hand, for items x and y and itemset Z, two correlation rules $[x, Z]$ and $[y, Z]$ can be easily obtained due to the strong explicit dependency between $x(y)$ and Z. Here, $[x, Z]$ infers that x and Z are positively correlated. Based on these correlation rules, two recommendation rules $Z \to x$ and $Z \to y$ are derived for recommending correlated item x and y to those who have just bought Z. Technically, such recommendations are less reliable than implicit rule-based recommendations due to the lack of consideration of the implicit relations between items (e.g., the implicit relation between x and y). Unfortunately, both the existing association rule-based recommendation and correlation rule-based recommendation do not take such kind of implicit relations into account.

Taking the ERD data as an example, let us consider under which conditions recommending item *coke* is more reliable. Two recommendation rules r_1: *pizza* → *coke* and r_2: *pizza* ∧ ¬*sprite* → *coke* can be derived based on correlation rules and implicit rules, respectively. The left-hand side of each rule indicates the conditions in which to recommend the right-hand side items. The first rule r_1 infers that once customers buy *pizza* we can recommend *coke* to them while the second rule means when customers have bought *pizza*, we need to check if the other items implicitly related to *coke* have already been bought, if they have not been bought, we would recommend *coke*. In practice, when we go back to the transactions in Table I, it is obvious that the conditions in which to recommend *coke* described by r_2 are more precise and reliable than that of r_1, which can also be illustrated by a higher reliability of 100% of r_2 than a lower reliability of 66.67% of r_1. This shows that taking into account implicit relations between items contributes to reliable recommendations.

In addition to the above theoretical benefits of implicit rules in increasing recommendation reliability, we also calculate the reliability of both implicit rule-based recommendations and correlation rule-based recommendations on two real-world datasets: ERD and MovieLens_1. The results are given in Tables VI and VII, where the mean values of the corresponding recommendation rules based on implicit rules and correlation rules, respectively, are given. Please note that here the parameter minIRS only exists in the implicit rule-based recommendation algorithm (IRRMiner), and under each minIRS value setting, the identical number of implicit rules and correlation rules are selected to compare their average reliability. It is obvious that, under all the minIRS value settings, implicit rule-based recommendation always has higher reliability than the correlation rule-based one, as shown in the first column in Table VI (35.54% for the Im_Rule compared to 33.7% for the Core_Rule), and in the first column in Table VII (49.41% for the Im_Rule compared to 43.98% for Core_Rule). It is clear that the reliability of all implicit rules (shown in the first column) increases with an increase in the minIRS. This is because, the implicit rules with weak implicit relations are filtered out during the increase of minIRS, and fewer and stronger implicit rules are selected. Such strong rules have higher reliability. Also less strong correlation rules are selected to guarantee the identical number of rules as implicit rules, so the reliability of the correlation rules also increases with an increase in minIRS. Please note that the top 10, top 5, and top 3 rules are rarely affected by the increase of minIRS as the IRS of these rules is usually much higher than minIRS.

In addition, some specific recommendation rules based on implicit rules and correlation rules are selected below. To differentiate these from recommendation rules, we use p'_1 and p'_2 to represent two implicit rules mined from the ERD dataset. Based on the relations between the items included in these implicit rules, some recommendation rules based on correlation rules (e.g., r'_{11}, r'_{21}) and implicit rules (e.g., r'_{13}, r'_{23}) are derived. In addition, to show the significant effect of the implicitly related items on recommendation reliability, we also add another rules (r'_{12}, r'_{22}). The name of each item is listed below.

$$\begin{aligned}
&\boldsymbol{R}'_1 \begin{cases} p'_1: i'_{11} \oplus i'_{48} | i'_{12} & \\ r'_{11}: i'_{11} \rightarrow i'_{12} & \text{reliability}(r'_{11}) = 42.4\% \\ r'_{12}: i'_{11} \wedge i'_{48} \rightarrow i'_{12} & \text{reliability}(r'_{12}) = 36.3\% \\ r'_{13}: i'_{11} \wedge \neg i'_{48} \rightarrow i'_{12} & \text{reliability}(r'_{13}) = 45.1\% \end{cases} \\
&\boldsymbol{R}'_2 \begin{cases} p'_2: i'_{19} \oplus i'_{48} | i'_{12} & \\ r'_{21}: i'_{19} \rightarrow i'_{12} & \text{reliability}(r'_{21}) = 41.5\% \\ r'_{22}: i'_{19} \wedge i'_{48} \rightarrow i'_{12} & \text{reliability}(r'_{22}) = 38.69\% \\ r'_{23}: i'_{19} \wedge \neg i'_{48} \rightarrow i'_{12} & \text{reliability}(r'_{23}) = 43.5\% \end{cases}
\end{aligned} \tag{12}$$

i'_{11}: a book titled "An Introduction to Secondary Data Analysis with IBM SPSS Statistics" (Book 1);

i'_{12}: Philips Peripherals SWR2122/27 Retractable USB Cable;

i'_{16}: Tableau software for data analysis;

i'_{19}: a book titled "Python Data Science Handbook: Essential Tools for Working with Data" (Book 2);

i'_{48}: a book titled "Microsoft Excel 2013 Data Analysis and Business Modeling (Introducing)" (Book 3);

i'_{61}: San-Disk Memory Card.

Please note that we target the scenario that, for a given item, in what conditions (described by the antecedent of the recommendation rules) it should be recommended to achieve greater reliability. In practice, a given item can be recommended in all the different conditions described by different rules (e.g., r'_{11}, r'_{12}, and r'_{13}), but usually the market wants to make a recommendation as reliable as possible. It is quite clear, given the same item (e.g., retractable USB cable), more reliable recommendation rules (r'_{13}, r'_{23}) can be achieved if more implicit relations (the relations between Books 3 and 1 and 2, respectively) are taken into consideration. This reflects the common shopping behavior that customers do not prefer to buy two similar items within one transaction. For the other dataset MovieLens_1, the movie name is not given in the source data MovieLens10M, so we do not show the specific rules mined on it.

VII. Conclusion

Identifying implicit relation-based complex rules is demanding but challenging and extends far beyond the traditional rule mining framework, such as association rule, causal rule, and correlation rule mining. In this paper, we have proposed a new approach which first captures the dependency between items and then links those items that share the same dependent items (or itemsets) to infer implicitly related rules. Thanks to the special new structures of implicit rules, the complex relations between multiple items are comprehensively revealed. Experimental results on real-world datasets show that our proposed implicit rule mining algorithm is very promising and can generate implicit rules which cannot be discovered by existing algorithms. Further, it runs much faster than an indirect association rule mining algorithm. Case studies demonstrate that it greatly benefits recommendation by increasing its reliability and reducing the number of redundant recommendations. In the future, we will explore the possibility of incorporating item features into our rule inference framework to reveal low-level intrinsic interitem relations (e.g., similarity). Hopefully, more informative implicit rules can be achieved to better help with real-world business.

References

[1] J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent pattern mining: Current status and future directions," *Data Min. Knowl. Disc.*, vol. 15, no. 1, pp. 55–86, 2007.

[2] J. Li *et al.*, "From observational studies to causal rule mining," *ACM Trans. Intell. Syst. Technol.*, vol. 7, no. 2, p. 14, 2016.

[3] P. Singh, N. R. Pal, S. Verma, and O. P. Vyas, "Fuzzy rule-based approach for software fault prediction," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 5, pp. 826–837, May 2017.

[4] J. Sahoo, A. K. Das, and A. Goswami, "An effective association rule mining scheme using a new generic basis," *Knowl. Inf. Syst.*, vol. 43, no. 1, pp. 127–156, 2015.

[5] E. Sokolova *et al.*, "Causal discovery from medical data: Dealing with missing values and a mixture of discrete and continuous data," in *Proc. 18th Conf. Artif. Intell. Med.*, Pavia, Italy, 2015, pp. 177–181.

[6] B. Singh and S. Jain, "Semicompatibility and fixed point theorems in fuzzy metric space using implicit relation," *Int. J. Math. Math. Sci.*, vol. 2005, no. 16, pp. 2617–2629, 2005.

[7] I. Beg and A. R. Butt, "Fixed point for set-valued mappings satisfying an implicit relation in partially ordered metric spaces," *Nonlin. Anal. Theory Methods Appl.*, vol. 71, no. 9, pp. 3699–3704, 2009.

[8] L. Cao, "Combined mining: Analyzing object and pattern relations for discovering and constructing complex yet actionable patterns," *Interdiscipl. Rev. Data Min. Knowl. Disc.*, vol. 3, no. 2, pp. 140–155, 2013.

[9] P. Spirtes and K. Zhang, "Causal discovery and inference: Concepts and recent methodological advances," *Appl. Informat.*, vol. 3, no. 1, pp. 1–28, 2016.

[10] C. C. Yang, X. Tang, Q. Dai, H. Yang, and L. Jiang, "Identifying implicit and explicit relationships through user activities in social media," *Int. J. Electron. Commerce*, vol. 18, no. 2, pp. 73–96, 2013.

[11] L. Peska and P. Vojtas, "Using implicit preference relations to improve recommender systems," *J. Data Semant.*, vol. 6, no. 1, pp. 15–30, 2017.

[12] L. Cao, "Actionable knowledge discovery and delivery," *Interdiscipl. Rev. Data Min. Knowl. Disc.*, vol. 2, no. 2, pp. 149–163, 2012.

[13] Q. Chen, L. Hu, J. Xu, W. Liu, and L. Cao, "Document similarity analysis via involving both explicit and implicit semantic couplings," in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Paris, France, Oct. 2015, pp. 1–10.

[14] S. Hamano and M. Sato, "Mining indirect association rules," in *Proc. Ind. Conf. Data Min.*, Leipzig, Germany, 2004, pp. 106–116.

[15] T. Herawan, A. Noraziah, Z. Abdullah, M. M. Deris, and J. H. Abawajy, "IPMA: Indirect patterns mining algorithm," in *Advanced Methods for Computational Collective Intelligence*. Heidelberg, Germany: Springer, 2013, pp. 187–196.

[16] Q. Wan and A. An, "Efficient mining of indirect associations using hi-mine," in *Proc. Conf. Can. Soc. Comput. Stud. Intell.*, 2003, pp. 206–221.

[17] C. S. Fleisher and B. E. Bensoussan, *Business and Competitive Analysis: Effective Application of New and Classic Methods*, 2nd ed. Upper Saddle River, NJ, USA: Pearson, 2015.

[18] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. 20th Int. Conf. Very Large Data Bases*, vol. 1215. 1994, pp. 487–499.

[19] J. Pearl, M. Glymour, and N. P. Jewell, *Causal Inference in Statistics: A Primer*. New York, NY, USA: Wiley, 2016.

[20] P. I. Garcia-Nunes, R. M. Souza, and A. E. A. da Silva, "Mental models analysis and comparison based on fuzzy rules: A case study of the protests of June and July 2013 in Brazil," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 8, pp. 2021–2033, Aug. 2017.

[21] J. Leng and P. Jiang, "Mining and matching relationships from interaction contexts in a social manufacturing paradigm," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 2, pp. 276–288, Feb. 2017.

[22] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," *ACM SIGMOD Rec.*, vol. 22, no. 2, pp. 207–216, 1993.

[23] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 1–12, 2000.

[24] G. Pyun, U. Yun, and K. H. Ryu, "Efficient frequent pattern mining based on linear prefix tree," *Knowl. Based Syst.*, vol. 55, pp. 125–139, Jan. 2014.

[25] S. Brin, R. Motwani, and C. Silverstein, "Beyond market baskets: Generalizing association rules to correlations," *ACM SIGMOD Rec.*, vol. 26, no. 2, pp. 265–276, 1997.

[26] E. R. Omiecinski, "Alternative interest measures for mining associations in databases," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 1, pp. 57–69, Jan./Feb. 2003.

[27] Z. Zhou, Z. Wu, C. Wang, and Y. Feng, "Mining both associated and correlated patterns," in *Proc. Int. Conf. Comput. Sci.*, Reading, U.K., 2006, pp. 468–475.

[28] U. K. Rage and M. Kitsuregawa, "Efficient discovery of correlated patterns using multiple minimum all-confidence thresholds," *J. Intell. Inf. Syst.*, vol. 45, no. 3, pp. 357–377, 2015.

[29] J. Pearl, "Causality: Models, reasoning and inference," *Econometric Theory*, vol. 19, pp. 675–685, 2003.

[30] G. F. Cooper, "A simple constraint-based algorithm for efficiently mining observational databases for causal relationships," *Data Min. Knowl. Disc.*, vol. 1, no. 2, pp. 203–224, 1997.

[31] C. Silverstein, S. Brin, R. Motwani, and J. Ullman, "Scalable techniques for mining causal structures," *Data Min. Knowl. Disc.*, vol. 4, nos. 2–3, pp. 163–192, 2000.

[32] C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos, "Local causal and Markov blanket induction for causal discovery and feature selection for classification Part I: Algorithms and empirical evaluation," *J. Mach. Learn. Res.*, vol. 11, no. 1, pp. 171–234, 2010.

[33] J. Li *et al.*, "Mining causal association rules," in *Proc. IEEE 13th Int. Conf. Data Min. Workshops*, Dallas, TX, USA, 2013, pp. 114–123.

[34] L. Chen, S. S. Bhowmick, and J. Li, "Mining temporal indirect associations," in *Proc. Pac.–Asia Conf. Knowl. Disc. Data Min.*, Singapore, 2006, pp. 425–434.

[35] W. Ouyang and Q. Huang, "Mining direct and indirect fuzzy sequential patterns in large transaction databases," in *Proc. Int. Conf. Intell. Comput.*, Qingdao, China, 2007, pp. 180–189.

[36] P. Kazienko, "Mining indirect association rules for Web recommendation," *Int. J. Appl. Math. Comput. Sci.*, vol. 19, no. 1, pp. 165–186, 2009.

[37] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 97–107, Jan. 2014.

[38] L. Cao, S. Y. Philip, C. Zhang, and Y. Zhao, *Domain Driven Data Mining*. New York, NY, USA: Springer, 2010.

[39] K. W. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Comput. Linguist.*, vol. 16, no. 1, pp. 22–29, 1990.

[40] G. Bouma, "Normalized (pointwise) mutual information in collocation extraction," in *Proc. 21st Int. Conf. German Soc. Comput. Linguist. Lang. Technol.*, pp. 31–40, 2009.

[41] F. Role and M. Nadif, "Handling the impact of low frequency events on co-occurrence based measures of word similarity: A case study of pointwise mutual information," in *Proc. 3rd Int. Conf. Knowl. Disc. Inf. Retrieval*, Paris, France, 2011, pp. 226–231.

[42] J. Hiltz-Laforge, R. Y. Nonez, A. Pourshahid, and G. A. Watts, "Pattern-based analysis recommendation," U.S. Patent 14/144 456, Dec. 30, 2013.

[43] J. Sun, S. Wang, B. J. Gao, and J. Ma, "Learning to rank for hybrid recommendation," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manag.*, 2012, pp. 2239–2242.

[44] K. Choi, D. Yoo, G. Kim, and Y. Suh, "A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis," *Electron. Commerce Res. Appl.*, vol. 11, no. 4, pp. 309–317, 2012.

[45] S. Nijssen, T. Guns, and L. De Raedt, "Correlated itemset mining in ROC space: A constraint programming approach," in *Proc. 15th Int. Conf. Knowl. Disc. Data Min.*, Paris, France, 2009, pp. 647–656.

Shoujin Wang (S'16) received the M.Sc. degree in information management and E-commerce from Shanghai University, Shanghai, China, in 2014. He is currently pursuing the Ph.D. degree with the Advanced Analytics Institute, Faculty of Engineering and IT, University of Technology Sydney, Sydney, NSW, Australia.

His current research interests include data mining, machine learning, and recommender system.

Longbing Cao (SM'06) received the Ph.D. degree in pattern recognition and intelligent systems and the Ph.D. degree in computing sciences.

He is currently a Professor with the University of Technology Sydney, Sydney, NSW, Australia, the Founding Director of the Advanced Analytics Institute. His current research interests include data science, big data analytics, data mining, machine learning, behavior informatics, complex intelligent systems, and their enterprise applications.