An Information-Theoretic Approach to Quantitative Association Rule Mining [⋆]

Yiping Ke James Cheng Wilfred Ng

Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong, China
{keyiping, csjames, wilfred}@cse.ust.hk

Abstract. Quantitative Association Rule (QAR) mining has been recognized an influential research problem over the last decade due to the popularity of quantitative databases and the usefulness of association rules in real life. Unlike Boolean Association Rules (BARs), which only consider boolean attributes, QARs consist of quantitative attributes which contain much richer information than the boolean attributes. However, the combination of these quantitative attributes and their value intervals always gives rise to the generation of an explosively large number of itemsets, thereby severely degrading the mining efficiency.

In this paper, we propose an information-theoretic approach to avoid unrewarding combinations of both the attributes and their value intervals being generated in the mining process. We study the mutual information between the attributes in a quantitative database and devise a normalization on the mutual information to make it applicable in the context of QAR mining. To indicate the strong informative relationships among the attributes, we construct a mutual information graph (MI graph), whose edges are attribute pairs that have normalized mutual information no less than a predefined information threshold. We find that the cliques in the MI graph represent a majority of the frequent itemsets. We also show that frequent itemsets that do not form a clique in the MI graph are those whose attributes are not informatively correlated to each other.

[⋆] A preliminary version of this paper [15] appeared as a poster paper in the proceedings of the 22nd International Conference on Data Engineering (ICDE), 2006.

By utilizing the cliques in the MI graph, we devise an efficient algorithm that significantly reduces the number of value intervals of the attribute sets to be joined during the mining process. Extensive experiments show that our algorithm speeds up the mining process by up to two orders of magnitude. Most importantly, we are able to obtain most of the high-confidence QARs, whereas the QARs that are not returned by MIC are shown to be less interesting.

Keywords: Quantitative Databases, Association Rules, Information-Theoretic Approach, Mutual Information

1 Introduction

Quantitative Association Rules (*QARs*) [25] have served as a useful tool in discovering association relationships among sets of attributes in business and scientific domains. In a QAR, attributes are not limited to being boolean but can be either *quantitative*, which are numeric values (e.g., age, salary), or *categorical*, which are enumerations (e.g., gender, brand). Being able to represent a wide variety of real-life attributes, QARs are far more expressive and informative than *Boolean Association Rules* (*BARs*) [2], which are restricted to only boolean attributes. An example QAR in an employee database is $\{\texttt{age[25, 40], gender[female]}\} \Rightarrow \{\texttt{salary[13500, 18700]}\}$ ($\texttt{supp} = 0.03$, $\texttt{conf} = 0.8$). The QAR states that "3% (*support*) of the employees are females aged between 25 and 40 and earning a salary of between $13,500 and $18,700", while "80% (*confidence*) of the female employees aged between 25 and 40 are earning a salary of between $13,500 and $18,700".

The problem of *QAR mining* [25] is: given a database, a *minimum support threshold* and a *minimum confidence threshold*, find all QARs with support and confidence no less than the given thresholds.

Due to the popularity of quantitative databases and the usefulness of association rules in real life, QAR mining has been identified as a long-standing research problem. Many studies [25, 19, 32, 28, 18] have aimed at developing feasible approaches to mining QARs over the last decade. A common approach to the QAR mining problem is to transform it into a problem of conventional BAR mining [2, 3]. The idea is that, for each distinct *value* of a quantitative or categorical attribute, the pair ⟨*attribute*, *value*⟩ is mapped to a boolean attribute, and then algorithms for mining BARs are applied. However, in many cases, the domain of a quantitative attribute is very large and may be continuous. Thus, a *discretization* process [25] is first used to partition the domain of a quantitative attribute into intervals. Then, each ⟨*attribute*, *interval*⟩ pair of the quantitative attribute is mapped to a boolean attribute.

Mining QARs by a generic BAR mining algorithm, however, is infeasible in most cases for the following reasons. First, QAR mining suffers from the same problem of a combinatorial explosion of attribute sets as does BAR mining; that is, given a set of $\mathcal{N}$ distinct attributes, the number of its non-empty subsets is $(2^{\mathcal{N}} - 1)$. In practice, the number of distinct attributes in a QAR mining problem may not be as large as that in a BAR mining problem. However, as shown by Srikant and Agrawal [25], it is necessary to combine the consecutive intervals of a quantitative attribute to gain sufficient support and more meaningful intervals. This leads to another combinatorial explosion problem: if the domain of a quantitative attribute is partitioned into n intervals, the total number of intervals of the attribute grows to $\mathcal{O}(n^2)$ after combining the consecutive intervals. When we join the attributes in the mining process, the number of *itemsets* (i.e., a set of ⟨*attribute*, *interval*⟩ pairs) can become prohibitively large if the number of intervals associated with an attribute is large. For example, it is common in a QAR mining problem that a quantitative attribute has 200 intervals; however, there are $(200 * (200 + 1)/2)^2 = 404,010,000$ different combinations of intervals if we join two such attributes, which is equivalent to 404,010,000 candidate attribute sets in a BAR mining problem. This number further grows exponentially when more than two attributes are joined. As a result, effective techniques to prune the large search space of QAR mining are necessary in order to develop an efficient algorithm for the problem.

In this paper, we adopt an information-theoretic approach to address the two combinatorial explosions in QAR mining by investigating the relationships between the attributes. We first define an *interaction graph* to formally represent the relationships between the attributes in the mining problem. The vertices of the interaction graph correspond to the attributes in the mining problem, while an edge represents a pair of attributes appearing in the same QAR. Thus, the set of attributes that compose a QAR forms a clique (i.e., a complete subgraph) in the interaction graph.

We introduce a framework, called *MIC* (which stands for Mutual Information and Clique), to mine the set of QARs. The MIC framework has three phases. The first phase prepares the database by discretizing the quantitative attributes. In the second phase, we first investigate the *mutual information* between each pair of attributes. Then, we propose a *normalization* on the mutual information. We define a pair of attributes to have a *strong informative relationship* if their *normalized mutual information* is no less than a predefined *minimum information threshold*, μ. We then establish a *Mutual Information graph* (*MI graph*) to represent attributes that have strong informative relationships. We show that the MI graph can retain all or most of the information carried by the interaction graph. Since each frequent itemset is represented by a clique in the interaction graph, the cliques in the MI graph are used in the final phase to facilitate the computation of frequent itemsets as well as to guide the generation of QARs.

Utilizing the cliques in the MI graph greatly alleviates both the combinatorial explosions of attribute sets and intervals in the QAR mining problem. Instead of joining the intervals for all attribute sets in the database, we only need to focus on those attribute sets that form a clique in the MI graph. Therefore, both the number of attribute sets and their intervals to be joined are significantly reduced. Moreover, the attributes in a clique of the MI graph are strongly informatively related as measured by normalized mutual information, thereby ensuring the quality of the QARs obtained.

Our Contribution. We study an information-theoretic approach that addresses the problem of QAR mining. Since the mutual information is able to capture the inherent co-occurrence relationships between the attributes, it is a good indicator for frequent itemsets and hence QARs. By applying the mutual information concept in the context of QAR mining, we effectively prune a large part of the search space that represents the insignificant informative relationships between the attributes. Our extensive experiments show that compared with the state-of-the-art QAR mining algorithm [25], MIC speeds up the mining process by up to two orders of magnitude on both synthetic and real datasets. Most importantly, MIC obtains most of the QARs that have high confidence. We also show that the QARs that are not returned by MIC are insignificant by a formal measure [5] of interestingness for association rules.

Organization. We give some preliminaries on QAR mining in Section 2. We then introduce the concept of interaction graphs in Section 3. In Section 4, we present the overall description of the MIC framework and describe the technical details in each phase of the framework. We give the experimental results in Section 5 and discuss the related work in Section 6. Finally, we conclude our paper in Section 7.

2 Preliminaries

In this section, we present the notions and basic concepts in the QAR mining problem.

Let $\mathcal{I} = \{x_1, x_2, \ldots, x_m\}$ be a set of distinct *attributes* or *random variables*[1]. An attribute can be either *quantitative* or *categorical.* Let $dom(x_j)$ be the domain of an attribute x_j, for $1 \leq j \leq m$. An *item*, denoted as $x[l_x, u_x]$, is an attribute x associated with an *interval* $[l_x, u_x]$, where $x \in \mathcal{I}$ and $l_x, u_x \in dom(x)$. We have $l_x = u_x$ if x is categorical and $l_x \leq u_x$ if x is quantitative. An *itemset* is a non-empty set of items with distinct attributes. Given an itemset X, we define its *attribute set* as $attr(X) = \{x \mid x[l_x, u_x] \in X\}$. An itemset X is called a k-itemset if $|attr(X)| = k$. Accordingly, the attribute set of a k-itemset is called k-attribute set. For brevity, we write an itemset $X = \{x_1[l_{x_1}, u_{x_1}], \ldots, x_k[l_{x_k}, u_{x_k}]\}$ as $x_1[l_{x_1}, u_{x_1}] \cdots x_k[l_{x_k}, u_{x_k}]$.

A *transaction* T is a sequence $\langle v_1, v_2, \ldots, v_m \rangle$, where $v_j \in dom(x_j)$, for $1 \leq j \leq m$. A transaction T *supports* an itemset X if $\forall\ x_i[l_i, u_i] \in X$, $l_i \leq v_i \leq u_i$, where $i \in \{1, \ldots, m\}$. Let $\mathcal{D}$ denote a *quantitative database*, which consists of a set of transactions. The *frequency* of X in $\mathcal{D}$, denoted by $freq(X)$, is the number of transactions in $\mathcal{D}$ that support X. The *support* of X, denoted by $supp(X)$, is the probability that a transaction T in $\mathcal{D}$ supports X, and is defined as $supp(X)$=$freq(X)/|\mathcal{D}|$. X is a *frequent itemset* if $supp(X) \geq \sigma$, where σ ($0 \leq \sigma \leq 1$) is a predefined *minimum support threshold.*

[1] We use the terms *attribute* and *random variable* interchangeably in subsequent discussions.

Table 1. The Employee Database

age	gender	salary	education	service years
23	F	12,000	High School	5
28	M	15,800	Bachelor	3
28	M	17,000	Master	1
30	M	21,300	Master	2
30	F	9,500	High School	1
37	M	28,000	PhD	1
39	M	20,000	Bachelor	8
41	M	36,500	PhD	11
44	M	32,000	Master	15
46	F	15,000	High School	23

A *Quantitative Association Rule* (QAR), r, is an implication of the form $X \Rightarrow Y$, where X and Y are itemsets, and $attr(X) \cap attr(Y) = \emptyset$. X and Y are called the *antecedent* and the *consequent* of r, respectively. We define the attribute set of r as $attr(r) = attr(X) \cup attr(Y)$. The *support* of r is defined as $supp(X \cup Y)$. The *confidence* of r is defined as $conf(r) = supp(X \cup Y)/supp(X)$, which is the conditional probability that a transaction T supports Y, given that T supports X.

Problem Description. Given a database $\mathcal{D}$, a *minimum support threshold* σ ($0 \leq \sigma \leq 1$), and a *minimum confidence threshold* c ($0 \leq c \leq 1$), the QAR mining problem is to find all the QARs with *support* and *confidence* no less than σ and c, respectively.

Note that *Boolean Association Rules* (BARs) [2] are a special case of QARs, where all the attributes are categorical attributes with boolean values.

Example 1 Table 1 shows an employee database having ten transactions. $\mathcal{I} = \{$`age`, `gender`, `salary`, `education`, `service years`$\}$, among which `age`, `salary` and `service years` are quantitative attributes. An example item is $\texttt{age}[25, 30]$. And $\texttt{age}[25, 30]\texttt{gender}[M, M]$ is a 2-itemset with frequency 3 and with support $3/10 = 0.3$. Given $\sigma = 0.3$ and $c = 0.6$, $\texttt{age}[25, 30] \Rightarrow \texttt{gender}[M, M]$ is a QAR since $\mathit{supp}(\texttt{age}[25, 30]\texttt{gender}[M, M]) = 0.3 \geq \sigma$ and $\mathit{conf}(\texttt{age}[25, 30] \Rightarrow \texttt{gender}[M, M]) = \frac{\mathit{supp}(\texttt{age}[25,\ 30]\texttt{gender}[\mathrm{M,\ M}])}{\mathit{supp}(\texttt{age}[25,\ 30])} = \frac{0.3}{0.4} = 0.75 \geq c$. □

3 Interaction Graph

In this section, we define an *interaction graph* to model the set of QARs obtained by QAR mining. Given a QAR mining problem $\mathcal{P}$, the *interaction graph* is defined as an undirected graph $G_I = (V_I, E_I)$, where the set of vertices $V_I = \mathcal{I}$, and the set of undirected edges $E_I = \{(x_i, x_j) \mid \exists\ r \in \mathit{Rules}(\mathcal{P})$ such that $x_i, x_j \in \mathit{attr}(r)\}$. Herein, $\mathit{Rules}(\mathcal{P})$ denotes the set of all QARs in $\mathcal{P}$. Thus, the interaction graph is a graph representation of $\mathit{Rules}(\mathcal{P})$.

According to the definition of G_I, for every rule r in $\mathit{Rules}(\mathcal{P})$, the attribute set $\mathit{attr}(r)$ corresponds to a clique (i.e., a complete subgraph) in G_I, since every pair of attributes in $\mathit{attr}(r)$ defines an edge. Thus, given G_I, we can obtain the set of all frequent itemsets by finding all the cliques in G_I and verifying whether the support of the corresponding itemsets satisfies the minimum support threshold. Then, we can restore all the QARs based on the frequent itemsets.

The interaction graph represents the relationships between attributes in a QAR mining problem. Thus, if we can obtain the interaction graph prior to performing QAR mining, we can restrict the search space to a much smaller one that encompasses all QARs. More specifically, by finding the cliques in the interaction graph, we can derive the set of attributes which is the attribute set of some QARs. Based on the attribute sets, we further find the qualified interval sets to produce the QARs. We show that most of the relationships of the attributes reflected in the interaction graph can be acquired by establishing a mutual information graph in the next section.

4 The MIC Framework

In this section, we introduce a framework, called *MIC*, for mining QARs. The MIC framework seamlessly incorporates the *mutual information* concept from information theory [24] into the context of QAR mining. We first give an overall description of the framework and then elaborate on the techniques in each phase.

4.1 Overall Description

There are three main phases in the MIC framework, as shown in Figure 1.

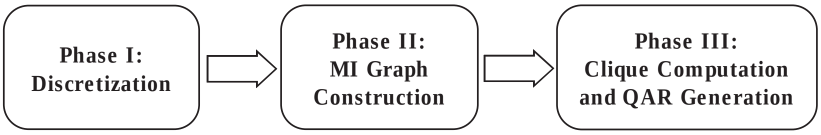


Fig. 1. Three Phases of the MIC Framework

- **Phase I: Discretization.** The domain of each quantitative attribute is partitioned into a set of *base intervals*.
- **Phase II: MI Graph Construction.** Based on the discretized database obtained in Phase I, we compute the normalized mutual information of the attributes. Then, we construct a mutual information graph G_{MI} that represents the strong informative relationships between the attributes.

- **Phase III: Clique Computation and QAR Generation.** We utilize the cliques in G_{MI} to compute frequent itemsets and to guide the generation of QARs.

Here, we briefly introduce Srikant and Agrawal's Mining approach [25] (denoted as *SAM*), which is the state-of-the-art QAR mining approach. Later in experiments in Section 5, we compare the performance of our MIC to that of SAM.

We can view the QAR mining problem at two conceptual levels: the *attribute level* that consists of the attributes and the *interval level* that consists of the corresponding intervals of the attributes. SAM directly operates on the interval level throughout the entire mining process. In other words, the pruning by the *Apriori* property is performed on the intervals of the attributes. On the contrary, MIC performs pruning first at the attribute level. Pruning at the attribute level results in substantial pruning at the interval level and hence significant performance improvement, because once the attribute set is pruned, none of the intervals associated with the attribute set is considered in the subsequent mining process. However, pruning at the attribute level is a challenging problem since pruning an attribute set mistakenly will miss all frequent itemsets and QARs that are generated from the attribute set.

MIC applies the concept of mutual information to perform pruning at the attribute level. Mutual information captures the informative relationships between the attributes, which have an implication for the frequent itemsets and the QARs. All pairs of attributes that do not have a strong informative relationship are not chosen to form an itemset and consequently all their intervals are also pruned. Meanwhile, MIC also performs pruning at the interval level using the Apriori property as does SAM. Thus, the search space of MIC is significantly smaller than that of SAM.

The pruning at the attribute level in MIC may miss some QARs in the mining result. However, as evidenced by our experiments, MIC obtains most of the QARs that are of high confidence and we also show that the missing QARs are of very low interest [5], because the attributes in the same QAR are informatively related to each other.

4.2 Phase I: Discretization

This phase is a preprocessing step in the mining process. The purpose of discretization is to map a large number of distinct values of a quantitative attribute to a smaller set of intervals to deal with the continuous domain and to speed up the mining process. In this phase, the domain of a quantitative attribute is partitioned into a set of n consecutive intervals, called *base intervals*. The base intervals are then labeled with a set of consecutive integers, $\{1, 2, \ldots, n\}$, such that the order of the base intervals is preserved. During the mining process, each base interval is considered as an indivisible unit, while consecutive base intervals may be combined into larger intervals. We also map the values of a categorical attribute to a set of consecutive integers. Thus, the raw values of the attributes are transparent to the mining algorithm in subsequent phases.

The discretization phase is a common preprocessing method in the QAR mining problem [25, 32, 28, 29]. If the domain knowledge for the meaningful attribute intervals is available, the database can be prepared according to the domain knowledge and then it passes through Phases II and III of the MIC framework to mine the QARs. However, in most cases, the domain knowledge is hard to obtain, which is the situation we consider in this paper. While a detailed discussion of discretization is not the focus of this paper, we remark that any discretization technique can be applied in this phase of the MIC framework. Here, we limit our discussion to the *equidepth* discretization technique used in SAM [25], which we compare with our approach. The equidepth discretization technique is proved to minimize the information loss caused by discretization in [25]. Equidepth partitions the domain of a quantitative attribute into n base intervals so that the number of transactions in each base interval is roughly the same. Note that the discretization is an information-lossy transformation; therefore, the number of base intervals n is an important factor since it determines the degree of information loss due to discretization. The larger the n, the less the information loss but the higher the computational cost to mine QARs. A smaller n results in more information loss. The following example helps to illustrate the idea of equidepth discretization.

Table 2. `Age`

Base Interval	Label
[23, 28]	1
[30, 39]	2
[41, 46]	3

Table 3. `Gender`

Value	Label
M	1
F	2

Table 4. `Salary`

Base Interval	Label
[9,500, 15,000]	1
[15,800, 20,000]	2
[21,300, 36,500]	3

Table 5. `Education`

Value	Label
High School	1
Bachelor	2
Master	3
PhD	4

Table 6. `Service Years`

Base Interval	Label
[1, 1]	1
[2, 8]	2
[11, 23]	3

Example 2 Given the employee database in Table 1 and using the equidepth discretization method, the quantitative attributes, `age`, `salary` and `service years`, are discretized into three base intervals, each with 3 or 4 (≈ 10/3) transactions. Tables 2-6 show the base intervals (or values) of the five attributes and their corresponding labels. The discretized employee database is shown in Table 7. □

Table 7. The Discretized Employee Database

age	gender	salary	education	service years
1	2	1	1	2
1	1	2	2	2
1	1	2	3	1
2	1	3	3	2
2	2	1	1	1
2	1	3	4	1
2	1	2	2	2
3	1	3	4	3
3	1	3	3	3
3	2	1	1	3

4.3 Phase II: Mutual Information Graph Construction

In this section, we discuss in detail how we apply the concepts of entropy and mutual information that originates from information theory [24] in the context of QAR mining.

4.3.1 Entropy and Mutual Information

Notation. Let x and y be two random variables. Given $v_x \in dom(x)$ and $v_y \in dom(y)$, we denote the probability parameters as follows:

- $p(v_x)$: the probability of x taking the value v_x.
- $p(v_x, v_y)$: the joint probability of x taking the value v_x and y taking the value v_y.
- $p(v_y|v_x)$: the conditional probability of y taking the value v_y given that x takes the value v_x. It is defined as $p(v_y|v_x) = p(v_x, v_y)/p(v_x)$.

In the QAR mining context, $p(v_x) = supp(x[v_x, v_x])$, $p(v_x, v_y) = supp(x[v_x, v_x]y[v_y, v_y])$ and $p(v_y|v_x) = conf(x[v_x, v_x] \Rightarrow y[v_y, v_y])$.

Entropy. Entropy is a central notion in information theory [24], which measures the uncertainty in a random variable. Entropy and mutual information are closely related and we use entropy to interpret many of the fundamental properties of mutual information and to elaborate the semantics of normalized mutual information. The *entropy* of a random variable x, denoted as $H(x)$, is defined as

$$H(x) = -\sum_{v_x \in dom(x)} p(v_x) \log p(v_x). \tag{1}$$

The *conditional entropy* of a random variable y given another variable x, denoted as $H(y|x)$, is defined as

$$H(y|x) = -\sum_{v_x \in dom(x)} \sum_{v_y \in dom(y)} p(v_x, v_y) \log p(v_y|v_x). \tag{2}$$

Since probabilities are defined in the range of $[0, 1]$, we have $H(x) \geq 0$ and $H(y|x) \geq 0$.

We use the following example to illustrate the application of entropy in the context of QAR mining.

Example 3 Consider the discretized database in Table 7, $H(\texttt{gender}) = -p(1) \cdot \log(p(1)) - p(2) \cdot \log(p(2)) = -0.7 \times \log(0.7) - 0.3 \times \log(0.3) = 0.88$. Similarly, we compute $H(\texttt{education}) = 1.97$. Thus, the attribute `education` exhibits a greater degree of uncertainty than `gender`, since $H(\texttt{education}) > H(\texttt{gender})$. Intuitively, we can say that we are more certain about the value of a `gender` instance than that of an `education` instance.

We can also compute $H(\texttt{gender}|\texttt{education}) = 0$, which indicates that given `education`, there is no uncertainty in `gender`. This may not be true in reality; however, in our designated database as shown in Table 1, given the education of an employee, we can determine his/her gender. In contrast, we cannot determine `education` given `gender` since

$H(\texttt{education}|\texttt{gender}) = 1.09 > 0$, although we are now more certain about `education` as indicated by $H(\texttt{education}|\texttt{gender}) < H(\texttt{education})$. □

Mutual Information. Mutual information describes how much information one random variable tells about another one. The *mutual information* of two random variables x and y, denoted as $I(x; y)$, is defined as

$$I(x; y) = \sum_{v_x \in dom(x)} \sum_{v_y \in dom(y)} p(v_x, v_y) \log \frac{p(v_x, v_y)}{p(v_x)p(v_y)}. \quad (3)$$

An important interpretation of mutual information comes from the following property.

Property 1 $I(x; y) = H(x) - H(x|y) = H(y) - H(y|x)$.

From Property 1, the information that y tells us about x is the reduction in uncertainty about x due to the knowledge of y, and similarly for the information that x tells about y. The greater the value of $I(x; y)$, the more information x and y tell about each other.

Example 4 Consider the discretized database presented in Table 7. According to Equation (3), we have the expression $I(\texttt{gender}; \texttt{education}) = \sum_{v_x \in \{1,2\}} \sum_{v_y \in \{1,2,3,4\}} p(v_x, v_y) \log \frac{p(v_x, v_y)}{p(v_x)p(v_y)} = 0.88$. By Example 3, we verify $I(\texttt{gender}; \texttt{education}) = H(\texttt{gender}) - H(\texttt{gender}|\texttt{education}) = 0.88 - 0 = 0.88$. Since we know for certain the value of `gender` given the value of `education`, the information that `education` tells us about `gender` is just the information that `gender` itself carries. We can also verify that $I(\texttt{gender}; \texttt{education}) = H(\texttt{education}) - H(\texttt{education}|\texttt{gender}) = 1.97 - 1.09 = 0.88$, which shows that the knowledge of `gender` causes a reduction of 1.09 in the uncertainty about `education`. □

We first explore some properties of mutual information that are used to develop a normalization for mutual information. Detailed proof of the properties can be found in [8].

Property 2 $I(x;y) = I(y;x)$.

Property 2 suggests that mutual information is *symmetric*, which means that the amount of information x tells about y is the same as that y tells about x.

Property 3 $I(x;x) = H(x)$.

Property 3 states that the mutual information of a random variable x with itself is the entropy of x. Thus, the entropy is also called *self-information*.

Property 4 $I(x;y) \geq 0$.

Property 4 gives the lower bound for mutual information. When $I(x;y) = 0$, $p(v_x, v_y) = p(v_x)p(v_y)$, which means that x and y are independent, that is, x and y tell us nothing about each other.

Property 5 $I(x;y) \leq H(x)$ and $I(x;y) \leq H(y)$.

Property 5 gives the upper bound for mutual information.

4.3.2 Normalized Mutual Information

Let $\mathcal{M}$ be a measure used to evaluate *the strongness of the relationship* between two attributes in a QAR mining problem. Given a predefined threshold μ, if $\mathcal{M} \geq \mu$, we say that the two attributes are *strongly related* to each other; otherwise, we say that they are not strongly related. Ideally, $\mathcal{M}$ is a measure being able to identify attributes that do not constitute any significant QARs. Thus, we do not need to consider joining these attributes to produce candidate frequent itemsets in the mining process.

Defining $\mathcal{M}$ as the mutual information between the attributes seems to be an ideal approach because mutual information, by definition, naturally measures the information that one attribute tells about another. For two attributes appearing in the same QAR, the strongness of their relationship is reflected by their mutual information.

However, we find that there are two crucial problems in the application of mutual information as such a measure of $\mathcal{M}$. The first is in reference to Property 5, $I(x; y) \leq min(H(x), H(y))$, which means that the mutual information of two attributes x and y is bounded by the minimum of their entropy. Since the entropy of different attributes varies greatly in most cases, the threshold μ cannot be determined globally so that it fits all attributes. For example, if we set $\mu = 1$ in Example 4, we will not join `gender` with `education` since $I(\texttt{gender}; \texttt{education}) = 0.88 < 1$. However, 0.88 is the greatest mutual information between `gender` and any other attributes, which is locally maximum. Therefore, it is very likely that joining `gender` and `education` will produce some frequent itemsets. But if μ is smaller, we may include some pairs of attributes that do not constitute any significant QARs. They are included just because their mutual information is globally large compared with that of other attributes, even though locally their mutual information is relatively small.

Second, Property 4 states that the mutual information of two attributes is a non-negative value, while a greater value indicates more information one attribute tells about the other. However, there is no unified scale for the mutual information measure. Thus, the threshold μ cannot intuitively reflect the amount of information that one attribute tells about another. This is a problem since we cannot tell how strong the relationship between the attributes is. For example, if we set the minimum confidence threshold at 0.9, we know that the QARs obtained are of high quality. However, if we set μ at 0.9, we do not know how much information the number "0.9" amounts to unless it is mapped to a unified scale.

To tackle the above-mentioned problems, we propose a normalization for mutual information.

Definition 1 (Normalized Mutual Information) The *normalized mutual information* of two attributes x and y, denoted as $\widetilde{I}(x;y)$, is defined as

$$\widetilde{I}(x;y) = \frac{I(x;y)}{I(x;x)}. \tag{4}$$

Our idea is to normalize the mutual information between x and y by the maximal value of mutual information between x and another attribute, which is $I(x;x) = H(x)$. As a result, we can get rid of the localness and make the normalized mutual information a global measure. Now, we present some useful properties of the normalized mutual information.

Property 6 $\widetilde{I}(x;y) \neq \widetilde{I}(y;x)$ if $I(x;x) \neq I(y;y)$.

Proof. From Definition 1, $\widetilde{I}(y;x) = \frac{I(y;x)}{I(y;y)}$. It follows from Property 2 that $I(x;y) = I(y;x)$. Hence, if $I(x;x) \neq I(y;y)$, then $\widetilde{I}(x;y) \neq \widetilde{I}(y;x)$. □

Property 6 shows that, unlike mutual information, normalized mutual information is not symmetric.

Property 7 $0 \leq \widetilde{I}(x;y) \leq 1$.

Proof. Since $I(x;x) \geq 0$ and $I(x;y) \geq 0$, $\widetilde{I}(x;y) \geq 0$. It follows by Properties 3 and 5 that $\widetilde{I}(x;y) \leq 1$. □

This property ensures that the value of normalized mutual information falls within the unit interval $[0,1]$.

Property 8 $\widetilde{I}(x;y) = \frac{H(x) - H(x|y)}{H(x)}$.

Proof. By Properties 1 and 3, we have $I(x;y) = H(x) - H(x|y)$ and $I(x;x) = H(x)$. It follows from Definition 1 that $\widetilde{I}(x;y) = \frac{I(x;y)}{I(x;x)} = \frac{H(x) - H(x|y)}{H(x)}$. □

Property 8 suggests the semantics of the normalized mutual information between x and y, which is *the percentage of reduction in uncertainty about x due to the knowledge of y*.

Thus, normalized mutual information gives the threshold μ an intuitive meaning and makes it relatively independent of specific attributes. Now the threshold μ indicates the minimum percentage of reduction in uncertainty about an attribute due to the knowledge of another attribute. We further illustrate this important point by the following example.

Example 5 In Example 4, when we say that the knowledge of `gender` causes a reduction of 1.09 in the uncertainty about `education`, we have little idea how much a reduction of 1.09 is. Now, we compute the normalized mutual information $\widetilde{I}(\texttt{education};\ \texttt{gender}) = \frac{I(\texttt{education; gender})}{H(\texttt{education})} = \frac{0.88}{1.97} = 0.45$, which implies a reduction of 45%. Similarly, we can also compute $\widetilde{I}(\texttt{gender};\ \texttt{education}) = \frac{I(\texttt{gender; education})}{H(\texttt{gender})} = \frac{0.88}{0.88} = 1.00$ and $\widetilde{I}(\texttt{age};\ \texttt{service years}) = \frac{\mathrm{I}(\texttt{age; service years})}{\mathrm{H}(\texttt{age})} = \frac{0.90}{1.57} = 0.57$.

We note that $I(\texttt{gender};\ \texttt{education}) < I(\texttt{age};\ \texttt{service years})$ but $\widetilde{I}(\texttt{gender};\ \texttt{education}) > \widetilde{I}(\texttt{age};\ \texttt{service years})$. This means that the percentage of uncertainty reduction of `gender` due to the knowledge of `education` is higher than that of `age` due to the knowledge of `service years`, although the mutual information of the former is smaller than that of the latter. This shows the advantage of using normalized mutual information.

We list the values of mutual information and normalized mutual information for all the attribute pairs in Table 8 to show clearly the change in mutual information after normalization. □

4.3.3 Mutual Information Graph Construction

Given a predefined *minimum information threshold* μ, we say that a pair of attributes, x_i and x_j, have a *strong informative relationship* with each other if $\widetilde{I}(x_i; x_j) \geq \mu$.

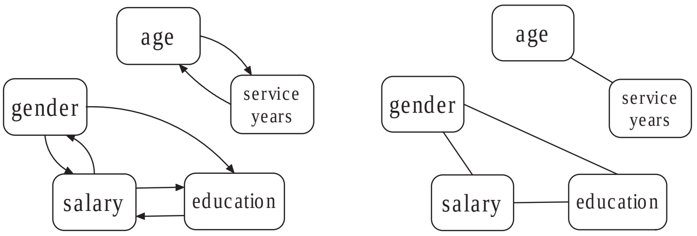
Table 8. The Mutual Information of Two Attributes Before and After Normalization

attribute x	attribute y	Mutual Information	Normalized Mutual Information
age	age	1.57	1.00
age	gender	5.80×10^{-3}	3.69×10^{-3}
age	salary	0.42	0.27
age	education	0.22	0.14
age	service years	0.90	0.57
gender	age	5.80×10^{-3}	6.58×10^{-3}
gender	gender	0.88	1.00
gender	salary	0.88	1.00
gender	education	0.88	1.00
gender	service years	5.80×10^{-3}	6.58×10^{-3}
salary	age	0.42	0.27
salary	gender	0.88	0.56
salary	salary	1.57	1.00
salary	education	1.30	0.82
salary	service years	0.22	0.14
education	age	0.22	0.11
education	gender	0.88	0.45
education	salary	1.30	0.66
education	education	1.97	1.00
education	service years	0.42	0.21
service years	age	0.90	0.57
service years	gender	5.80×10^{-3}	3.69×10^{-3}
service years	salary	0.22	0.14
service years	education	0.42	0.27
service years	service years	1.57	1.00

Given a QAR mining problem, we construct a *Mutual Information graph* (*MI graph*), which is a directed graph, $G_{MI} = (V_{MI}, E_{MI})$, where the set of vertices $V_{MI} = \mathcal{I}$ and the set of directed edges $E_{MI} = \{(x_i, x_j) \mid$

$\widetilde{I}(x_i; x_j) \geq \mu\}$. Thus, the MI graph retains and represents the strong informative relationships between the attributes in a QAR mining problem.

Example 6 Given the employee database in Table 7 and $\mu = 0.5$, we construct the corresponding G_{MI} as shown in Figure 2(a). For example, the attribute pair (`age`, `service years`) forms an edge because the uncertainty of `age` is reduced by more than half ($0.57 > \mu$) given the knowledge of `service years`. In other words, if we know the value of `service years`, we can infer the value of `age` with a higher accuracy. □



(a) G_{MI}

(b) $\hat{G}_{MI}$ (Undirected Graph from G_{MI})

Fig. 2. The Graphs G_{MI} and $\hat{G}_{MI}$ of Table 7

We provide the user with the flexibility to specify the threshold μ to be a value in the range of $[0, 1]$, according to the user's requirement of the strongness of the relationship between the attributes. One way to set the value of μ, without any domain knowledge, is based on the *density* of the MI graph. The graph density is defined as the number of edges in the graph divided by the number of edges in the corresponding complete graph. We first specify a graph density d for the MI graph. Then, we set μ to be the normalized mutual information value that attains a density of d for the MI graph. For example, consider the employee database in Table 7. Since there are five attributes, the corresponding complete graph has

$(5 \times 5 - 5 = 20)$ edges. (Self-loops are not considered in the MI graph because the antecedent and the consequent of a QAR are disjoint.) We first compute all the values of normalized mutual information between each distinct pair of attributes as listed in Table 8. (Table 8 also lists the normalized mutual information in the form of $\widetilde{I}(x; x)$ for clear illustration.) We then sort these values in descending order. If we specify the density d of the MI graph to be 20%, the derived MI graph has $(20 \times 20\% = 4)$ edges. Therefore, μ is set to be the fourth largest value of the normalized mutual information in the sorted list.

4.4 Phase III: Clique Computation and QAR Generation

In this final phase of MIC, we find all the cliques in G_{MI} and simultaneously compute the set of frequent itemsets based on the cliques. We then generate the QARs from the frequent itemsets.

4.4.1 Clique Computation and Frequent Itemset Generation

Since there is no direction between the attributes in an itemset, we ignore the direction of the edges in G_{MI} and consider its corresponding *undirected graph* $\hat{G}_{MI}$. Figure 2(b) shows the $\hat{G}_{MI}$ that corresponds to the G_{MI} in Figure 2(a).

As we have discussed in Section 3, the attributes in a QAR (as well as in a frequent itemset) form a clique in the interaction graph G_I. Thus, a clique in G_I represents the set of attributes in a potential frequent itemset. Since $\hat{G}_{MI}$ is constructed to recover the edges in G_I that represent strong informative relationships, we can obtain most of the attribute sets that potentially form frequent itemsets by finding all the cliques in $\hat{G}_{MI}$. Essentially, we utilize $\hat{G}_{MI}$ to do the pruning at the attribute level. Only the attribute sets, which form a clique in $\hat{G}_{MI}$, are considered to generate frequent itemsets. Meanwhile, we also check the support condition of the itemsets to make sure that they are frequent.

We compute all the cliques in $\hat{G}_{MI}$ and generate frequent itemsets using a prefix tree structure. Given $\hat{G}_{MI}$, we construct a prefix tree level by level as follows.

First, a root node is created at Level 0. Then, we create a node for each attribute as a child of the root at Level 1. Each node at Level 1 is labeled with the corresponding attribute name and is attached with a set of intervals whose support is no less than σ. Consecutive base intervals are combined and also attached to the node as long as the support of the combined intervals are no less than σ. However, the larger the range of a combined interval, the less specific is the meaning of the interval. For example, the interval `[1,100]` for the attribute `age` is trivial. To avoid the occurrence of too general combined intervals, a *maximum support threshold* σ_m [25] is specified as an upper bound of the support of a combined interval. In this way, the intervals are combined as long as their support is no greater than σ_m.

Algorithm 1 describes *CliqueMine*(u), which recursively computes all the cliques containing the node u. The algorithm starts from each child of the root of the prefix tree. In the algorithm, *RightSibling*(u) denotes the set of right siblings of u and $Child(u)$ denotes the set of children of u. For each of u's right sibling, v, we check whether there is an edge (u, v) in $\hat{G}_{MI}$ (Line 3). If the edge exists, we create a new node w that has the same label as v and insert w into the tree as a child of u (Line 4). Note that each node is attached with a set of frequent itemsets that have the same attribute set but different value intervals. Thus, the attribute set and the value intervals of u and v are joined to give the attribute set and the value intervals of w (Line 5). If the support of an interval obtained from the join is no less than σ, we attach it to w (Lines 6-8). After we have created all the children for u, we output the set of frequent itemsets attached with u to release the memory (Line 9). Then, we call *CliqueMine* recursively for each child of u, until the tree cannot be further expanded (Lines 10-14).

Algorithm 1 *CliqueMine*(u)

```
1. if ($|RightSibling(u)| > 0$)
2.   for each node $v \in RightSibling(u)$ do
3.     if ($(u, v) \in \hat{G}_{MI}$)
4.       Add a new node $w$, with the same label as $v$, as $u$'s child;
5.       Join the sets of frequent itemsets associated with $u$ and $v$;
6.       for each itemset, $X$, obtained from the join do
7.         if ($supp(X) \geq \sigma$)
8.           Attach $X$ to the node $w$;
9.   Output the set of frequent itemsets associated with $u$;
10.  if ($|Child(u)| > 0$)
11.    for each node $w \in Child(u)$ do
12.      $CliqueMine(w)$;
13. else
14.  Output the set of frequent itemsets associated with $u$;
```

In the prefix tree constructed by Algorithm 1, each path from a child of the root at Level 1 to a node at Level k represents a *k-clique* in $\hat{G}_{MI}$, where a k-clique is a clique that consists of k nodes. We prove this observation in the following lemma.

Lemma 1 Let u_1 be a node at Level 1 in the prefix tree and u_k a node at Level k. A path from u_1 to u_k, $\langle u_1, \ldots, u_k \rangle$, represents a k-clique in $\hat{G}_{MI}$, where $\{u_1, \ldots, u_k\}$ is the set of nodes in the k-clique.

Proof. We prove the lemma by induction on k.

(Basis.) When $k = 1$ and $k = 2$, it is trivial that u_1 is a 1-clique and the edge (u_1, u_2) forms a 2-clique.
(Induction.) Assume the lemma holds for $2 \leq j \leq k$. Consider a path $P_{k+1} = \langle u_1, \ldots, u_{k-1}, u_k, u_{k+1} \rangle$. By Algorithm 1, if P_{k+1} exists, u_k must have a right sibling u_{k+1} and the edge (u_k, u_{k+1}) exists in $\hat{G}_{MI}$. By the inductive hypothesis, $P_k = \langle u_1, \ldots, u_{k-1}, u_k \rangle$ and $P'_k = \langle u_1, \ldots, u_{k-1}, u_{k+1} \rangle$ represent two k-cliques, which implies that $\forall u \in \{u_1, \ldots, u_{k-1}\}$ and $\forall v \in \{u_1, \ldots, u_{k-1}, u_k, u_{k+1}\}$, (u, v) exists in $\hat{G}_{MI}$. Thus, adding the edge (u_k, u_{k+1}) forms a $(k + 1)$-clique that consists of the same set of nodes, $\{u_1, \ldots, u_{k-1}, u_k, u_{k+1}\}$, as on the path P_{k+1}. □

It then follows directly from Lemma 1 that the attribute set of each node at Level k is represented by the k-path from the node at Level 1 to that node at Level k in the prefix tree. Note that the reverse statement of Lemma 1 is not true, that is each k-clique in $\hat{G}_{MI}$ may not represent a path $\langle u_1, \ldots, u_k \rangle$ in the prefix tree. This is because due to the checking of the support condition, some path in the prefix tree may not be constructed if there is no frequent itemset produced for the corresponding attribute set, even though there is a corresponding clique in $\hat{G}_{MI}$.

We use the following example to illustrate how the computation of frequent itemsets can be guided by enumerating the cliques in $\hat{G}_{MI}$.

Example 7 Let $\sigma = 0.3$ and $\sigma_m = 0.6$. Figure 3 shows the prefix tree that we construct from the $\hat{G}_{MI}$ shown in Figure 2(b). Each solid rectangle represents a node labeled with an attribute name, while each node in the prefix tree except the root node is associated with a set of intervals, which are the intervals of frequent itemsets. The intervals are shown in a dashed rectangle attached to the node. In Figure 3, we only show the intervals of three nodes for illustration and omit those of others for simplicity. It can be easily verified that all the paths in the prefix tree represent the cliques in $\hat{G}_{MI}$.

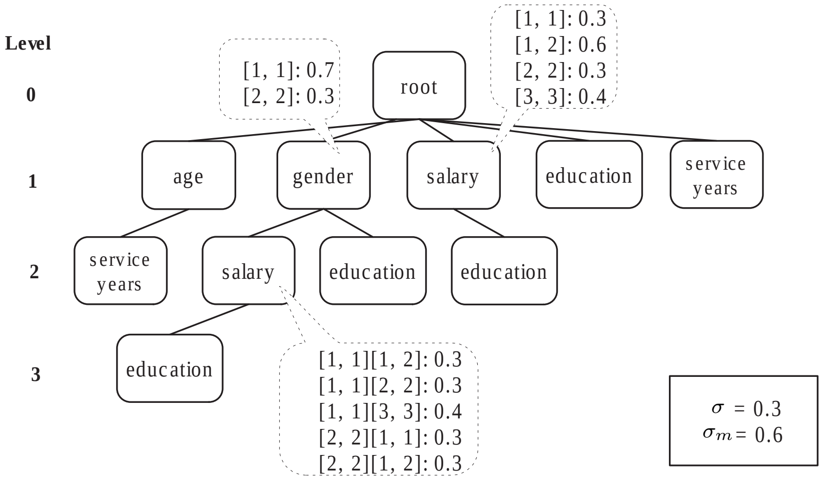


Fig. 3. Prefix Tree for $\hat{G}_{MI}$ in Figure 2(b)

We demonstrate the execution of Algorithm 1 on the subtree rooted at the node `gender`, which is the second child of the root. We begin with the first right sibling of `gender`, that is, the node `salary`. Since the edge (`gender`, `salary`) exists in $\hat{G}_{MI}$, we create a new node labeled `salary` and add it as the first child of `gender`.

Then, we join the set of intervals attached with `gender` and that attached with `salary`. The set of intervals attached with `gender` is {`([1,1]: 0.7)`, `([2,2]: 0.3)`} and that attached with `salary` is {`([1,1]: 0.3)`, `([1,2]: 0.6)`, `([2,2]: 0.3)`, `([3,3]: 0.4)`}, where the number following colon symbol "`:`" is the support of the corresponding itemset. Note that the intervals `[1,1]` and `[2,2]` of `salary` are combined to produce the interval `[1,2]` because $supp(\texttt{salary}[1,2]) = 0.3 + 0.3 = 0.6 \leq \sigma_m$. The join of `gender` and `salary` produces five frequent 2-itemsets. Since these five 2-itemsets have the same attribute set, {`gender`, `salary`}, we attach their intervals, `([1,1][1,2]:3)`, `([1,1][2,2]:3)`, `([1,1][3,3]:4)`, `([2,2][1,1]:3)` and `([2,2][1,2]:3)` with the child node `salary` of `gender`. Similarly, we create the node `education` as the second child of `gender`, with the set of intervals, {`([1,1][3,3]:3)`, `([2,2][1,1]:3)`}, that are obtained by joining the intervals of `gender` and `education`.

We proceed to the next level and process the children of `gender`. Since there is an edge between `salary` and its right sibling `education` in $\hat{G}_{MI}$, we create a new node labeled `education` as a child of `salary`. Note that the path ⟨`gender`, `salary`, `education`⟩ represents the 3-clique {`gender`, `salary`, `education`} in $\hat{G}_{MI}$. We then perform the join on the intervals of `salary` and `education` at Level 2 and generate two frequent 3-itemsets.

In a similar way, we follow the clique enumeration process to generate all other frequent itemsets. □

By enumerating the cliques in $\hat{G}_{MI}$ with a prefix tree structure, we limit the search space of the frequent itemset computation to the prefix tree representation of all cliques in $\hat{G}_{MI}$. Without using the normalized mutual information concept, the search space is equivalent to a prefix tree representation of a *complete graph* with all attributes as vertices. Thus, the search space is drastically reduced.

It is known that the complexity of enumerating all cliques in a graph is NP-complete [7]. However, we emphasize that utilizing the cliques in $\hat{G}_{MI}$ does not mean to solve the NP-complete problem. Instead, we seamlessly incorporate the clique enumeration into the computation of frequent itemsets, such that the only extra processing incurred on the computation of frequent itemsets is a test of whether an edge between a node and its right sibling exists in $\hat{G}_{MI}$ (as shown in Line 3 of Algorithm 1), which is a trivial operation.

We adopt *diffset* [30] on the prefix tree, so that we only scan the database twice: one for computing the frequent items, and another one for computing the initial diffsets (i.e., sets of transaction IDs). All other frequent itemsets are then computed using the diffsets. We also remark that the first scan of the database also computes the normalized mutual information between the attributes.

4.4.2 QAR Generation

After the set of frequent itemsets is derived, we simply map each frequent itemset into a boolean itemset. Then, the algorithm for BAR generation in [3] can be trivially applied to generate the QARs.

4.5 Theoretical Bounds for QARs

In this section, we first study the theoretical bounds on the confidence of QARs for a given frequent itemset and the minimum information threshold. Then, we introduce the measure of *interest* as to further assess the quality of QARs. We further provide the theoretical bounds on the interest of QARs.

4.5.1 Theoretical Bounds for the Confidence of QARs

We formalize the connections between the normalized mutual information, and the support and confidence of QARs. The significance of our result is twofold. First, we guarantee that any pair of attributes pruned by normalized mutual information cannot form a QAR with a confidence greater than the derived bound. Second, we ensure that the attributes retained in the MI graph generate QARs with confidence greater than the given bound.

Given two attributes x and y, we let n_x and n_y denote the number of distinct values of x and y, respectively.

Theorem 1 Let $x[v_x, v_x]y[v_y, v_y]$ be a frequent itemset. Then the confidence $\mathit{conf}(y[v_y, v_y] \Rightarrow x[v_x, v_x])$ has
(a) an upper bound if $\widetilde{I}(x; y) < \mu$, and
(b) a lower bound if $\widetilde{I}(x; y) \geq \mu$.

Proof. Without loss of generality, we assume that the itemset $x[v_{x_1}, v_{x_1}]y[v_{y_1}, v_{y_1}]$ is frequent.

In order to establish the result in Part(a), we show that if $\widetilde{I}(x; y) < \mu$, $\mathit{conf}(y[v_{y_1}, v_{y_1}] \Rightarrow x[v_{x_1}, v_{x_1}])$ (i.e., $p(v_{x_1}|v_{y_1})$) has an upper bound.

Since $\widetilde{I}(x; y) < \mu$, by Property 8, we have $\widetilde{I}(x; y) = (1 - \frac{H(x|y)}{H(x)}) < \mu$, and hence $\frac{H(x|y)}{H(x)} > (1 - \mu)$. We start by deriving an upper bound for $\frac{H(x|y)}{H(x)}$.

$$
\begin{aligned}
&\frac{H(x|y)}{H(x)} \\
= &\frac{-\sum_{i=1}^{n_x}\sum_{j=1}^{n_y} p(v_{x_i}, v_{y_j}) \cdot \log p(v_{x_i}|v_{y_j})}{-\sum_{i=1}^{n_x} p(v_{x_i}) \cdot \log p(v_{x_i})} \\
= &\frac{p(v_{x_1}, v_{y_1}) \cdot \log p(v_{x_1}|v_{y_1}) + \sum_{i\neq1\&j\neq1} p(v_{x_i}, v_{y_j}) \cdot \log \frac{p(v_{x_i}, v_{y_j})}{p(v_{y_j})}}{p(v_{x_1}) \cdot \log p(v_{x_1}) + \sum_{i\neq1} p(v_{x_i}) \cdot \log p(v_{x_i})} \\
\leq &\frac{p(v_{x_1}, v_{y_1}) \cdot \log p(v_{x_1}|v_{y_1}) + (1 - p(v_{x_1}, v_{y_1})) \cdot \log \frac{1-p(v_{x_1}, v_{y_1})}{n_x - p(v_{y_1})}}{p(v_{x_1}) \cdot \log p(v_{x_1}) + \sum_{i\neq1} p(v_{x_i}) \cdot \log p(v_{x_i})} \qquad (5) \\
\leq &\frac{p(v_{x_1}, v_{y_1}) \cdot \log p(v_{x_1}|v_{y_1}) + (1 - p(v_{x_1}, v_{y_1})) \cdot \log \frac{1-p(v_{x_1}, v_{y_1})}{n_x - p(v_{y_1})}}{p(v_{x_1}) \cdot \log p(v_{x_1}) + (1 - p(v_{x_1})) \cdot \log(1 - p(v_{x_1}))}. \qquad (6)
\end{aligned}
$$

Equation (5) is the application of the log sum inequality for the second term in the numerator, leading to the inequality of $\sum_{i\neq1\&j\neq1}(p(v_{x_i}, v_{y_j}) \cdot \log \frac{p(v_{x_i}, v_{y_j})}{p(v_{y_j})}) \geq (\sum_{i\neq1\&j\neq1} p(v_{x_i}, v_{y_j}))\cdot\log \frac{\sum_{i\neq1\&j\neq1} p(v_{x_i}, v_{y_j})}{\sum_{i\neq1} p(v_{y_1}) + \sum_{i=1}^{n_x}\sum_{j=2}^{n_y} p(v_{y_j})}$. Equation (6) holds because in the denominator, we have $p(v_{x_i}) \leq (1 - p(v_{x_1}))$ whenever $i \neq 1$.

Since $x[v_{x_1}, v_{x_1}]y[v_{y_1}, v_{y_1}]$ is frequent, we have $\sigma \leq p(v_{x_1}, v_{y_1}) \leq \sigma_m$ and $\sigma \leq p(v_{x_1}) \leq \sigma_m$. Thus, it follows that:

$$\frac{H(x|y)}{H(x)} \leq \frac{\sigma_m \cdot \log p(v_{x_1}|v_{y_1}) + (1-\sigma) \cdot \log \frac{1-\sigma_m}{n_x-\sigma}}{\sigma \cdot \log \sigma_m + (1-\sigma_m) \cdot \log(1-\sigma)}.$$

Finally, since we have $\frac{H(x|y)}{H(x)} > (1-\mu)$, it follows that:

$$(1-\mu) < \frac{\sigma_m \cdot \log p(v_{x_1}|v_{y_1}) + (1-\sigma) \cdot \log \frac{1-\sigma_m}{n_x-\sigma}}{\sigma \cdot \log \sigma_m + (1-\sigma_m) \cdot \log(1-\sigma)}.$$

So, we have the following upper bound for *conf*$(y[v_{y_1}, v_{y_1}] \Rightarrow x[v_{x_1}, v_{x_1}])$:

$$p(v_{x_1}|v_{y_1}) < ((\sigma_m^{\sigma} \cdot (1-\sigma)^{1-\sigma_m})^{1-\mu} \cdot (\frac{n_x - \sigma}{1-\sigma_m})^{1-\sigma})^{\frac{1}{\sigma_m}}.$$

If we allow a looser upper bound, the above expression can be further simplified as follows:

$$p(v_{x_1}|v_{y_1}) < \sigma_m^{\sigma(1-\mu)} \cdot (\frac{n_x}{1-\sigma_m}).$$

In order to prove Part(b) we show that if $\widetilde{I}(x;y) \geq \mu$, *conf*$(y[v_{y_1}, v_{y_1}] \Rightarrow x[v_{x_1}, v_{x_1}])$ (i.e., $p(v_{x_1}|v_{y_1})$) has a lower bound.

Similar to the proof in Part(a), we first derive a lower bound for $\frac{H(x|y)}{H(x)}$.

$$\begin{aligned} & \frac{H(x|y)}{H(x)} \\ = & \frac{p(v_{x_1}, v_{y_1}) \cdot \log p(v_{x_1}|v_{y_1}) + \sum_{i \neq 1 \& j \neq 1} p(v_{x_i}, v_{y_j}) \cdot \log \frac{p(v_{x_i}, v_{y_j})}{p(v_{y_j})}}{p(v_{x_1}) \cdot \log p(v_{x_1}) + \sum_{i \neq 1} p(v_{x_i}) \cdot \log p(v_{x_i})} \\ \geq & \frac{p(v_{x_1}, v_{y_1}) \cdot \log p(v_{x_1}|v_{y_1})}{p(v_{x_1}) \cdot \log p(v_{x_1}) + (1-p(v_{x_1})) \cdot \log \frac{1-p(v_{x_1})}{n_x - 1}} \\ \geq & \frac{\sigma \cdot \log p(v_{x_1}|v_{y_1})}{\sigma_m \cdot \log \sigma + (1-\sigma) \cdot \log \frac{1-\sigma_m}{n_x-1}}. \end{aligned} \qquad (7)$$

Equation (7) holds, since we apply the log sum inequality for the second term in the denominator, which is similar to Part(a). The second term in the numerator is negative because the conditional probability falls within the range [0,1].

Finally, since $\widetilde{I}(x;y) = (1 - \frac{H(x|y)}{H(x)}) \geq \mu$, so $\frac{H(x|y)}{H(x)} \leq (1-\mu)$, that is,

$$(1-\mu) \geq \frac{\sigma \cdot \log p(v_{x_1}|v_{y_1})}{\sigma_m \cdot \log \sigma + (1-\sigma) \cdot \log \frac{1-\sigma_m}{n_x - 1}}.$$

Therefore, we have the following lower bound for *conf*$(y[v_{y_1}, v_{y_1}] \Rightarrow x[v_{x_1}, v_{x_1}])$:

$$p(v_{x_1}|v_{y_1}) \geq (\sigma^{\sigma_m} \cdot (\frac{1-\sigma_m}{n_x - 1})^{1-\sigma})^{\frac{1-\mu}{\sigma}}.$$

If we allow a looser lower bound, the above expression can be further simplified as follows:

$$p(v_{x_1}|v_{y_1}) \geq (\sigma \cdot (\frac{1-\sigma_m}{n_x}))^{\frac{1-\mu}{\sigma}}.$$

□

The following corollary shows that Theorem 1 can be generalized to the itemsets with intervals instead of single values.

Corollary 1 Let $x[l_x, u_x]y[l_y, u_y]$ be a frequent itemset. Then the confidence *conf*$(y[l_y, u_y] \Rightarrow x[l_x, u_x])$ has an upper bound if $\widetilde{I}(x;y) < \mu$, and has a lower bound if $\widetilde{I}(x;y) \geq \mu$.

Proof. It directly follows from Theorem 1, since the derived equations are based on probabilities. Once $\widetilde{I}(x;y)$ refers to the one with respect to the intervals of frequent itemsets, we can simply sum up the probabilities of the composite values of a given interval to obtain the same bounds. □

The next corollary shows that Theorem 1 can also be generalized to the QARs.

Corollary 2 If $\mathit{conf}(y[l_y, u_y] \Rightarrow x[l_x, u_x]) < c$, then for any rule $(y[l_y, u_y] \Rightarrow x[l_x, u_x]Z)$, where Z is an itemset, we have $\mathit{conf}(y[l_y, u_y] \Rightarrow x[l_x, u_x]Z) < c$.

Proof. By the definition of the confidence of a rule, we have the following expression:

$$\begin{aligned}
\mathit{conf}(y[l_y, u_y] \Rightarrow x[l_x, u_x]Z) &= \frac{\mathit{supp}(x[l_x, u_x]y[l_y, u_y]Z)}{\mathit{supp}(y[l_y, u_y])} \\
&\leq \frac{\mathit{supp}(x[l_x, u_x]y[l_y, u_y])}{\mathit{supp}(y[l_y, u_y])} \\
&= \mathit{conf}(y[l_y, u_y] \Rightarrow x[l_x, u_x]) \\
&< c.
\end{aligned}$$

□

Corollary 2 is important, since it shows that if the confidence of a rule has an upper bound, the confidence of all the rules formed by augmenting more items in the consequent of the rule also have the the same upper bound. Therefore, the upper bound derived in the proof of Theorem 1 is not limited to the rule having one single item in both antecedent and consequent, but also generally holds for the rules that have more items in the consequent.

4.5.2 Theoretical Bounds for the Interest of QARs

To formally assess the quality of the mined QARs, we employ another well-established measure for association rules, called *interest* [5]. The interest of a rule, $X \Rightarrow Y$, is the statistical definition of dependence of X and Y, given as follows:

$$\mathit{interest}(X \Rightarrow Y) = \frac{\mathit{supp}(X \cup Y)}{\mathit{supp}(X)\mathit{supp}(Y)}.$$

The range of the interest of an association rule is from 0 to ∞. Interest values above 1 indicate positive dependence, while values below 1 indicate negative dependence. An interest value of 1 implies that X and Y are independent, while the further the value is from 1, the greater is the positive or negative dependence between X and Y.

Similar to the results of Section 4.5.1, we formalize the connections between the normalized mutual information, and the support and interest of QARs.

Theorem 2 Let $x[v_x, v_x]y[v_y, v_y]$ be a frequent itemset. Then, the interest, $\mathit{interest}(y[v_y, v_y] \Rightarrow x[v_x, v_x])$, has
(a) an upper bound if $\widetilde{I}(x; y) < \mu$, and
(b) a lower bound if $\widetilde{I}(x; y) \geq \mu$.

Proof. To establish the result of Part (a), we refer to the proof of Theorem 1. By Equation (6), we have:

$$\frac{H(x|y)}{H(x)}$$
$$\leq \frac{p(v_{x_1}, v_{y_1}) \cdot (\log \frac{p(v_{x_1}|v_{y_1})}{p(v_{x_1})} + \log p(v_{x_1})) + (1 - p(v_{x_1}, v_{y_1})) \cdot \log \frac{1-p(v_{x_1}, v_{y_1})}{n_x - p(v_{y_1})}}{p(v_{x_1}) \cdot \log p(v_{x_1}) + (1 - p(v_{x_1})) \cdot \log(1 - p(v_{x_1}))}.$$

Therefore, it follows that:

$$(1 - \mu) < \frac{\sigma_m \cdot \log \frac{p(v_{x_1}, v_{y_1})}{p(v_{x_1})p(v_{y_1})} + \sigma_m \cdot \log \sigma + (1 - \sigma) \cdot \log \frac{1-\sigma_m}{n_x - \sigma}}{\sigma \cdot \log \sigma_m + (1 - \sigma_m) \cdot \log(1 - \sigma)}.$$

So, we have the following upper bound for $\mathit{interest}(y[v_{y_1}, v_{y_1}] \Rightarrow x[v_{x_1}, v_{x_1}])$:

$$\frac{p(v_{x_1}, v_{y_1})}{p(v_{x_1})p(v_{y_1})} < \frac{1}{\sigma} \cdot ((\sigma_m^{\sigma} \cdot (1 - \sigma)^{1-\sigma_m})^{1-\mu} \cdot (\frac{n_x - \sigma}{1 - \sigma_m})^{1-\sigma})^{\frac{1}{\sigma_m}}.$$

If we allow a looser upper bound, the above expression can be further simplified as follows:

$$\frac{p(v_{x_1}, v_{y_1})}{p(v_{x_1})p(v_{y_1})} < \sigma_m^{\sigma(1-\mu)} \cdot (\frac{n_x}{\sigma(1 - \sigma_m)}). \quad (8)$$

Similarly, in order to prove Part (b), by Equation (7), we have

$$\begin{aligned} &\frac{H(x|y)}{H(x)} \\ \geq\ & \frac{p(v_{x_1}, v_{y_1}) \cdot (\log \frac{p(v_{x_1}|v_{y_1})}{p(v_{x_1})} + \log p(v_{x_1}))}{p(v_{x_1}) \cdot \log p(v_{x_1}) + (1 - p(v_{x_1})) \cdot \log \frac{1-p(v_{x_1})}{n_x - 1}}. \end{aligned}$$

Therefore, it follows that,

$$(1-\mu) \geq \frac{\sigma \cdot \log \frac{p(v_{x_1}, v_{y_1})}{p(v_{x_1})p(v_{y_1})} + \sigma \cdot \log \sigma_m}{\sigma_m \cdot \log \sigma + (1-\sigma) \cdot \log \frac{1-\sigma_m}{n_x - 1}}.$$

So we have the following lower bound for *interest*$(y[v_{y_1}, v_{y_1}] \Rightarrow x[v_{x_1}, v_{x_1}])$:

$$\frac{p(v_{x_1}, v_{y_1})}{p(v_{x_1})p(v_{y_1})} \geq \frac{1}{\sigma_m} \cdot (\sigma^{\sigma_m} \cdot (\frac{1-\sigma_m}{n_x - 1})^{1-\sigma})^{\frac{1-\mu}{\sigma}}.$$

If we allow a looser lower bound, the above expression can be further simplified as follows:

$$\frac{p(v_{x_1}, v_{y_1})}{p(v_{x_1})p(v_{y_1})} \geq \frac{1}{\sigma_m} \cdot (\sigma \cdot (\frac{1-\sigma_m}{n_x}))^{\frac{1-\mu}{\sigma}}.$$

□

The results in Theorem 2 can be further generalized to itemsets with intervals, as shown in the following corollary. We skip the proof since it is similar to that of Corollary 1.

Corollary 3 Let $x[l_x, u_x]y[l_y, u_y]$ be a frequent itemset. Then, the interest, *interest*$(y[l_y, u_y] \Rightarrow x[l_x, u_x])$, has an upper bound if $\widetilde{I}(x; y) < \mu$, and has a lower bound if $\widetilde{I}(x; y) \geq \mu$.

The following corollary describes the connection between the confidence and the interest of a QAR.

Corollary 4 Let $x[l_x, u_x]y[l_y, u_y]$ be a frequent itemset. If *conf*$(y[l_y, u_y] \Rightarrow x[l_x, u_x]) \geq c$, then *interest*$(y[l_y, u_y] \Rightarrow x[l_x, u_x]) \geq \frac{c}{\sigma_m}$.

Proof. By the definition of the interest of a rule, we have the following expressions:

$$\begin{aligned} \mathit{interest}(y[l_y, u_y] \Rightarrow x[l_x, u_x]) &= \frac{\mathit{supp}(x[l_x, u_x]y[l_y, u_y])}{\mathit{supp}(y[l_y, u_y]) \cdot \mathit{supp}(x[l_x, u_x])} \\ &= \frac{\mathit{conf}(y[l_y, u_y] \Rightarrow x[l_x, u_x])}{\mathit{supp}(x[l_x, u_x])} \\ &\geq \frac{c}{\sigma_m}. \end{aligned}$$

□

Corollary 4 shows that, if the confidence of a rule has a lower bound, the interest of a rule also has a lower bound that is related to the bound of confidence. Because of this connection, we can simply specify a confidence threshold for mining QARs, while we still have guarantee on the interest of QARs. However, since the range of interest is different from that of confidence, we still have to study the interest measure in order to assess the quality of QARs, as what we are going to show in Section 5.

4.6 Discussions on the Interestingness of Missing QARs

A QAR is an implication on a local set of transactions that satisfy the antecedent of the rule. The NMI measure, however, computes the dependency relationship between two attributes on the whole set of transactions and takes into account all values in the attribute domain. As a result, the NMI pruning may eliminate some QARs that are interesting locally within a small set of transactions (i.e., the QARs have low support values). This problem can also be seen from Equation (8) in Theorem 2: when σ decreases, the upper bound of the interest of the missing QARs increases.

A possible solution to this problem is to allow the user to specify a *maximum interest threshold* θ ($\theta > 0$) for the missing QARs that he/she can tolerate. Then, according to Equation (8), we can derive a lower bound for the value of μ as follows:

$$\mu \geq 1 - \frac{\log(\theta \cdot \sigma(1-\sigma_m)) - \log n_x}{\sigma \log \sigma_m}.$$

The above bound provides a useful reference for setting μ in terms of σ, σ_m and θ. In this way, we can avoid missing the QARs with interest higher than θ by setting a suitable μ.

5 Experimental Evaluation

We evaluate the performance of our MIC framework on both synthetic and real datasets. We use SAM [25] as the baseline for comparison on the efficiency of the algorithms and quality of the mined QARs. Recall that MIC operates on an MI graph that captures the strong informative relationships between the attributes, while SAM operates on the *complete* graph that assumes all attributes have a strong informative relationship with each other. In order to make a fair comparison, we test SAM by inputting a complete graph into our program, so that the performance improvement is indeed only due to the pruning as a result of using the MI graph. Thus, the SAM used in our experiment is not the Apriori-like algorithm proposed in [25], but a more efficient prefix-tree implementation using *diffset* [30]. Since SAM uses the equidepth discretization with the number of base intervals n calculated by an equation to minimize the information loss, we also apply the same discretization in MIC. The equation is given by $n = \frac{2 \times m}{\sigma \times (K-1)}$, where m is the number of quantitative attributes and K is the partial completeness level. We choose $K = 1.5$ in the experiments as suggested in SAM. After generating all the frequent itemsets, we apply the rule generation algorithm in [3] to obtain the QARs. All the experiments are run on an XP machine with a 3.0 GHz Intel P4 and 2 GB RAM.

5.1 The *Interest* Measure

Since we perform pruning at the attribute level of the QAR mining problem, the set of frequent itemsets produced by MIC is a subset of that produced by SAM. Consequently, the set of QARs generated by MIC is also a subset of that generated by SAM. However, we emphasize that our method is not an approximation technique that improves the efficiency at the expense of accuracy. Instead, we show that MIC not only significantly outperforms SAM, but the rules we obtain are also of higher quality than that obtained by SAM, as measured using *interest* [5], which is a well-established measure for the interestingness of an association rule.

In particular, we show that the *missing QARs*, i.e., QARs that are missed by MIC but returned by SAM, are rules whose attributes are of low dependency on each other. For example, consider two boolean attributes x and y, $supp(x[1,1]y[1,1]) = 0.81$, $supp(x[1,1]y[0,0]) = 0.09$, $supp(x[0,0]y[1,1]) = 0.09$, and $supp(x[0,0]y[0,0]) = 0.01$. Although the rules $x[1,1] \Rightarrow y[1,1]$ and $y[1,1] \Rightarrow x[1,1]$ have a high confidence of 0.9, x and y are independent of each other since $supp(x,y){=}supp(x) \cdot supp(y)$ for all possible values of x and y. Clearly, the two rules are of little significance. They are derived simply because the occurrences of $x[1,1]$ and $y[1,1]$ are prevalent in the database (each of them occurs in 90% of the transactions). Thus, it just happens to be the case that whenever we have $x[1,1]$, we are likely to have $y[1,1]$ as well. Such rules are not generated by MIC, because these attributes have very low normalized mutual information and are hence excluded from the MI graph.

In our experiments, we first use support and confidence to obtain the high-confidence QARs. Then, we compute the mean and variance of the interest of the missing QARs. The maximum interest of missing QARs is also presented. We justify that most of the missing QARs are of low interest. To unify the scale of the positive dependent interest values

$((1, \infty])$ and negative ones $([0, 1))$, we convert the negative dependent interest values into their inverse when computing their mean, variance and maximum.

5.2 Datasets and Parameters

In this section, we introduce the datasets and the parameters we are going to study in the subsequent subsections.

We use both synthetic and real datasets to justify the effectiveness and efficiency of MIC. The synthetic datasets are generated by the IBM Quest Synthetic Data Generator [12]. We modify their code to generate three extra boolean attributes, using Functions 1-3 described in [1]. Thus, each dataset has six quantitative and six categorical attributes. We generate five datasets of sizes from 100K to 1,000K transactions as a scalability test for MIC. The four real datasets we test are chosen from the commonly used UCI machine learning repository [11]. Table 9 lists the name, the number of transactions and the number of attributes of all the datasets. The number of quantitative attributes of each dataset is given in the brackets.

Table 9. Dataset Description

Dataset	Number of Transactions	Number of all Attributes (Quantitative Attributes)
synthetic	100,000 - 1,000,000	12(6)
covtype	581,012	55(10)
letter-recognition	20,000	17(16)
ann-thyroid	7,200	22(6)
yeast	1,484	9(8)

In the following subsection, we first study the effect of μ in the MIC framework. Meanwhile, we also demonstrate the scalability of MIC by varying the size of synthetic datasets. Then, we study the effect of minimum support threshold σ on real datasets.

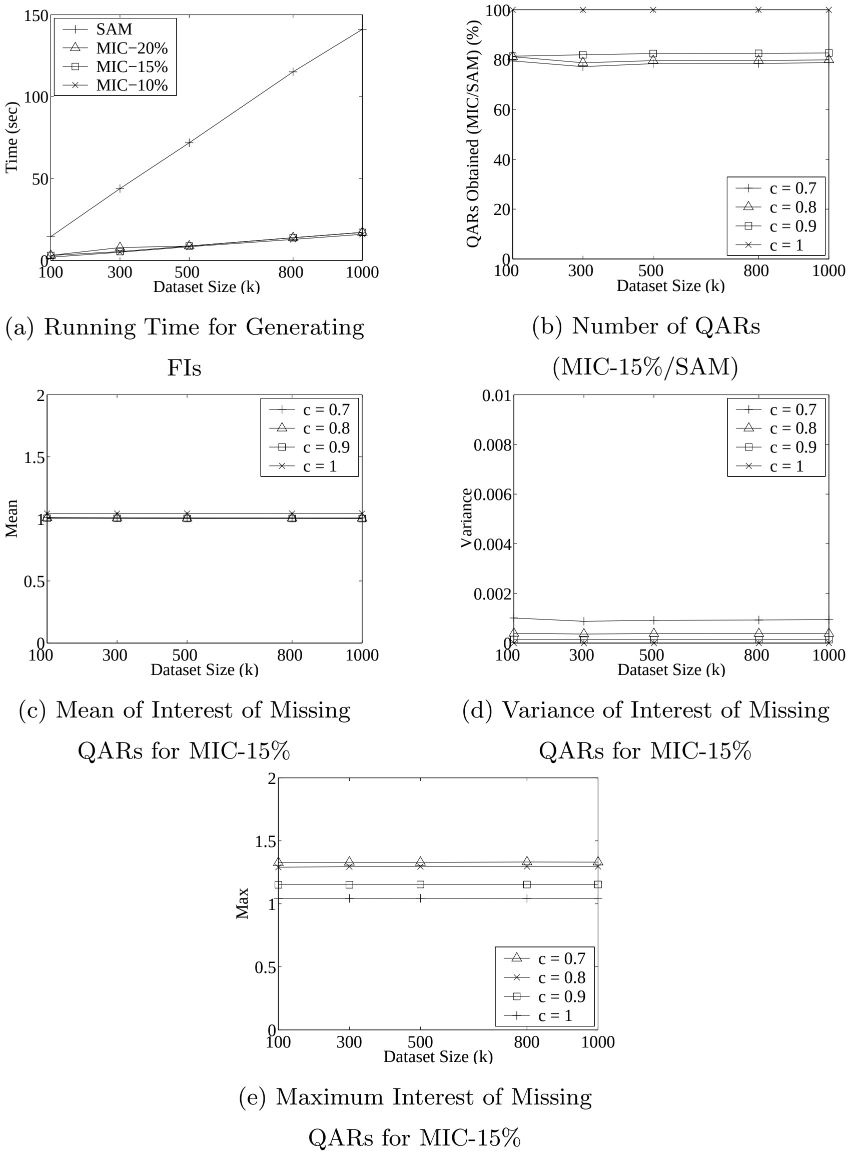
5.3 Experimental Results

5.3.1 Experiments on Synthetic Datasets

We set $\sigma = 0.1$ and $\sigma_m = 0.13$. We test three values of μ by ensuring the density of the MI graph at 20%, 15% and 10%, respectively. For each dataset, we generate four sets of QARs, at the minimum confidence threshold $c = 0.7$, $c = 0.8$, $c = 0.9$ and $c = 1$, respectively.

Figure 4(a) shows the running time for generating the frequent itemsets. For all the three values of μ, MIC runs significantly faster than SAM. While the running time of SAM increases linearly when the size of the dataset increases, the running time of MIC remains relatively stable. When the density of the MI graph decreases from 20% to 10% (i.e. the value of μ increases), the running time of MIC decreases only slightly. The decrease in the running time is because the size of the MI graph is smaller for larger μ and hence the search space pruned is also larger. However, the decrease in running time is small because the difference in the MI graphs of the three respective μ is small. More specifically, the MI graph computed on the dataset with size of 1,000k at density 15% only has three more edges than that at density 10%. These three edges consist of at least one categorical attribute which has only two distinct values. Thus, the number of itemsets that are produced from these three edges is also small.

Figure 4(b) shows the ratio of the number of QARs obtained by MIC at density 15% to that obtained by SAM. On average, MIC obtains 80% of QARs that have a confidence over 0.7, while it obtains almost all QARs that have a confidence of 1. Most importantly, we show in Figures 4(c-d)



(a) Running Time for Generating FIs

(b) Number of QARs (MIC-15%/SAM)

(c) Mean of Interest of Missing QARs for MIC-15%

(d) Variance of Interest of Missing QARs for MIC-15%

(e) Maximum Interest of Missing QARs for MIC-15%

Fig. 4. Performance on Synthetic Datasets

that the mean of the interest of the missing QARs is approximately 1 in all cases, with a very small variance of less than 0.001. Figure 4(e) further shows that the maximum interest of any missing QARs is averagely 1.2, which is very close to 1. This result implies that the attributes composing a missing QAR are independent to each other.

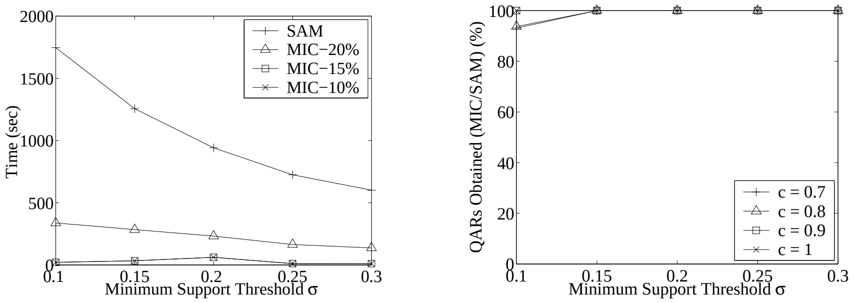
5.3.2 Experiments on Real Datasets

We test real datasets by varying σ from 0.1 to 0.3. When discretizing the datasets, the number of base intervals n is given by the equation $\frac{2\times m}{\sigma\times(K-1)}$, which is inverse proportional to σ. Therefore, when σ increases from 0.1 to 0.3, the number of base intervals decreases accordingly. We set σ_m at 0.03, 0.3, 0.03, and 0.1 higher than the respective σ for *covtype*, *letter-recognition*, *ann-thyroid*, and *yeast*, respectively. The above values of σ_m are the maximum values of σ_m at which SAM does not run out of memory. We test three values of μ by ensuring the density of the MI graph at 20%, 15% and 10%, respectively. The values of μ vary for different datasets and are shown in Table 10. We generate QARs at $c = 0.7$, $c = 0.8$, $c = 0.9$ and $c = 1$, respectively.

Table 10. Values of μ for Real Datasets

Dataset	MIC-20%	MIC-15%	MIC-10%
covtype	0.0845636	0.140902	0.2064044
letter-recognition	0.104507	0.151669	0.182439
ann-thyroid	0.0833698	0.1241558	0.1765774
yeast	0.251732	0.268517	0.279695

Figure 5(a) shows that MIC computes the frequent itemsets approximately two orders of magnitude faster than does SAM on the *covtype* dataset when the graph density is lower than 15%. When the graph density is 20%, MIC becomes slower but still significantly outperforms SAM. The dramatic improvement is because many of the quantitative attributes of this dataset have a large domain. MIC is able to remove the edges be-



(a) Running Time for Generating FIs

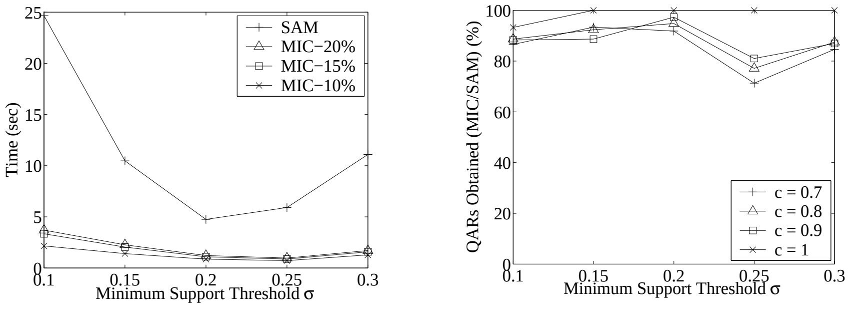
(b) Number of QARs (MIC-15%/SAM)

Fig. 5. Performance on *covtype* Dataset

tween those attributes that do not have a strong informative relationship, thereby drastically reducing the number of intervals to be combined. We also show in Figure 5(b) that only in the case when $\sigma = 0.1$, MIC misses a very small number of QARs of confidence above 0.7 and 0.8, respectively. In all other cases, MIC obtains exactly the same set of QARs as does SAM. We thus omit the figures for the interest measures due to the negligible number of missing QARs.

We notice an unexpected, slight increase in the running time of MIC when σ becomes larger in Figure 5(a), when the graph density is 15% and 10%. This is because in QAR mining, the number of frequent itemsets is also determined by σ_m, since a greater σ_m implies that more intervals can be combined to generate more frequent itemsets. Thus, the number of itemsets generated at $\sigma = 0.1$ and $\sigma_m = 0.13$ can be smaller than that generated at $\sigma = 0.15$ and $\sigma_m = 0.18$, as is with this dataset. Therefore, the time spent on processing frequent itemsets at $\sigma = 0.1$ and $\sigma_m = 0.13$ can be less than that at $\sigma = 0.15$ and $\sigma_m = 0.18$. Without using the MI graph, most of the time is spent on joining the unpromising intervals and the smaller the σ, the more the time used. However, the MI graph of *covtype* at density lower than 15% almost prunes all irrelevant search space and thus the time spent on joining the unpromising intervals becomes insignificant. As for the graph density of 20%, the MI graph still consists of some edges that involve attributes with weak relationship. Thus, the running time of MIC at the density of 20% still follows the trend of that of SAM. This result again verifies that the MI graph can indeed capture the strong informative relationships between the attributes.

Figure 5(a) also reports the effect of the number of base intervals on the performance of MIC and SAM. When the number of base intervals increases, i.e., σ decreases, the running time of SAM increases rapidly, while the running time of MIC remains relatively stable at all values of μ. This is because larger number of base intervals aggravates the problem of combinatorial explosions of attribute intervals. Without pruning the irrelevant search space, the performance of SAM is severely degraded by the increase in the number of base intervals. On the contrary, the performance of MIC is almost not affected by the increase in the number of base intervals since the generation of unpromising itemsets is avoided by the effective pruning.



(a) Running Time for Generating FIs

(b) Number of QARs (MIC-15%/SAM)

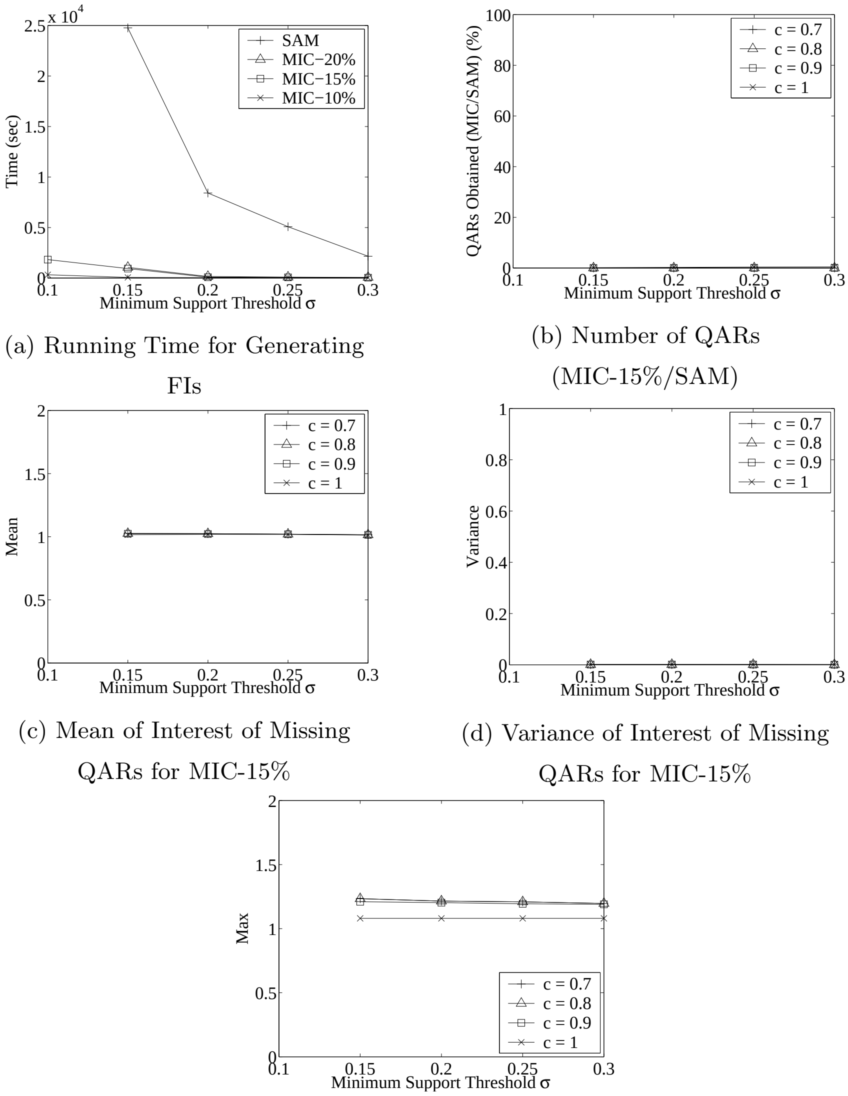
Fig. 6. Performance on *letter-recognition* Dataset

Figure 6(a) shows that MIC is on average six times faster than SAM on the *letter-recognition* dataset when computing the frequent itemsets. All the quantitative attributes in this dataset have a small domain (16 values). Therefore, no matter what value of σ is used for the discritization, the number of base intervals is the same as the size of the domain for all values of σ. Hence, with the increase in σ, the running time of both algorithms is not affected by the number of base intervals but majorally by σ_m. This explains the abnormal trend of SAM in Figure 6(a). The value of σ_m has little influence on MIC due to the effective pruning of attributes with low mutual dependency. Figure 6(b) shows that the set of QARs obtained by MIC is over 90% of that obtained by SAM, except when $\sigma = 0.25$, the percentage is slightly lower. This is because the number of QARs at $\sigma = 0.25$ is the smallest among all values of σ. We omit the figures for the interest measures since the number of missing rules is small.

Figure 7(a) shows the running time of MIC and SAM on the *ann-thyroid* dataset. MIC computes the frequent itemsets up to three orders of magnitude faster than SAM. We are not able to obtain the results of SAM and MIC-20% at $\sigma = 0.1$ since they run out of memory due to the large number of base intervals.

Figure 7(b) shows that the set of QARs obtained by MIC is less than 1% of that obtained by SAM. This result is because SAM generates a prohibitively large number of QARs (up to 1 billion and consumes over 50GB of space). By capturing the strong informative relationships of attributes, MIC produces a reasonable number of interesting QARs (about 60K). Moreover, Figures 7(c-e) show that the missing QARs are indeed uninteresting, since the mean and the maximum interest are almost 1 and the variance of the interest is 0 in all cases.

Figure 8(a) shows the running time of MIC and SAM on the *yeast* dataset. On average, MIC computes the frequent itemsets four times



(a) Running Time for Generating FIs

(b) Number of QARs (MIC-15%/SAM)

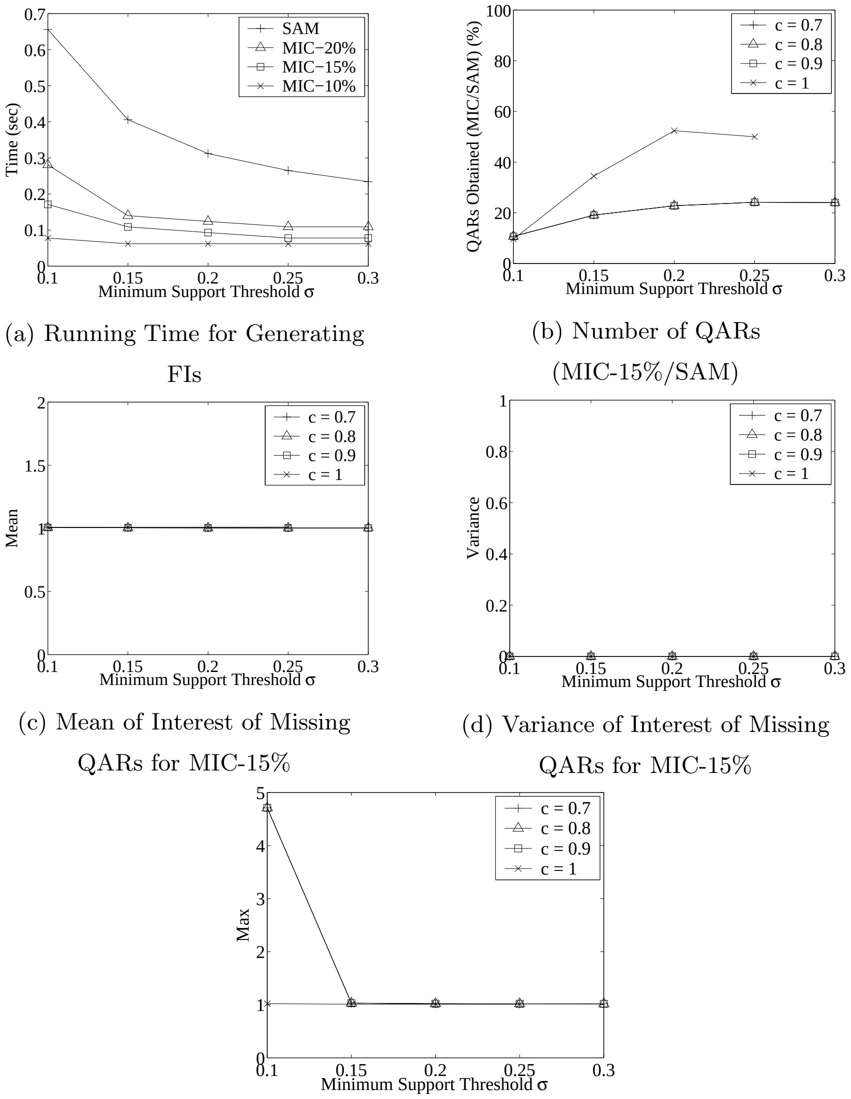
(c) Mean of Interest of Missing QARs for MIC-15%

(d) Variance of Interest of Missing QARs for MIC-15%

(d) Maximum Interest of Missing QARs for MIC-15%

Fig. 7. Performance on *ann-thyroid* Dataset

faster than SAM. The improvement is not as significant as that on other datasets because the dataset is very small (only 1,484 transactions). The MI graph of a small dataset does not reflect the relationships between



(a) Running Time for Generating FIs

(b) Number of QARs (MIC-15%/SAM)

(c) Mean of Interest of Missing QARs for MIC-15%

(d) Variance of Interest of Missing QARs for MIC-15%

(d) Maximum Interest of Missing QARs for MIC-15%

Fig. 8. Performance on *yeast* Dataset

the attributes as good as does the MI graph of a large dataset, because larger datasets are statistically more stable.

Figure 8(b) shows that the set of QARs obtained by MIC at the density of 15% is only about 20-50% of that obtained by SAM. However, Figures 8(c-d) show that the mean and variance of the interest of the missing QARs are 1 and 0 in all cases. Figure 8(e) shows that the maximum interest of the missing QARs is 1 in all cases, except when $\sigma = 0.1$, MIC misses four QARs with interest around 4.6. Thus, the results once again show that the QARs missed by MIC are of low interest.

5.4 Summary of Experiments

Since MIC outperforms SAM in all the experiments, we conclude that utilizing normalized mutual information indeed enables us to effectively reduce the number of attributes to be joined and hence the number of intervals to be joined between attributes. Although the results reveal that the improvement of MIC for small datasets is not as significant as that for large datasets, for most QAR mining problems in practice, the datasets are large and their attributes have a large domain. MIC achieves remarkable performance on such datasets, as verified by the experiments on the large synthetic datasets and on the large real dataset *covtype*.

Another important finding is that the QARs returned by SAM but missed by MIC mostly have an interest value of 1, i.e., the attributes composing the missing QARs are independent on each other. Thus, in addition to the improvement in efficiency, the set of QARs mined by MIC is also of higher quality than that mined by SAM.

6 Related Work

QAR mining is first studied by Piatetshy-Shapiro [20] but the QARs are restricted to a single attribute in both the antecedent and the consequent of a QAR. Srikant and Agrawal [25] generalize the work by allowing multiple attributes in both the antecedent and the consequent. We are also aware of mining four variants of association rules in quantitative databases.

The first one is *optimized association rule mining* [9, 21, 6, 17, 18, 14, 23], which contains certain uninstantiated attributes and the mining problem is to determine values for the uninstantiated attributes such that one measure (e.g., support, confidence or gain) is maximized and another measure satisfies a predefined threshold. Inspired by the problem of image segmentation in computer vision, Fukuda et al. [9] propose a geometric method to compute the optimized region for association rules. However, the rules they produce are limited to having two quantitative attributes. Later, the work [21, 6] generalizes the study in [9] by allowing disjunctions over an arbitrary number of uninstantiated attributes. Another novel approach is proposed to use genetic algorithms to mine optimized association rules. Mata et al. [17, 18] use a genetic algorithm to optimize the support of an interval for a quantitative attribute. However, their approach does not guarantee to produce high confidence rules. Kaya and Alhajj [14] categorize the rules into partial optimized rules and complete optimized rules. A multi-objective genetic algorithm based method is proposed to achieve both types of optimizations. Recently, Salleb-Aouissi et al. [23] develop a system based on a genetic algorithm to achieve optimizing both the support and confidence of a rule. Mining optimal association rules tackles a different problem from ours. It focuses on finding the optimal values of certain given attributes instead of mining general QARs without any constraint on the attributes.

The second type is based on *statistics* [4, 29, 31], in which the consequent of a rule is a statistical measure (e.g., mean, variance) or an aggregate (e.g., min, max) of a quantitative attribute. This type of rule is mainly used to provide a statistical view of the attributes, rather than giving the interval information of the attributes, which is more detailed and intuitive.

The third type proposes a new representation of QARs based on half-spaces [22]. The antecedent and the consequent of a rule are a weighted sum of the attributes tested against a threshold. As a result, this type of rule is very complex and more suitable to scientific analyses.

The fourth type is privacy-preserving QARs [33, 13] proposed recently. The problem is to mine QARs without revealing the private information of parties who share distributed data. Therefore, the mining algorithm mainly focuses on secure computation instead of the efficiency.

We are also aware of different applications of normalized mutual information in literature. The first one is used for data clustering [26], in which the normalized mutual information of two attributes x and y is defined as $\frac{2 \cdot I(x;y)}{\max_{A \in \{1,\ldots,k\}}(H(A)) + \max_{B \in \{1,\ldots,g\}}(H(B))}$, where A represents possible cluster labels and B represents possible category labels. Normalized mutual information is also found in the area of image processing [27], where it is defined as $\frac{H(x)+H(y)}{H(x,\ y)}$. In our recent work [16], we define normalized mutual information as $\frac{I(x;y)}{MAX\{I(x;x),I(y;y)\}}$ for mining *quantitative correlated patterns*. This definition is symmetric, since a correlated pattern requires all attributes in the pattern be strongly correlated, while normalized mutual information in this paper is directional because a QAR is an implication of the antecedent on the consequent. We emphasize that the work in this paper and the work in [16] are complementary to each other, since correlated patterns are in fact originally proposed as a complement to association rules. While the focus in [16] is to mine patterns with strongly correlated items, the contribution of this paper is to obtain the implication of one quantitative pattern on the high probability of the occurrence of another pattern, which is different from the scope of [16].

7 Conclusions

In this paper, we present an MIC framework that adopts an information-theoretic approach to mine QARs. We propose the concept of normalized mutual information and then apply it to discover the informative relationships between the attributes in a QAR mining problem. Based on normalized mutual information, we construct an MI graph that captures the strong informative relationships between the attributes. By defining an interaction graph that reflects the true relationships between the attributes in QARs, we find that the cliques in the MI graph correspond to the potential frequent itemsets in the mining problem. We incorporate the enumeration of the cliques seamlessly into the computation of frequent itemsets. The clique enumeration limits the mining process to a smaller but more relevant search space, thereby significantly improving the mining efficiency. Our experimental results show that MIC speeds up the mining process for up to orders of magnitudes. More importantly, MIC obtains most of the high-confidence QARs, while the QARs that are not returned by MIC are shown to be of little significance based on the *interest* measure.

As an on-going work, we consider to incorporate the concept of near-clique, which is a clique except for one edge, for computing frequent itemsets into our framework. This may help tolerate the noise in forming a clique in the MI graph. Other measures of inter-dependence [10] in the context of QAR mining also deserve attention as a future work.

Acknowledgement We thank the anonymous reviewers for their valuable comments that substantially improve the paper.

References

1. R. Agrawal, T. Imielinski, and A. Swami. Database mining: A performance perspective. *IEEE TKDE*, 5(6), Dec 1993.

2. R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *Proc. of SIGMOD*, 1993.

3. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proc. of VLDB*, pages 487–499, 1994.

4. Y. Aumann and Y. Lindell. A statistical theory for quantitative association rules. *Journal of Intelligent Information Systems*, 20(3):255–283, 2003.

5. S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: Generalizing association rules to correlations. In *SIGMOD*, pages 265–276, 1997.

6. S. Brin, R. Rastogi, and K. Shim. Mining optimized gain rules for numeric attributes. *DMKD*, 15(2):324–338, 2003.

7. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms.* The MIT Press, 2001.

8. T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, Inc., 1991.

9. T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Data mining with optimized two-dimensional association rules. *ACM TODS*, 26(2):179–213, 2001.

10. J. Furnkranz. Separate-and-conquer rule learning. *Artificial Intelligence Review*, 13(1):3–54, February 1999.

11. S. Hettich, C.L. Blake, and C.J. Merz. UCI repository of machine learning databases.

12. IBM Quest Synthetic Data Generation Code for Classification. *http://www.almaden.ibm.com/software /quest/Resources/index.shtml.*

13. Weiwei Jing, Liusheng Huang, Yonglong Luo, Weijiang Xu, and Yifei Yao. An algorithm for privacy-preserving quantitative association rules mining. In *DASC '06: Proceedings of the 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC'06)*, pages 315–324, 2006.

14. Mehmet Kaya and Reda Alhajj. Novel approach to optimize quantitative association rules by employing multi-objective genetic algorithm. In *IEA/AIE'2005: Proceedings of the 18th international conference on Innovations in Applied Artificial Intelligence*, pages 560–562, 2005.

15. Y. Ke, J. Cheng, and W. Ng. MIC framework: An information-theoretic approach to quantitative association rule mining. In *ICDE*, page 112, 2006.

16. Y. Ke, J. Cheng, and W. Ng. Mining quantitative correlated patterns using an information-theoretic approach. In *KDD*, pages 227–236, 2006.

17. J. Mata, J. L. Alvarez, and J. C. Riquelme. Discovering numeric association rules via evolutionary algorithm. In *PAKDD*, pages 40–51, 2002.

18. J. Mata, J. L. Alvarez, and J. C. Riquelme. An evolutionary algorithm to discover numeric association rules. In *SAC*, pages 590–594, 2002.

19. R. J. Miller and Y. Yang. Association rules over interval data. In *Proc. of SIGMOD*, 1997.

20. G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In *Knowledge Discovery in Databases*, pages 229–248. 1991.

21. R. Rastogi and K. Shim. Mining optimized association rules with categorical and numeric attributes. *IEEE TKDE*, 14(1):29–50, 2002.

22. U. Rückert, L. Richter, and S. Kramer. Quantitative association rules based on half-spaces: An optimization approach. In *ICDM*, pages 507–510, 2004.

23. Ansaf Salleb-Aouissi, Christel Vrain, and Cyril Nortet. Quantminer: A genetic algorithm for mining quantitative association rules. In *IJCAI*, pages 1035–1040, 2007.

24. C. Shannon. A mathematical theory of communication, i and ii. *The Bell System Technical Journal*, pages 379–423, 623–656, 1948.

25. R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. In *Proc. of SIGMOD*, 1996.

26. A. Strehl. Relationship-based clustering and cluster ensembles for high-dimensional data mining. *PhD Thesis*, 2002.

27. C. Studholme, D.J. Hawkes, and D.L.G. Hill. An overlap invariant entropy measure of 3d medical image alignment. *Pattern Recognition*, pages 71–86, 1999.

28. K. Wang, S. H. W. Tay, and B. Liu. Interestingness-based interval merger for numeric association rules. In *KDD*, pages 121–128, 1998.

29. G. I. Webb. Discovering associations with numeric variables. In *KDD*, pages 383–388, 2001.

30. M. J. Zaki and K. Gouda. Fast vertical mining using diffsets. In *KDD*, pages 326–335, 2003.

31. H. Zhang, B. Padmanabhan, and A. Tuzhilin. On the discovery of significant statistical quantitative rules. In *KDD*, pages 374–383, 2004.

32. Z. Zhang, Y. Lu, and B. Zhang. An effective partitioning-combining algorithm for discovering quantitative association rules. In *PAKDD*, pages 241–251, 1997.

33. CHEN Zi-Yang and LIU Guo-Hua. Quantitative association rules mining methods with privacy-preserving. In *PDCAT '05: Proceedings of the Sixth International Conference on Parallel and Distributed Computing Applications and Technologies*, pages 910–912, 2005.