Alternative Interest Measures for Mining Associations in Databases

Edward R. Omiecinski, *Member*, *IEEE Computer Society*

Abstract—Data mining is defined as the process of discovering significant and potentially useful patterns in large volumes of data. Discovering associations between items in a large database is one such data mining activity. In finding associations, support is used as an indicator as to whether an association is interesting. In this paper, we discuss three alternative interest measures for associations: any-confidence, all-confidence, and bond. We prove that the important downward closure property applies to both all-confidence and bond. We show that downward closure does not hold for any-confidence. We also prove that, if associations have a minimum all-confidence or minimum bond, then those associations will have a given lower bound on their minimum support and the rules produced from those associations will have a given lower bound on their minimum confidence as well. However, associations that have that minimum support (and likewise their rules that have minimum confidence) may not satisfy the minimum all-confidence or minimum bond constraint. We describe the algorithms that efficiently find all associations with a minimum all-confidence or minimum bond and present some experimental results.

Index Terms—Data mining, associations, interest measures, databases, performance.

1 Introduction

The past few years has seen a tremendous interest in the area of data mining. Data mining is generally thought of as the process of finding hidden, nontrivial, and previously unknown information in a large collection of data [22]. Exploiting large volumes of data for superior decision making by looking for interesting patterns in the data has become a main task in today's business environment. In particular, finding associations between items in a database of customer transactions, such as the sales data collected at super market check out counters [3], [5], [11], [13], [14], [17], [20], [25], [26], [27], [28] has become an important data mining task. Association rules identify items that are most often bought along with certain other items by a significant fraction of the customers. For example, we may find that "95 percent of the customers who bought bread also bought milk." A rule may contain more than one item in the antecedent and the consequent of the rule. Every rule must satisfy two user specified constraints: one is a measure of statistical significance called *support* and the other a measure of goodness of the rule called *confidence*.

In this paper, we concentrate on finding associations, but with a different slant. That is, we take a different view of significance. Instead of *support*, we consider three other measures, which we call *any-confidence*, *all-confidence*, and *bond*. We show that these are other measures of significance that have their place in mining associations that are interesting. All three measures are indicators of the degree to which items in an association are related to each other. These measures also resemble the similarity measures used in information retrieval systems, (i.e., the degree to which documents (terms) in a collection are related to each other).

• *The author is with the College of Computing, Georgia Institute of Technology, Atlanta, GA 30332. E-mail: edwardo@cc.gatech.edu.*

Manuscript received 26 Jan. 2000; revised 6 Oct. 2000; accepted 21 June 2001. For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number 111313.

In the next section, we review the necessary background for studying the association rule problem and some of the related work. In Section 3, we present the intuition behind our interest measures and in Section 4, we formally define our interest measures and prove some important properties about them. In Section 5, we highlight the differences between bond and all-confidence as well as some other metrics. In Section 6, we present the algorithms for bond and all-confidence and in Section 7, we present a performance study of our algorithms.

2 Background

Formally, the association rule problem can be stated as follows [3], [5]: Let $\mathcal{I} = \{i_1, i_2, \ldots, i_m\}$ be a set of m distinct literals called *items*. $\mathcal{D}$ is a set of variable length transactions over $\mathcal{I}$. Each transaction *contains* a set of items $i_i, i_j, \ldots, i_k \subset \mathcal{I}$. A transaction also has an associated unique identifier called TID. An *association rule* is an implication of the form $X \longrightarrow Y$, where $X, Y \subset \mathcal{I}$ and $X \cap Y = \emptyset$. X is called the antecedent and Y is called the consequent of the rule.

In general, a set of items (such as the antecedent or the consequent of a rule) is called an *itemset*. The number of items in an itemset is called the *length* of an itemset. Itemsets of some length k are referred to as k-itemsets. For an itemset $X \cup Y$, if Y is an m-itemset then Y is called an $m\text{-}extension$ of X.

Each itemset has an associated measure of statistical significance called $support$. For an itemset $X \subset \mathcal{I}$, $support(X) = s$, if the fraction of transactions in $\mathcal{D}$ containing X equals s. A rule has a measure of its strength called *confidence* defined as the ratio $support(X \cup Y)/support(X)$.

The problem of mining association rules is to generate all rules that have support and confidence greater than some user specified minimum support and minimum confidence thresholds, respectively. This problem can be decomposed into the following subproblems:

1. All itemsets that have support above the user specified minimum support are generated. These itemsets are called the *large* itemsets. All others are said to be *small*.
2. For each large itemset, all the rules that have minimum confidence are generated as follows: for a large itemset X and any $Y \subset X$, if $support(X)/support(X-Y) \geq$ *minimum_confidence*, then the rule $X - Y \longrightarrow Y$ is a valid rule.

To reduce the combinatorial search space, algorithms exploit the following property, called *antimonotonicity* [19]: whenever the support of a set S of items violates the frequency constraint (i.e., the support falls below the specified threshold), then all supersets of S must also violate the frequency constraint. Some researchers refer to an equivalent property called *downward closure* [7]: if an itemset is large then every subset of that large itemset must also be large. The *antimonotonicity* or equivalently, the *downward closure* property is used by existing algorithms for mining association rules (e.g., the Apriori algorithm [5]) as follows. Initially, support for all itemsets of length one (1-itemsets) are tested by scanning the database. The itemsets that are found to be small are discarded. A set of 2-itemsets called *candidate itemsets* are generated by extending the large 1-itemsets generated in the previous pass by one (1-extensions) and their support is tested by scanning the database. Itemsets that are found to be large are again extended by one and their support is tested. In general, some kth iteration contains the following steps:

1. The set of candidate k-itemsets is generated by 1-extensions of the large $(k-1)$-itemsets generated in the previous iteration.
2. Supports for the candidate k-itemsets are generated by a pass over the database.
3. Itemsets that do not have the minimum support are discarded and the remaining itemsets are called large k-itemsets.

This process is repeated until no more large itemsets are found.

Recent work [7], [8], [18] deals with finding rules based on other metrics besides support and confidence. Other current work deals with efficiently supporting constraints on the antecedent and/or consequent for association rule mining [15], [19]. Still, other work involves computing large itemsets online [13], computing association rules online [1], mining for negative associations [26], and parallel mining of association rules [4], [10]. Since our work is concerned with alternative measures of interestingness, we will briefly review some of the work which is most closely related.

In [7], the authors mine association rules that identify correlations and consider both the absence and presence of items as a basis for generating the rules. The measure of significance of associations that is used is the *chi-squared test* for correlation from classical statistics. In [8], the authors still use support as part of their measure of interest of an association. However, when rules are generated, instead of using confidence, the authors use a metric they call *conviction*, which is a measure of implication and not just cooccurrence. In [18], the authors also look at alternative measures of interest, namely the *gini index*, *entropy gain*, and *chi-squared*. The problem examined in [18] is to find association rules that segment large categorical databases into two parts which are optimal according to some objective function. The functions used are information-theoretic measures which are used to indicate the extent of which the divided data distribution differs from the original data distribution. In [6], the notion of mining optimized rules is presented where the authors show that rules which satisfy a number of different interest metrics such as support. Confidence, entropy, chi-squared, and conviction reside along a support/confidence border. Hence, mining rules along this border will retrieve rules satisfying all the above metrics.

In [16], the authors present an approach to the *rare item problem*. The dilemma that arises in the *rare item problem* is that searching for rules that involve infrequent (i.e., rare) items requires a low support but using a low support will typically generate many rules that are of no interest. Using a high support typically reduces the number of rules mined but will eliminate the rules with rare items. The authors attack this problem by allowing users to specify different minimum supports for the various items in their mining algorithm. So, frequent items may have high support and infrequent items low support. They generate large itemsets with possible combinations of frequent and rare items based on their *sorted closure property*. As we will see, our metrics will also allow us to find infrequent associations that may be interesting to the user but by using one minimum threshold value.

3 Any-confidence, All-confidence, and Bond as Interest Measures

Any-confidence is our first measure of the interestingness of an association. With this measure, an association is deemed interesting if **any** rule that can be produced from that association has a confidence greater than or equal to our minimum any-confidence value. Any-confidence is like the Overlap similarity coefficient [24] in information retrieval systems. In current association mining algorithms, this would be the same as saying that we want all rules that have a confidence greater than or equal to the minimum confidence, without regard to any support criteria. However, the problem with this as mentioned in other work [7], is that it cannot be computed efficiently. As we will later see, it does not satisfy the important downward closure property. We present this measure simply for completeness and to show the relationship of it with our other measures of interestingness.

All-confidence is our second measure of association interestingness and is a variation of the first. With this measure, an association is deemed interesting if **all** rules that can be produced from that association have a confidence greater than or equal to our minimum all-confidence value. This indicates that there is a dependency between all of the items in the association. The degree of the dependency, of course, is based on the threshold value. For example, if the all-confidence threshold is one, then, for any itemset $\mathcal{L}$, which satisfies the threshold, any subset of $\mathcal{L}$ would imply the remaining items with a confidence of 100 percent. Certainly, in that case there is a high degree of dependency between the items in $\mathcal{L}$. However, if the all-confidence threshold is 0.5, then any subset of $\mathcal{L}$ would

imply the remaining items with a confidence of at least 50 percent. There still exists a dependency between the items in $\mathcal{L}$, but to a lesser degree. In contrast to the any-confidence measure, all-confidence can be computed efficiently. In other words, all-confidence satisfies the downward closure property as we will later show.

For an example of the use of all-confidence, consider data collected about a particular part maufactured by a company. Assume we have data about n occurrences of that part and a small number of those occurrences, ϵ, show all three defects: $\{D_1, D_2, D_3\}$. It may be that ϵ/n is much lower than the minimum support needed to produce an association between D_1, D_2, and D_3. So, an association involving the three defects would be considered uninteresting. However, it may be that an occurence of any one of the defects occurs in no more than $\epsilon + \tau$ parts, where $\tau \ll \epsilon$. This would be indicative of a 3-way dependence between the defects and could be of interest to the data miner. For our purpose, a 3-way dependence (or a general n-way dependence) refers to the fact that any combination of the three (n) defects implies the remaining defects with a confidence of at least $\epsilon/(\epsilon + \tau)$. This is the type of relationship (pattern) that all-confidence will find.

Bond is our third measure of the interestingness of an association. It is similar to the Jaccard similarity coefficient [24] in information retrieval systems and to the support coverage ratio [9] used for web mining. With regard to data mining, it is similar to support but with respect to a subset of the data rather than the entire data set. This has similarities to the work in [23] except in their work they define data subsets based on the data satisfying certain time constraints. The idea is to find all itemsets that are frequent in a set of user-defined time intervals. In our case, the characteristics of the data define the subsets not the end-user.

For a practical example of the use of Bond, consider a medical application where we have n patients, a small number of those patients, ρ, exhibit at least one of the three symptoms X, Y, and Z and a number of those patients, ϵ, exhibit all three symptoms X, Y, and Z. It may be that ϵ/n is lower (even much lower) than the minimum support needed to produce an association between X, Y, and Z. Hence, that association would be deemed uninteresting. However, a physician may still be interested in that association if ϵ is close to ρ, that is ϵ/ρ is greater than or equal to some minimum value. The relationship of ϵ/ρ is what we call *bond*. To be more concrete, consider a data file with 10,000 patients where five of those patients exhibit a specific set of symptoms, S. It may also be that the number of patients that exhibit any of those specific symptoms S is 10. The support for an association containing the symptoms in S would only be 0.0005. However, the bond would be 0.5.

4 Formal Properties of Any-confidence, All-confidence, and Bond

In this section, we present a formal definition of any-confidence, all-confidence, and bond, and prove a number of properties about them. Regardless of the measure of interestingness, it is important to be able to efficiently determine the itemsets that have a value (for that measure) greater than the minimum threshold. To accomplish this, we would like to be able to prune the space of possible itemsets. This was done with respect to support for the Apriori algorithm [5] which used the property that if a set of items is not a frequent itemset, then any superset of that set is not a frequent itemset.

TABLE 1
Set of Five Transactions (Items Per Transaction Indicated by a 1)

	Items						
Transaction	A	B	C	D	E	F	G
T_1	1	1	0	0	0	0	0
T_2	1	1	1	0	0	0	1
T_3	0	0	1	1	0	0	0
T_4	0	0	1	1	0	0	0
T_5	0	0	0	0	1	1	0

We previously defined the set of m items $\mathcal{I}$ as $\{i_1, i_2, \ldots, i_m\}$ and the set of variable length transactions over $\mathcal{I}$ as $\mathcal{D}$. Each transaction contains a set of items which are a subset of $\mathcal{I}$. In the following definitions, we use $\mathcal{P}(\mathcal{L})$ to represent the power set of $\mathcal{L}$, i.e., the set of all subsets of $\mathcal{L}$. It is important to point out that the use of the power set in the following definitions is used to more clearly convey the meaning of the metrics. In the algorithm, which implements our metrics, the power set will not be computed. Since we will be comparing our metrics to support, we will start by defining support using the notation we will use for defining the other metrics.

Definition: *The support of a set of items, $\mathcal{L}$ is*

$$\frac{|\{d \mid d \in \mathcal{D} \wedge \mathcal{L} \subset d\}|}{|\mathcal{D}|}.$$

The numerator represents the number of transactions where each transaction contains the set of items $\mathcal{L}$. The denominator is simply the total number of transactions.

Definition. *The any-confidence of a set of items, $\mathcal{L}$ is*

$$\frac{|\{d \mid d \in \mathcal{D} \wedge \mathcal{L} \subset d\}|}{MIN\{i \mid \forall l (l \in \mathcal{P}(\mathcal{L}) \wedge l \neq \emptyset \wedge l \neq \mathcal{L} \wedge i = |\{d \mid d \in \mathcal{D} \wedge l \subset d\}|)\}}.$$

The denominator is the minimum count of transactions that contain any subset of $\mathcal{L}$, excluding the empty set and the improper subset. This formal definition simply states that any-confidence is the largest confidence of any rule for the set of items, $\mathcal{L}$. Ideally, this measure allows a data miner to focus on rules that have high confidence without regard to any other stipulations about the data (e.g., ignoring support). However, it cannot be determined efficiently by reducing the search space of alternatives or, in other words, does not satisfy the downward closure property. Although similar claims have been shown in [7], but with a different name, we include a simple counter example to the downward closure property here for completeness. As an example, consider a database with the following transactions (also shown in Table 1) $T_1 = \{A, B\}$, $T_2 = \{A, B, C, G\}$, $T_3 = \{C, D\}$, $T_4 = \{C, D\}$, and $T_5 = \{E, F\}$, where $\mathcal{I}$ is $\{A, B, C, D, E, F\}$. Suppose that the minimum any-confidence which is required is one. Using these transactions, any-confidence ($\{A, C\}$) is $1/2$, since $\{A, C\}$ appears in one transaction, $\{A\}$ appears in two transactions, and $\{C\}$ appears in three transactions. Although the itemset $\{A, C\}$ does not satisfy the minimum threshold, we see that an extension of this itemset does, namely $\{A, C, G\}$,

TABLE 2
Support, B ond, All-Confidence, and Any-Confidence Values Using Data from Table 1

Itemset	Support	Bond	All-confidence	Any-confidence
A	2/5	1	1	1
B	2/5	1	1	1
C	3/5	1	1	1
D	2/5	1	1	1
E	1/5	1	1	1
F	1/5	1	1	1
G	1/5	1	1	1
AB	2/5	1	1	1
AC	1/5	1/4	1/3	1/2
AG	1/5	1/2	1/2	1
BC	1/5	1/4	1/3	1/2
BG	1/5	1/2	1/2	1
CD	2/5	2/3	2/3	1
CG	1/5	1/3	1/3	1
EF	1/5	1	1	1
ABC	1/5	1/4	1/3	1
ABG	1/5	1/2	1/2	1
ACG	1/5	1/4	1/3	1
BCG	1/5	1/4	1/3	1
ABCG	1/5	1/4	1/3	1

where any-confidence$(\{A, C, G\})$ *is* 1. *Hence, any-confidence will not be used as a measure of interestingness in our work.*

Definition. *The all-confidence of a set of items,* $\mathcal{L}$ *is*

$$\frac{|\{d \mid d \in \mathcal{D} \wedge \mathcal{L} \subset d\}|}{MAX\{i \mid \forall l (l \in \mathcal{P}(\mathcal{L}) \wedge l \neq \emptyset \wedge l \neq \mathcal{L} \wedge i = |\{d \mid d \in \mathcal{D} \wedge l \subset d\}|)\}}.$$

The denominator is the maximum count of transactions that contain any subset of $\mathcal{L}$, *excluding the empty set and the improper subset. We should note that the maximum value will occur when the subset of* $\mathcal{L}$ *consists of a single item. Adding additional items cannot increase the count of transactions. Hence, the power set need not be computed. This formal definition simply states that all-confidence is the smallest confidence of any rule for the set of items,* $\mathcal{L}$. *That is, all rules produced from this item set would have a confidence greater than or equal to its all-confidence value.*

Definition. *The bond of a set of items,* $\mathcal{L}$ *is*

$$\frac{|\{d \mid d \in \mathcal{D} \wedge \mathcal{L} \subset d\}|}{|\{d \mid d \in \mathcal{D} \wedge \exists l (l \in \mathcal{P}(\mathcal{L}) \wedge l \neq \emptyset \wedge l \subset d)\}|}.$$

This formal definition simply states that bond is the ratio of the cardinality of the set of transactions that contain all items in $\mathcal{L}$ *and the cardinality of the union of transactions that contain any item of* $\mathcal{L}$. *In the algorithm, which implements the bond metric, the power set is not computed, instead each transaction is checked to see if it contains any of the items in* $\mathcal{L}$. *We should note that the bond of* $\mathcal{L}$ *where* $|\mathcal{L}| = 1$ *is one. If a set of items* $\mathcal{L}$ *does not appear in any transaction, then the bond of* $\mathcal{L}$ *is zero. Once again, consider the database shown in Table 1. The support and bond for all itemsets with a nonzero support are shown in Table 2.*

The relationship between the associations that satisfy the different metrics is displayed in Fig. 1. From a practical point, the relationship between all-confidence and bond is important. This relationship tells us that if we compute associations using the all-confidence metric for a minimum value v, then the resulting associations will include those that satisfy the bond criteria for the value v. To prove the relationship between the three metrics, we present the following lemma.

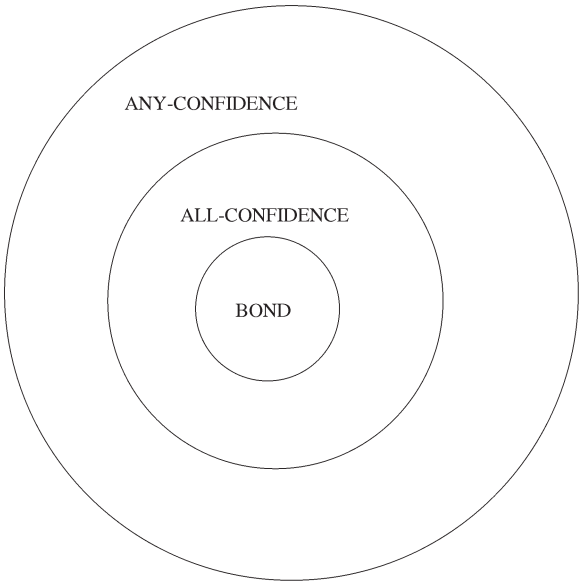

Fig. 1. Relationship between associations produced by the three metrics.

Lemma 1. *Given an itemset* $\mathcal{L}$, *the following relationship holds between the metrics as applied to* $\mathcal{L}$: *any-confidence* $(\mathcal{L}) \geq$ *all-confidence* $(\mathcal{L}) \geq$ *bond*$(\mathcal{L})$.

Proof. All three metrics have the same numerator and the relationship between the denominator of any-confidence ($\mathcal{L}$), all-confidence ($\mathcal{L}$), and bond ($\mathcal{L}$), for $|\mathcal{L}| = k$, where $n = 2^k - 2$, is

$$\begin{aligned} &min(|A_1|, |A_2|, \ldots, |A_n|) \\ &\leq max(|A_1|, |A_2|, \ldots, |A_n|) \\ &\leq |A_1 \cup A_2 \cup \ldots \cup A_n|, \end{aligned}$$

where A_i represents the set of transactions that contain a subset of the items in $\mathcal{L}$. □

An important property for any measure of interestingness is *downward closure*. We present the following two lemmas and their proofs to show that the *downward closure* property with respect to all-confidence and bond holds. This will allow us to discard any itemset that does not meet the minimum all-confidence (or bond) threshold. Three basic properties are used in the lemmas and are proven in the appendix. Similar observations for our all-confidence metric, but with regard to mining with constraints, have appeared in [21].

Lemma 2. *The downward closure property holds with respect to all-confidence. That is, If* $\mathcal{L}$ *is an itemset and all-confidence*$(\mathcal{L})$ *is greater than or equal to minall then the all-confidence of every subset,* $\mathcal{L}'$, *of* $\mathcal{L}$ *will be greater than or equal to minall. More formally, if*

$$\frac{|\{d \mid d \in \mathcal{D} \wedge \mathcal{L} \subset d\}|}{MAX\{i \mid \forall l(l \in \mathcal{P}(\mathcal{L}) \wedge l \neq \emptyset \wedge l \neq \mathcal{L} \wedge i = |\{d \mid d \in \mathcal{D} \wedge l \subset d\}|)\}} \geq minall.$$

Then, $\forall \mathcal{L}' \subset \mathcal{L}$

$$\frac{|\{d \mid d \in \mathcal{D} \wedge \mathcal{L}' \subset d\}|}{MAX\{i \mid \forall l(l \in \mathcal{P}(\mathcal{L}') \wedge l \neq \emptyset \wedge l \neq \mathcal{L}' \wedge i = |\{d \mid d \in \mathcal{D} \wedge l \subset d\}|)\}} \geq minall.$$

Proof. Since $\mathcal{L}' \subset \mathcal{L}$, we know that

$$|\{d \mid d \in \mathcal{D} \wedge \mathcal{L}' \subset d\}| \geq |\{d \mid d \in \mathcal{D} \wedge \mathcal{L} \subset d\}|.$$

This is Property 1 in the Appendix. Also, since $\mathcal{P}(\mathcal{L}') \subset \mathcal{P}(\mathcal{L})$, we know that

$$\begin{aligned} &MAX\{i \mid \forall l(l \in \mathcal{P}(\mathcal{L}') \wedge l \neq \emptyset \wedge l \neq \mathcal{L}' \wedge i \\ &= |\{d \mid d \in \mathcal{D} \wedge l \subset d\}|)\} \\ &\leq MAX\{i \mid \forall l(l \in \mathcal{P}(\mathcal{L}) \wedge l \neq \emptyset \wedge l \neq \mathcal{L} \wedge i \\ &= |\{d \mid d \in \mathcal{D} \wedge l \subset d\}|)\}. \end{aligned}$$

This is Property 2 in the Appendix. So, all-confidence ($\mathcal{L}'$) $\geq$ all-confidence ($\mathcal{L}$) $\geq minall$. □

Lemma 3. *The downward closure property holds with respect to bond. That is, if $\mathcal{L}$ is an itemset and $bond(\mathcal{L})$ is greater than or equal to $minbond$, then, the bond of every subset, $\mathcal{L}'$ of $\mathcal{L}$ will be greater than or equal to $minbond$. More formally, if*

$$\frac{|\{d \mid d \in \mathcal{D} \wedge \mathcal{L} \subset d\}|}{|\{d \mid d \in \mathcal{D} \wedge \exists l(l \in \mathcal{P}(\mathcal{L}) \wedge l \neq \emptyset \wedge l \subset d)\}|} \geq minbond.$$

Then, $\forall \mathcal{L}' \subset \mathcal{L}$

$$\frac{|\{d \mid d \in \mathcal{D} \wedge \mathcal{L}' \subset d\}|}{|\{d \mid d \in \mathcal{D} \wedge \exists l(l \in \mathcal{P}(\mathcal{L}') \wedge l \neq \emptyset \wedge l \subset d)\}|} \geq minbond.$$

Proof. Since $\mathcal{L}' \subset \mathcal{L}$, we know that

$$|\{d \mid d \in \mathcal{D} \wedge \mathcal{L}' \subset d\}| \geq |\{d \mid d \in \mathcal{D} \wedge \mathcal{L} \subset d\}|.$$

This is Property 1 in the Appendix. Also, since $\mathcal{P}(\mathcal{L}') \subset \mathcal{P}(\mathcal{L})$, we know that

$$\begin{aligned} &|\{d \mid d \in \mathcal{D} \wedge \exists l(l \in \mathcal{P}(\mathcal{L}') \wedge l \neq \emptyset \wedge l \subset d)\}| \\ &\leq |\{d \mid d \in \mathcal{D} \wedge \exists l(l \in \mathcal{P}(\mathcal{L}) \wedge l \neq \emptyset \wedge l \subset d)\}|. \end{aligned}$$

This is Property 3 in the Appendix. So,

$$bond(\mathcal{L}') \geq bond(\mathcal{L}) \geq minbond.$$

□

The following two lemmas provide information about the relationship between bond and support. Lemma 4 shows that the support for an itemset will be less than or equal to the bond of the itemset. Lemma 5 shows that the minimum support for an itemset that satisfies a minimum bond threshold can be as low as the smallest possible support.

Lemma 4. *The support for a set of items, $\mathcal{L}$, will be less than or equal to the bond for $\mathcal{L}$.*

Proof. This can be seen directly from the definition of bond and support. Since the number of transactions that contain subsets of $\mathcal{L}$ must be less than or equal to the total number of transactions, we have that

$$|\{d \mid d \in \mathcal{D} \wedge \exists l(l \in \mathcal{P}(\mathcal{L}) \wedge l \neq \emptyset \wedge l \subset d)\}| \leq |\mathcal{D}|.$$

Hence, $support(\mathcal{L}) \leq bond(\mathcal{L})$. □

Lemma 5. *The support for a set of items, $support(\mathcal{L})$, where $bond(\mathcal{L})$ is greater than or equal to any minimum bond threshold, can be as low as $1/|\mathcal{D}|$.*

Proof. Consider a set of transactions $\mathcal{D}$, itemset $\mathcal{L}$ and any minimum bond value $minbond$ where $0 < minbond \leq 1$. Since the greatest value for the minimum bond threshold is one, any itemset with a bond value of one will satisfy every minimum bond requirement for itemsets. Consider a transaction that contains the items in $\mathcal{L}$ and the items in $\mathcal{L}$ appear in only one transaction. This gives a $bond(\mathcal{L})$ of one. The $support(\mathcal{L})$ is $1/|\mathcal{D}|$. Hence, for $bond(\mathcal{L}) \geq minbond$ we have $support(\mathcal{L}) = 1/|\mathcal{D}|$. □

The significance of Lemma 5 is that using a metric based on support to find large itemsets that meet a given bond threshold can be extremely inefficient. That is, using support to find the itemsets that satisfy any minimum bond threshold can result in generating every possible association. Whereas, by using the bond metric directly, the search space of itemsets can be pruned as indicated by Lemma 3.

We formalize the relationship between bond and confidence by way of Theorem 1.

Theorem 1. *The lower bound for the confidence of any rule produced from a set of items $\mathcal{L}$ such that $\mathcal{L}$ has $bond(\mathcal{L})$ is $minbond$.*

Proof. Suppose we have a set of items $\mathcal{L}$ such that $bond(\mathcal{L}) \geq$ minbond. The confidence of a rule, $\mathcal{L}' \longrightarrow \mathcal{L} - \mathcal{L}'$, where $\mathcal{L}' \subset \mathcal{L}$, is defined as

$$\frac{support(\mathcal{L})}{support(\mathcal{L}')}$$

which is in our notation,

$$\frac{|\{d \mid d \in \mathcal{D} \wedge \mathcal{L} \subset d\}|}{|\{d \mid d \in \mathcal{D} \wedge \mathcal{L}' \subset d\}|}.$$

The $bond(\mathcal{L})$ is

$$\frac{|\{d \mid d \in \mathcal{D} \wedge \mathcal{L} \subset d\}|}{|\{d \mid d \in \mathcal{D} \wedge \exists l(l \in \mathcal{P}(\mathcal{L}) \wedge l \neq \emptyset \wedge l \subset d)\}|}.$$

Since $\mathcal{L}' \subset \mathcal{L}$, we know the following:

$$\begin{aligned} &|\{d \mid d \in \mathcal{D} \wedge \exists l(l \in \mathcal{P}(\mathcal{L}') \wedge l \neq \emptyset \wedge l \subset d)\}| \\ &\leq |\{d \mid d \in \mathcal{D} \wedge \exists l(l \in \mathcal{P}(\mathcal{L}) \wedge l \neq \emptyset \wedge l \subset d)\}|. \end{aligned}$$

This is Property 3 in the Appendix. We also know that since $\mathcal{L}' \in \mathcal{P}(\mathcal{L}')$ that $|\{d \mid d \in \mathcal{D} \wedge \mathcal{L}' \subset d\}| \leq |\{d \mid d \in \mathcal{D} \wedge \exists l(l \in \mathcal{P}(\mathcal{L}') \wedge l \neq \emptyset \wedge l \subset d)\}|$. So,

TABLE 3
Associations with Bond $\geq$ 0.6 and Their Rules

Association	Bond	Support	Rule	Confidence
$\{A, B\}$	1	0.4		
			$A \longrightarrow B$	1
			$B \longrightarrow A$	1
$\{C, D\}$	0.66	0.4		
			$C \longrightarrow D$	0.66
			$D \longrightarrow C$	1
$\{E, F\}$	1	0.2		
			$E \longrightarrow F$	1
			$F \longrightarrow E$	1

$$\frac{|\{d \mid d \in \mathcal{D} \wedge \mathcal{L} \subset d\}|}{|\{d \mid d \in \mathcal{D} \wedge \mathcal{L}' \subset d\}|} \geq bond(\mathcal{L}) \geq minbond.$$

Hence, the confidence $(\mathcal{L}' \longrightarrow \mathcal{L} - \mathcal{L}') \geq$ minbond. □

To illustrate the meaning of Lemma 5 and Theorem 1, we examine the transactions shown in Table 1. We will use a *minbond* of 0.6, i.e., 60 percent. From Lemma 5, we know that any association that satisfies this minimum bond value will have a support of $1/5$, at least. We also know, from Theorem 1, that any rule produced by an association with minimum bond will have a confidence of at least 0.6. If we examine Table 2, we see that there are three associations (of size greater than 1), that satisfy the minimum bond requirement. They are displayed in Table 3 along with their associated rules. One point to make is that, just because an association has the lower bound for support and confidence, it does not necessarily satisfy the minbond requirement. In Table 2, all itemsets satisfy the lower bound for support but only the itemsets in Table 3 satisfy the minimum bond requirement. If we were to lower the minimum bond to 0.5, we would still have the results shown in Table 3. However, the itemset $\{A, C\}$ would not only satisfy the equivalent lower bound for support (i.e., 0.2), but also the rule $A \longrightarrow C$ would satisfy the lower bound for confidence (i.e., 0.5). However, the itemset $\{A, C\}$ would not satisfy the minimum bond requirement of 0.5. Hence, generating associations and rules that satisfy the lower bound for support and confidence would not produce only associations and rules that satisfy the minimum bond requirement. The bottom line is that the output from such an approach (e.g., a priori [5]) would be a superset of the solution but the exact subset (for the bond metric) could not be determined without having to make an additional pass over the transaction data.

5 Comparison of All-confidence, Bond, and Other Metrics

In this section, we relate some of the metrics presented in Section 1 to our own metrics of all-confidence and bond. The conviction [8] and lift [6] metrics can be used to determine which rules, generated from a large itemset, are the most interesting. Interest (lift) is a measure of departure from independence and is symmetric. Interest is defined [8] for an itemset $\{A, B\}$ as $\frac{support(\{A,B\})}{support(\{A\}) \cdot support(\{B\})}$. As such, interest is a measure of cooccurrence as is bond. Although bond does provide some measure of implication (w.r.t. confidence) as shown by Theorem 1. All-confidence also provides information about implication in that it ensures a lower bound on confidence for any rule of an itemset, which satisfies the minumim threshold for all-confidence. Consider the data in Table 4. The interest of itemset $\{A, B\}$ is computed as 1.05, which is only slightly above one (the interest for items that are independent). However, the support ($\{A, B\}$) is 0.6, the bond ($\{A, B\}$) is 0.64, and the all-confidence ($\{A, B\}$) is 0.67. The high value for these metrics indicate to the user that the items $\{A, B\}$ occur very often in the data set (according to the metric used) and may be of interest to the user. On the other hand, such a high value may indicate an obvious relationship, which the user may already know. However, if few itemsets have such a high support, bond, or all-confidence, those itemsets may be interesting. Consider a second example where the data is shown in Table 5. The interest value for itemset $\{A, B\}$ is 1.11 while the support, bond, and all-confidence values are 0.05, 0.056, and 0.056, respectively. Since the itemset $\{A, B\}$ occurs in only 5 percent of the transactions, it may be very interesting and not obvious to the user, although the interest measure indicates that A and B are independent. We should point out that the minimum threshold value for bond and all-confidence (as well as support) is domain (data) dependent. The data mining task will typically be an iterative process where the threshold value is adjusted so as to find a value where the algorithm discovers the interesting associations (not too many and not too few).

TABLE 4
Sample Itemset and Support for a File of 30,000 Transactions

Itemset	Support
A	19,000
B	27,000
AB	18,000

Conviction is a measure of implication because it is directional. It is complimentary to our metrics. All-confidence and bond find items that cooccur, and once they are found, the conviction metric could be used to determine the most interesting rules from the large itemsets.

The chi-squared [7] metric is used to determine the (in)dependence between items. It is based on statistical theory and takes into account all combinations of both the presence and absence of items. Thus, positive and negative correlations can be determined. All-confidence and bond (like support and confidence) only take into account the

TABLE 5
Sample Itemset and Support for a File of 1,000,000 Transactions

Itemset	Support
A	50,000
B	850,000
AB	50,000

TABLE 6
Two Files Containing 10,000 Transactions
(Only Three Items Are Shown)

	File 1			File 2		
Transaction	A	B	C	A	B	C
T_1	1	1	1	1	1	1
T_2	1	1	1	1	1	1
T_3	1	1	1	1	1	1
T_4	1	0	0	1	0	0
T_5	1	0	0	0	1	0
T_6	1	0	0	0	0	1
T_7	0	0	0	0	0	0
T_8	0	0	0	0	0	0
	.	.	.	.	.	.
	.	.	.	.	.	.
	.	.	.	.	.	.
T_{10000}	0	0	0	0	0	0

presence of items and requires a minimum threshold value. However, there are conditions when the chi-squared test would be appropriate for data analysis and in typical basket data analysis the necessary conditions for use do not always hold [7]. For example, when the expected values in the contingency table are small, which typically happens when the number of cells becomes large, the chi-squared statistic becomes increasingly inaccurate [7].

To conclude this section, we give a simple example comparing the bond and all-confidence metrics. The data is shown in Table 6. In both files, bond ($\{A, B, C\}$) is 0.5. However, in File 1, the all-confidence ($\{A, B, C\}$) is 0.5 while in File 2, the all-confidence ($\{A, B, C\}$) is 0.75. All of the rules generated from itemset $\{A, B, C\}$ have a confidence greater than or equal to 50 percent in File 1 and greater than or equal to 75 percent in File 2. A higher value of all-confidence indicates a greater dependency between all of the attributes in the itemset. As a further point of interest, the support ($\{A, B, C\}$) is only 0.0003.

6 Association Finding Algorithm Using All-confidence or Bond Metrics

The main task of the association finding algorithm, shown in Fig. 2, is to generate the large itemsets that satisfy either the minimum all-confidence requirement or the minimum bond requirement. We adapted our Partition algorithm [25], which efficiently computes large itemsets based on support. However, there are other newer algorithms [2], [12] that appear to be more efficent, which we could have adapted as well.

Associated with each itemset is a list, called the *tidlist*. The tidlist consists of all transaction identifiers of the transactions containing the itemset. Included with the tidlist is its size (i.e., the count of the number of transaction identifiers in the list). The count value for 1-itemsets will be used if the all-confidence metric is chosen. If the bond metric is chosen, then, also associated with an itemset is the *union_tidlist*, (i.e., the set of transactions that contain any of the individual items in that itemset). The cardinality of the *tidlist* divided by the cardinality of the *union_tidlist* is the bond for the associated itemset. The bond for an extension

```
procedure gen_large_itemsets
L_i = the set of large itemsets of length i
l_j = an individual candidate large itemset contained in L_i
BOND will be set to true if the bond metric is to be used and false
  if the all-confidence metric is to be used

1)  L_1 = {large 1-itemsets along with their tidlists and counts}
2)  for (k = 2; L_k ≠ ∅; k++) do begin
3)     forall itemsets l_1 ∈ L_{k-1} do begin
4)        forall itemsets l_2 ∈ L_{k-1} do begin
5)           if l_1[1] = l_2[1] ∧ l_1[2] = l_2[2] ∧ ... ∧ l_1[k-1] < l_2[k-1] then
6)              c = l_1[1] · l_1[2] ··· l_1[k-1] · l_2[k-1]
7)              if c cannot be pruned then
8)                 c.tidlist = l_1.tidlist ∩ l_2.tidlist
9)                 if BOND then
10)                   c.unlist = l_1[1].tidlist ∪ ... ∪ l_1[k-1].tidlist ∪ l_2[k-1].tidlist
11)                   if | c.tidlist | / | c.unlist | ≥ minbond then
12)                      L_k = L_k ∪ {c}
13)                else
14)                   allmax = MAX(L_1.count[l_1[1]], L_1.count[l_1[2]], ...,
                                 L_1.count[l_1[k-1]], L_1.count[l_2[k-1]])
15)                   if | c.tidlist | /allmax ≥ minallconf then
16)                      L_k = L_k ∪ {c}
17)       end
18)    end
19) end
20) return ∪_k L_k
```

Fig. 2. Procedure gen_large_itemsets.

of the itemset is determined as follows: suppose t_1 and t_2 are the tidlists associated with itemsets l_1 and l_2, and c_3 is an itemset obtained by extending l_1 with l_2 (as explained below). The bond for c_3 is given by the number of transactions that contain c_3 (i.e., the intersection) divided by the number of unique transactions that contain any item in c_3 (i.e., the union). The main computational difference in computing bond versus all-confidence is the cost of computing the *union_tidlist* versus the cost of finding the maximum of the counts of the 1-itemsets belonging to the current k-itemset.

For example, let $\{T_1, T_3, T_4\}$ be the list of transactions associated with itemset $\{1, 2\}$ and $\{T_1, T_4, T_7\}$ be the list associated with $\{1, 3\}$. Now, the transactions that contain the candidate itemset $\{1, 2, 3\}$ are given by the intersection of the lists of transactions associated with itemsets $\{1, 2\}$ and $\{1, 3\}$, i.e., $\{T_1, T_4\}$. Let the tidlist for itemset $\{1\}$ be $\{T_1, T_3, T_4, T_5, T_7\}$, the tidlist for itemset $\{2\}$ be $\{T_1, T_3, T_4, T_6\}$ and the tidlist for itemset $\{3\}$ be $\{T_1, T_4, T_7\}$. The bond for itemset $\{1, 2, 3\}$ is the cardinality of the intersection of tids for $\{1, 2\}$ and $\{1, 3\}$ divided by the cardinality of the union of tids for $\{1\}, \{2\}$, and $\{3\}$. If this satisfies the minimum bond then $\{1, 2, 3\}$ is a large itemset. The all-confidence value for itemset $\{1, 2, 3\}$ is the cardinality of the intersection of tids for $\{1, 2\}$ and $\{1, 3\}$ divided by the maximum of the counts of the 1-itemsets for $\{1\}, \{2\}$, and $\{3\}$. If this satisfies the minimum all-confidence then, $\{1, 2, 3\}$ is a large itemset.

Initially, a 1-itemset is created for every item in the database. The tidlists for these itemsets are generated by reading the database. For all 1-extensions (2-itemsets) of these itemsets, the tidlist is generated by intersecting the tidlists of both the itemsets in the extension. For the 2-itemsets, the union of the 1-itemsets is simply computed as the sum of the counts of the two 1-itemsets minus the count of the 2-itemset. The 2-itemsets that do not satisfy the minimum bond are discarded. The remaining itemsets are the large itemsets. These itemsets are extended by one and the process is repeated. The extensions of the itemsets are created as follows: let l_1 and l_2 be two k-itemsets, containing $\{i_j, i_k, \ldots, i_m\}$ and $\{i_p, i_q, \ldots, i_t\}$, respectively. A 1-extension of l_1 (a $(k+1)$-itemset) is generated if the following condition is satisfied: $i_j = i_p \wedge i_k = i_q \wedge \ldots \wedge i_m \leq i_t$. The $(k+1)$-itemset consists of $\{i_j, i_k, \ldots, i_m, i_t\}$. This technique is similar to the candidate generation step described in [5].

For fast computation of the intersection, the tidlists are maintained as arrays and the sort-merge join algorithm is used. Recall that the *TID*s are in ascending order in the database. Hence, the tidlists are in the sort order initially and all resulting tidlists are automatically generated in the sort order. This operation is of linear complexity on the length of the tidlist.

In our implementation, the tidlists of itemsets of length greater than one are not materialized. For example, to compute the support for $\{A, B, C, D\}$, the tidlists for A, B, C, and D are intersected. No tidlist is generated for the itemset $\{A, B, C, D\}$. The advantage of this approach is that we need storage for the tidlists of only the 1-itemsets and, hence, the memory requirement can be estimated quite accurately.

The procedure `gen_large_itemsets` generates all large itemsets (of all lengths). The procedure is the same as used in our previous work [25]. The prune step is performed as follows:

prune(c: k-itemset)
forall $(k-1)$-subsets s of c do
  if $s \notin L_{k-1}$ **then**
    return "c can be pruned"

The prune step eliminates extensions of $(k-1)$-itemsets which are not found to be large, from being considered for calculating the bond. For example, if L_3 is found to be $\{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 3, 5\}, \{2, 3, 4\}\}$, the candidate generation initially generates the itemsets $\{1, 2, 3, 4\}$ and $\{1, 3, 4, 5\}$. However, itemset $\{1, 3, 4, 5\}$ is pruned since $\{1, 4, 5\}$ is not in L_3. This technique is same as the one described in [5] except in our case, as each candidate itemset is generated, its bond is determined immediately.

7 Performance Results

In this section, we describe the experimental results of using our technique for generating associations with a minimum bond. We performed two sets of experiments, one using synthetic data and the other using a subset of the 1990 United States census data. We should point out that, if the threshold value is set too low, then, many large itemsets will be produced and this will negatively impact our algorithm's performance. This is also true for the a priori algorithm [5]. However, there are newer and more efficient algorithms such as the FP-tree [12] that we could adapt for use with our interest measures, in place of our Partition algorithm [25].

7.1 Synthetic Data

The synthetic data is generated such that it simulates customer buying patterns in a retail market environment. We have used the same basic method as described in [27]. All of the synthetic data sets consisted of 100,000 transactions taken over 1,000 items. The data labeled *T10.I4* had an average transaction size of 10 and a maximum transaction size of 40. The data labeled *T20.I4* had an average transaction size of 20 and a maximum transaction size of 50. The data labeled *T10.I4Y* consisted of 99,900 transactions generated by the synthetic data generator and 100 additional transactions. Those 100 transactions were made up of subsets of seven items which only appear in those 100 transactions. All of the 100 transactions contain the same three items and a random number of the remaining four items.

A comparison of the running time for the algorithm using the all-confidence metric for data sets *T10.I4* and *T20.I4* is shown in Fig. 3. A comparison of the algorithm's running time using bond for data sets *T10.I4* and *T20.I4* is shown in Fig. 4. The amount of data processed (in bytes) for *T20.I4* was approximately twice the amount of data processed for *T10.I4*. This was simply due to the larger average transaction size. In Fig. 3 and Fig. 4, we see that the running time for each given data set was fairly constant, regardless of the all-confidence or bond value. The reason for this is due to the relatively small number of large

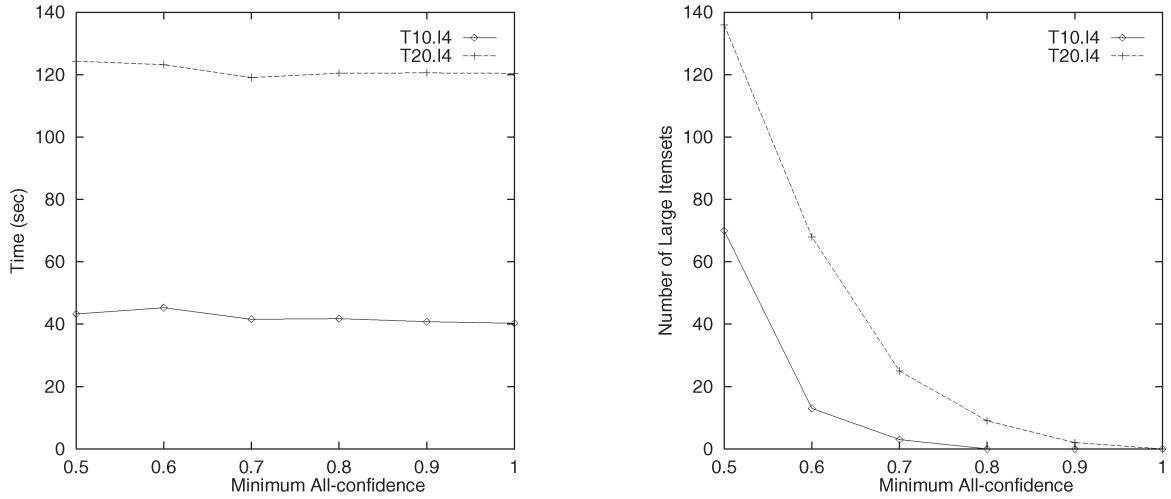

Fig. 3. Algorithm performance with all-confidence metric using synthetic data.

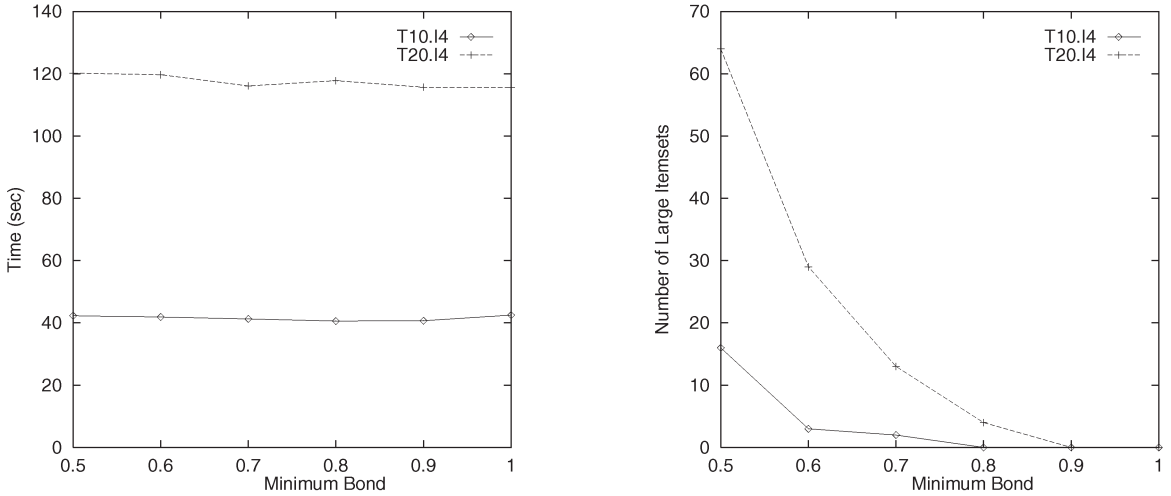

Fig. 4. Algorithm performance with bond metric using synthetic data.

itemsets generated for any of the desired all-confidence or bond values. For the *T10.I4* data set and the all-confidence metric, the number of large itemsets ranged from zero to 70. For the *T20.I4* data set and the all-confidence metric, the number of large itemsets ranged from zero to 136. For the *T10.I4* data set and the bond metric, the number of large itemsets ranged from zero to 16. For the *T20.I4* data set and the bond metric, the number of large itemsets ranged from zero to 64. If the number of large itemsets were to increase dramatically, the running time would do so as well. This can be seen in the association finding algorithms that use support as well.

The results of running the algorithm for data set *T10.I4Y* using all-confidence is shown in Fig. 5 and using bond is shown in Fig. 6. In these experiments, we intentionally placed sets of items in transactions so as to satisfy the bond requirement and hence satisfy the all-confidence metric as well. The number of large itemsets varied from four for a bond of 1.0 to 44 for a bond of 0.5. For all-confidence, the number of large itemsets varied from four for a value of 1.0 to 98 for a value of 0.5. Once again, since the number of large itemsets did not vary much, the running times remained fairly constant.

In Table 7, we show what the corresponding minimum support would be for the large itemsets that were determined based on bond. For a minimum bond value of 0.5, the algorithm determined 27 large itemsets of size two, of which the minimum support was 0.02 percent.

7.2 Census Data

The data used in the next set of experiments was obtained from the US Census Bureau through their online data extraction system available on the Web at www.census.gov/DES/www/welcome.html. The data is a subset of the 1990 Decennial Census Public Use Microdata 5 percent Samples. The data consisted of 53,847 records for people living in Florida of Hispanic origin. For these experiments, we chose a subset of the available record fields, which included *age, citizenship, disability1, disability2, English, fertile, Hispanic origin, hours89, income1, language, martial, means, military, race, sex, year, school, and immigrated*. Since the fields were not all Boolean valued, we converted the numeric values into disjoint ranges and associated a unique field

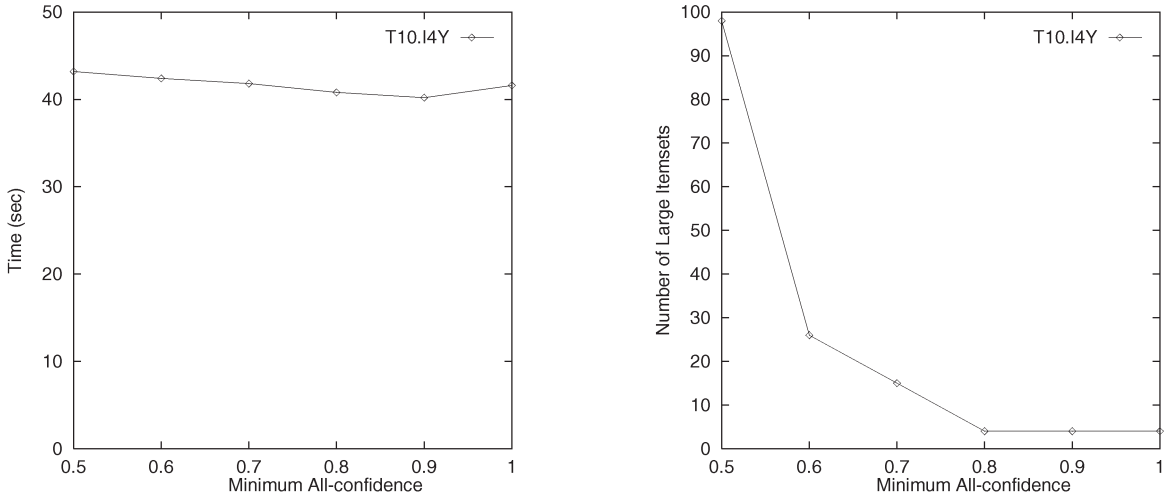

Fig. 5. Algorithm performance with all-confidence metric using synthetic data.

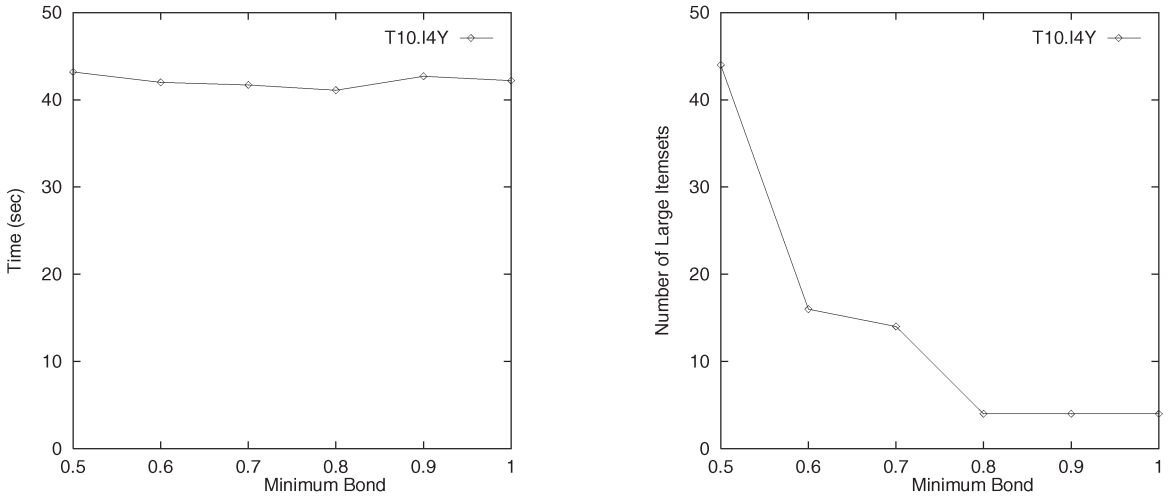

Fig. 6. Algorithm performance with bond metric using synthetic data.

with each. The ranges were chosen based on the online summaries provided by the US Census Bureau. The data was converted into 118 items, but each record only contained a maximum of 20 items.

The result of running the association finding algorithm for this census data subset using the all-confidence metric is shown in Fig. 7 and using the bond metric in Fig. 8. The number of large itemsets varied from four for an all-confidence value of 1.0 to 183 for a value of 0.5. The number of large itemsets varied from four for a bond of 1.0 to 102 for a bond of 0.5. In these experiments, the running time using the different all-confidence and bond values was not relatively constant (as with the synthetic data) since the number of large itemsets increased much more with a lower bond value. An interesting point about the all-confidence metric can be seen when we compare the results from Fig. 7 and Fig. 8. Although the bond metric generates fewer itemsets than the all-confidence metric (e.g., for a value of 0.5, bond produces 102 and all-confidence produces 183), the running time using bond is much higher. This is due to the fact that the all-confidence metric (i.e., the denominator) is computed once for the 1-itemsets and reused for larger itemsets whereas, the bond metric involves a union operation that has to be computed for each itemset.

If we examine the associations produced for a minimum bond value of 1.0, and look at the largest association (i.e., size three) produced, we see that it includes the following items: *work limitation status is not applicable, person is less than 16 years of age AND work prevention status is not applicable, person is less than 16 years of age AND military service is not applicable, person is less than 16 years of age*. These three items appeared in 11,427 records out of the 53,847 records. The same set of associations is obtained using 1.0 as the minimum all-confidence value. The results show us that the three items always appeared together in the data. We also see that these highly correlated items are not very interesting.

If we examine some of the associations produced for a lower minimum bond value, we find somewhat more interesting associations. For example, with a minimum bond value of 0.7, one association that was found was the following: *not limited from working AND not prevented from working AND speaks another language*. If we look at the

TABLE 7
Minimum Support (in Percent) and Count for Large Itemsets with Minimum Bond

Minimum Bond	large itemset size 2		3		4		5	
	sup	count	sup	count	sup	count	sup	count
1.0	0.1	3	0.1	1				
0.9	0.1	3	0.1	1				
0.8	0.1	3	0.1	1				
0.7	0.06	9	0.08	4	0.08	1		
0.6	0.04	11	0.08	4	0.08	1		
0.5	0.02	27	0.06	11	0.06	5	0.06	1

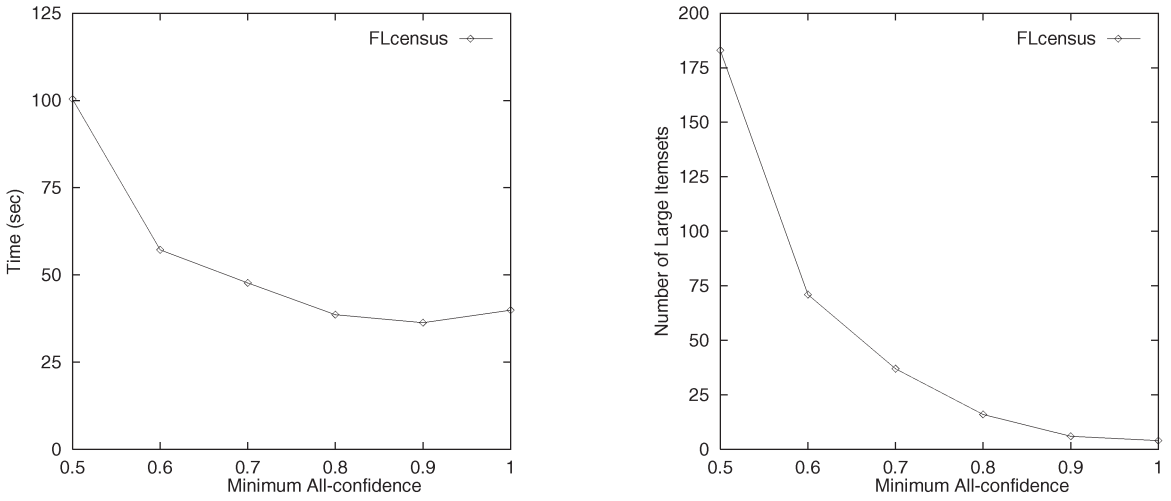

Fig. 7. Algorithm performance with all-confidence metric using US census data.

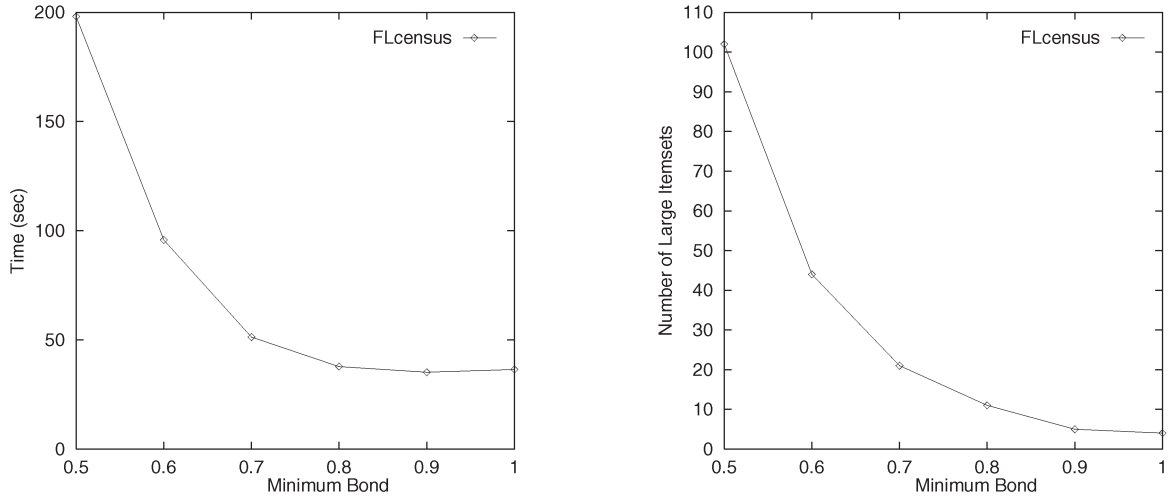

Fig. 8. Algorithm performance with bond metric using US census data.

associations produced for all-confidence, using 0.7, considering only itemsets of length four, we have:

- *not limited from working AND not prevented from working AND speaks another language AND no military service and*
- *not limited from working AND not prevented from working AND no military service AND immigrated to the United States.*

For a minimum bond value of 0.5, some of the associations included *Hispanic origin is Puerto Rican AND born in Puerto Rico*. About half of the people of Puerto Rican origin were born in Puerto Rico. A corresponding association was not found for persons of other Hispanic origin such as Mexican or Cuban. Another sample association was *Hispanic origin is Cuban AND speaks another language*. Of the 32,934 persons of Cuban origin and the 45,000 people that

speak another language, 29,709 persons speak another language and are of Cuban origin. Using 0.5 for the all-confidence metric, we have additional associations which include (as a sample):

- *not limited from working AND not prevented from working AND Hispanic origin is Cuban AND speaks another language AND no military service and*
- *not limited from working AND not prevented from working AND speaks another language AND no military service AND race is white AND immigrated to the United States.*

However, since the all-confidence metric is useful for finding dependencies in the data, a minimum value of 0.5 may be too low to produce interesting results.

8 Conclusion

In this paper, we presented three alternative interest measures for associations: any-confidence, all-confidence, and bond. We proved that the important downward closure property applies to both all-confidence and bond. We showed that downward closure does not hold for any-confidence. We also proved that if associations have a minimum all-confidence or minimum bond, then those associations will have a given lower bound on their minimum support and the rules produced from those associations will have a given lower bound on their minimum confidence as well. We described the algorithms that find all associations with a minimum all-confidence or minimum bond and presented some experimental results using both synthetic data and real-world census data. The performance results showed that the algorithm can find large itemsets efficiently.

Appendix

Here, we provide the basic properties that are used in the proofs of the lemmas and theorem.

Property 1. *If* $\mathcal{L}' \subset \mathcal{L}$, *then,* $|\{d \mid d \in \mathcal{D} \wedge \mathcal{L}' \subset d\}| \geq |\{d \mid d \in \mathcal{D} \wedge \mathcal{L} \subset d\}|$.

Proof. Let $\mathcal{L}' = \{a_1, a_2, \ldots, a_k\}$ and

$$\mathcal{L} = \{a_1, a_2, \ldots, a_k, a_{k+1}, \ldots, a_n\}.$$

A transaction that contains the set of items $\mathcal{L}$ must obviously contain items in $\mathcal{L}'$. So, $|\{d \mid d \in \mathcal{D} \wedge \mathcal{L} \subset d\}|$ cannot be greater than $|\{d \mid d \in \mathcal{D} \wedge \mathcal{L}' \subset d\}|$. If all transactions that contain the set of items $\mathcal{L}'$ also contain the set of items $\{a_k, a_{k+1}, \ldots, a_n\}$ then,

$$|\{d \mid d \in \mathcal{D} \wedge \mathcal{L}' \subset d\}| = |\{d \mid d \in \mathcal{D} \wedge \mathcal{L} \subset d\}|.$$

If at least one transaction contains the set of items $\mathcal{L}'$ but not the set of items $\{a_k, a_{k+1}, \ldots, a_n\}$ then $|\{d \mid d \in \mathcal{D} \wedge \mathcal{L}' \subset d\}| \geq |\{d \mid d \in \mathcal{D} \wedge \mathcal{L} \subset d\}|$. Hence, $|\{d \mid d \in \mathcal{D} \wedge \mathcal{L}' \subset d\}| \geq |\{d \mid d \in \mathcal{D} \wedge \mathcal{L} \subset d\}|$. □

Property 2. *If* $\mathcal{L}' \subset \mathcal{L}$, *then*

$$\begin{aligned} &MAX\{i \mid \forall l(l \in \mathcal{P}(\mathcal{L}') \wedge l \neq \emptyset \wedge l \neq \mathcal{L}' \wedge i \\ &= |\{d \mid d \in \mathcal{D} \wedge l \subset d\}|)\} \\ &\leq MAX\{i \mid \forall l(l \in \mathcal{P}(\mathcal{L}) \wedge l \neq \emptyset \wedge l \neq \mathcal{L} \wedge i \\ &= |\{d \mid d \in \mathcal{D} \wedge l \subset d\}|)\}. \end{aligned}$$

Proof. Since $\mathcal{P}(\mathcal{L})$ contains all the members of $\mathcal{P}(\mathcal{L}')$, we have two cases:

1. A member of $\mathcal{P}(\mathcal{L}')$ is contained in the most transactions. Hence, we have equality between the left-hand and right-hand side of the expression.
2. If the member of $\mathcal{P}(\mathcal{L})$ which is not an element of $\mathcal{P}(\mathcal{L}')$ is contained in the most transactions, then, the left-hand side of the expression is less than the right hand side.

Hence,

$$\begin{aligned} &MAX\{i \mid \forall l(l \in \mathcal{P}(\mathcal{L}') \wedge l \neq \emptyset \wedge l \neq \mathcal{L}' \wedge i \\ &= |\{d \mid d \in \mathcal{D} \wedge l \subset d\}|)\} \\ &\leq MAX\{i \mid \forall l(l \in \mathcal{P}(\mathcal{L}) \wedge l \neq \emptyset \wedge l \neq \mathcal{L} \wedge i \\ &= |\{d \mid d \in \mathcal{D} \wedge l \subset d\}|)\}. \end{aligned}$$

□

Property 3. *If* $\mathcal{L}' \subset \mathcal{L}$, *then,*

$$\begin{aligned} &|\{d \mid d \in \mathcal{D} \wedge \exists l(l \in \mathcal{P}(\mathcal{L}') \wedge l \neq \emptyset \wedge l \subset d)\}| \\ &\leq |\{d \mid d \in \mathcal{D} \wedge \exists l(l \in \mathcal{P}(\mathcal{L}) \wedge l \neq \emptyset \wedge l \subset d)\}|. \end{aligned}$$

Proof. Since $\mathcal{P}(\mathcal{L})$ contains all members of $\mathcal{P}(\mathcal{L}')$, the number of transactions that contain members of $\mathcal{P}(\mathcal{L}')$ cannot be greater than the number of transactions that contain members of $\mathcal{P}(\mathcal{L})$. The left-hand side of the expression can be equal when transactions that contain any of the items in $\mathcal{L}'$ are the same transactions that contain any of the items in $\mathcal{L}$. The left-hand side can be less when there are transactions that contain any of the additional items in $\mathcal{L} - \mathcal{L}'$ and those transactions do not contain any of the items in $\mathcal{L}'$. Hence,

$$\begin{aligned} &|\{d \mid d \in \mathcal{D} \wedge \exists l(l \in \mathcal{P}(\mathcal{L}') \wedge l \neq \emptyset \wedge l \subset d)\}| \\ &\leq |\{d \mid d \in \mathcal{D} \wedge \exists l(l \in \mathcal{P}(\mathcal{L}) \wedge l \neq \emptyset \wedge l \subset d)\}|. \end{aligned}$$

□

Acknowledgments

This work was supported in part by Grant LM 06726 from the National Library of Medicine. The author would like to thank Carlos Ordonez for his comments on an earlier draft of this paper and would also like to thank the anonymous referees for their invaluable comments.

References

[1] C. Aggarwal and P. Yu, "Online Generation of Association Rules," *Proc. Int'l Conf. Data Eng.*, Feb. 1998.
[2] R. Agrawal, C. Aggarwal, and V. Prasad, "Depth-First Generation of Large Itemsets for Association Rules," *Proc. 2000 ACM Knowledge Discovery and Data Mining Conf.*, pp. 108-118, 2000.

[3] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules Between Sets of Items in Large Databases," *Proc. 1993 ACM SIGMOD Int'l Conf. Management of Data,* pp. 207-216, May 1993.
[4] R. Agrawal and J. Shafer, "Parallel Mining of Association Rules," *IEEE Trans. Knowledge and Data Eng.,* vol. 8, no. 6, pp. 962-969, Dec. 1996.
[5] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," *Proc. 20th Int'l Conf. Very Large Data Bases,* Aug. 1994.
[6] R. Bayardo and R. Agrawal, "Mining the Most Interesting Rules," *Proc. Knowledge Discovery and Data Mining Conf.,* pp. 145-154, Aug. 1999.
[7] S. Brin, R. Motwani, and C. Silverstein, "Beyond Market Baskets: Generalizing Association Rules to Correlations," *Proc. ACM SIGMOD Conf.,* pp. 265-276, May 1997.
[8] S. Brin, R. Motwani, J. Ullman, and S. Tsur, "Dynamic Itemset Counting and Implication Rules for Market Basket Data," *Proc. ACM SIGMOD Conf.,* pp. 255-264, May 1997.
[9] R. Cooley, P. Tan, and J. Srivastava, "Discovery of Interesting Usage Patterns from Web Data," *Proc. WEBKDD Workshop.,* 1999.
[10] E. Han, G. Karypis, and V. Kumar, "Scalable Parallel Data Mining for Association Rules," *Proc. 1997 ACM SIGMOD Int'l Conf. Management of Data,* pp. 277-288, May 1997.
[11] J. Han and Y. Fu, "Discovery of Multiple-Level Association Rules from Large Databases," *Proc. Very Large Databases Conf.,* pp. 420-431, Sept. 1995.
[12] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," *Proc. 2000 ACM SIGMOD Int'l Conf. Management of Data,* pp. 1-12, May 2000.
[13] C. Hidber, "Online Association Rule Mining," *Proc. ACM-SIGMOD Conf. Management of Data,* pp. 145-156, June 1999.
[14] M. Houtsma and A. Swami, "Set-Oriented Mining of Association Rules," *Proc. Int'l Conf. Data Eng.,* Mar. 1995.
[15] L. Lakshmanan, R. Ng, J. Han, and A. Pang, "Optimization of Constrained Frequent Set Queries with 2-Variable Constraints," *Proc. ACM-SIGMOD Conf. Management of Data,* pp. 157-168, June 1999.
[16] B. Liu, W. Hsu, and Y. Ma, "Mining Association Rules with Multiple Minimum Supports," *Proc. Knowledge Discovery and Data Mining Conf.,* pp. 337-341, Aug. 1999.
[17] H. Mannila, H. Toivonen, and A.I. Verkamo, "Efficient Algorithms for Discovering Association Rules," *Proc. Knowledge Discovery and Data Mining '94: AAAI Workshop Knowledge Discovery in Databases,* pp. 181-192, July 1994.
[18] Y. Morimoto, T. Fukuda, H. Matsuzawa, T. Tkuyama, and K. Yoda, "Algorithms for Mining Associations Rules for Binary Segmentation of Huge Categorical Databases," *Proc. Very Large Databases Conf.,* pp. 380-391, Sept. 1998.
[19] R. Ng, L. Lakshmanan, J. Han, and A. Pang, "Exploratory Mining and Pruning Optimizations of Constrained Associations Rules," *Proc. ACM-SIGMOD Conf. Management of Data,* pp. 13-24, June 1998.
[20] J.S. Park, M-S. Chen, and P.S. Yu, "An Effective Hash Based Algorithm for Mining Association Rules," *Proc. ACM-SIGMOD Conf. Management of Data,* pp. 229-248, May 1995.
[21] J. Pei and J. Han, "Can We Push More Constraints into Frequent Pattern Mining?" *Proc. 2000 ACM Knowledge Discovery and Data Mining Conf.,* 2000.
[22] *Knowledge Discovery in Databases,* G. Piatetsky-Shapiro and W.J. Frawley, eds. MIT Press, 1991.
[23] S. Ramaswamy, S. Mahajan, and A. Silbershatz, "On the Discovery of Interesting Patterns in Association Rules," *Proc. Very Large Databases Conf.,* pp. 368-379, Sept. 1998.
[24] G. Salton and M. McGill, *Introduction to Modern Information Retrieval.* New York: McGraw Hill, 1983.
[25] A. Savasere, E. Omiecinski, and S. Navathe, "An Efficient Algorithm for Mining Association Rules," *Proc. Very Large Databases Conf.,* pp. 432-444, Sept. 1995.
[26] A. Savasere, E. Omiecinski, and S. Navathe, "Mining for Strong Negative Associations in a Large Database of Customer Transactions," *Proc. IEEE Data Eng. Conf.,* Feb. 1998.
[27] R. Srikant and R. Agrawal, "Mining Generalized Association Rules," *Proc. Very Large Databases Conf.,* pp. 407-419, Sept. 1995.
[28] M. Zaki, "Generating Non-Redundant Association Rules," *Proc. 2000 ACM Knowledge Discovery and Data Mining Conf.,* pp. 34-43, 2000.

Edward R. Omiecinski received the PhD degree from Northwestern University in 1984. He is currently an associate professor at Georgia Tech in the College of Computing. He has published more than 50 papers in international journals and conferences dealing with database systems. His research has been funded by the US National Science Foundation, the Defense Advanced Research Projects Agency (DARPA), and the National Library of Medicine (NLM). His currently funded work deals with the discovery of knowledge in cardiac imagebases which is a collaborative effort between Georgia Tech and Emory University researchers. He is a member of the ACM and IEEE Computer Society.