Generating Non-Redundant Association Rules

Mohammed J. Zaki
Computer Science Department
Rensselaer Polytechnic Institute, Troy NY 12180
Phone: 518-276-6340, FAX: 518-276-4033, Email: {zaki}@cs.rpi.edu

Abstract

The traditional association rule mining framework produces many redundant rules. The extent of redundancy is a lot larger than previously suspected. We present a new framework for associations based on the novel concept of *closed* frequent itemsets. The number of non-redundant rules produced by the new approach is exponentially (in the length of the longest frequent itemset) smaller than the rule set from the traditional approach. Experiments using several "hard" real and synthetic databases confirm the utility of our framework in terms of reduction in the number of rules presented to the user, and in terms of time.

1 Introduction

Association rule discovery, a successful and important mining task, aims at uncovering all frequent patterns among transactions composed of data attributes or items. Results are presented in the form of rules between different sets of items, along with metrics like the joint and conditional probabilities of the antecedent and consequent, to judge a rule's importance.

It is widely recognized that the set of association rules can rapidly grow to be unwieldy, especially as we lower the frequency requirements. The larger the set of frequent itemsets the more the number of rules presented to the user, many of which are redundant. This is true even for sparse datasets, but for dense datasets it is simply not feasible to mine all possible frequent itemsets, let alone to generate rules between itemsets. In such datasets one typically finds an exponential number of frequent itemsets. For example, finding long itemsets of length 20 or 30 is not uncommon [2].

Prior research has mentioned that the traditional association rule mining framework produces too many rules, but the extent of redundancy is a lot larger than previously suspected. We present a new framework for association rule mining based on the novel concept of *closed* frequent itemsets. The set of all closed frequent itemsets can be orders of magnitude smaller than the set of all frequent itemsets, especially for real (dense) datasets. At the same time, we don't loose any information (which we would, if we were to use maximal frequent itemsets); the closed itemsets uniquely determine the set of all frequent itemsets. We show that the new framework produces exponentially (in the length of the longest frequent itemset) fewer rules than the traditional approach, again without loss of information. Our framework allows us to mine even dense datasets, where it is not possible to find all frequent itemsets. Experiments using several "hard" (i.e., dense) databases confirm the utility of our framework in terms of reduction in the number of rules presented to the user, and in terms of time.

The rest of the paper is organized as follows. Section 2 describes the association mining task. Section 3 introduces the new notion of closed itemsets. Section 4 looks at the problem of eliminating redundant rules. We discuss related work in Section 5. We experimentally validate out theoretical results in Section 6, and conclude in Section 7 (the Appendix contains proofs for all theorems presented in this paper, and it can be read at the discretion of the reviewer).

2 Association Rules

The association mining task can be stated as follows: Let $\mathcal{I} = \{1, 2, \cdots, m\}$ be a set of items, and let $\mathcal{T} = \{1, 2, \cdots, n\}$ be a set of transaction identifiers or *tids*. The input database is a binary relation $\delta \subseteq \mathcal{I} \times \mathcal{T}$. If an item i occurs in a transaction t, we write it as $(i, t) \in \delta$, or alternately as $i\delta t$. Typically the database is arranged as a set of transactions, where each transaction contains a set of items. For example, consider the database shown in Figure 1, used as a running example throughtout this paper. Here $\mathcal{I} = \{A, C, D, T, W\}$, and $\mathcal{T} = \{1, 2, 3, 4, 5, 6\}$. The second transaction can be represented as $\{C\delta 2, D\delta 2, W\delta 2\}$; all such pairs from all transactions, taken together form the binary relation δ.

A set $X \subseteq \mathcal{I}$ is called an *itemset*, and a set $Y \subseteq \mathcal{T}$ is called a *tidset*. For convenience we write an itemset $\{A, C, W\}$ as ACW, and a tidset $\{2, 4, 5\}$ as 245. The *support* of an itemset X, denoted $\sigma(X)$, is the number of transactions in which it occurs as a subset. An itemset is *frequent* if its support is more than or equal to a user-specified *minimum support (minsup)* value, i.e., if $\sigma(X) \geq$ *minsup*.

An *association rule* is an expression $A \xrightarrow{p} B$, where A and B are itemsets, and $A \cap B = \emptyset$. The *support* of the rule is given as $\sigma(A \cup B)$ (i.e., the joint probability of a transaction containing both A and B), and the *confidence* as $p = \sigma(A \cup B)/\sigma(A)$ (i.e., the conditional probability that a transaction contains B, given that it contains A). A rule is frequent if the itemset $A \cup B$ is frequent. A rule is *confident* if its confidence is greater than or equal to a user-specified *minimum confidence (minconf)* value, i.e, $p \geq minconf$.

DISTINCT DATABASE ITEMS

Jane Austen	Agatha Christie	Sir Arthur Conan Doyle	Mark Twain	P. G. Wodehouse
A	C	D	T	W

DATABASE

Transcation	Items
1	A C T W
2	C D W
3	A C T W
4	A C D W
5	A C D T W
6	C D T

ALL FREQUENT ITEMSETS

MINIMUM SUPPORT = 50%

Support	Itemsets
100% (6)	C
83% (5)	W, CW
67% (4)	A, D, T, AC, AW CD, CT, ACW
50% (3)	AT, DW, TW, ACT, ATW CDW, CTW, ACTW

Figure 1: Generating Frequent Itemsets

Association rule mining consists of two steps [1]: 1) Find all frequent itemsets, and 2) Generate high confidence rules.

Finding frequent itemsets This step computationally and I/O intensive. As a running example, consider Figure 1, which shows a bookstore database with six customers who buy books by different authors. It shows all the frequent itemsets with $minsup = 50\%$ (i.e., 3 transactions). $ACTW$ and CDW are the maximal-by-inclusion frequent itemsets (i.e., they are not a subset of any other frequent itemset).

Let $|\mathcal{I}| = m$ be the number of items. The search space for enumeration of all frequent itemsets is 2^m, which is exponential in m. One can prove that the problem of finding a frequent set of a certain size is NP-Complete, by reducing it to the balanced bipartite clique problem, which is known to be NP-Complete [7, 13]. However, if we assume that there is a bound on the transaction length, the task of finding all frequent itemsets is essentially linear in the database size, since the overall complexity in this case is given as $O(r \cdot n \cdot 2^l)$, where $|\mathcal{T}| = n$ is the number of transactions, l is the length of the longest frequent itemset, and r is the number of maximal frequent itemsets.

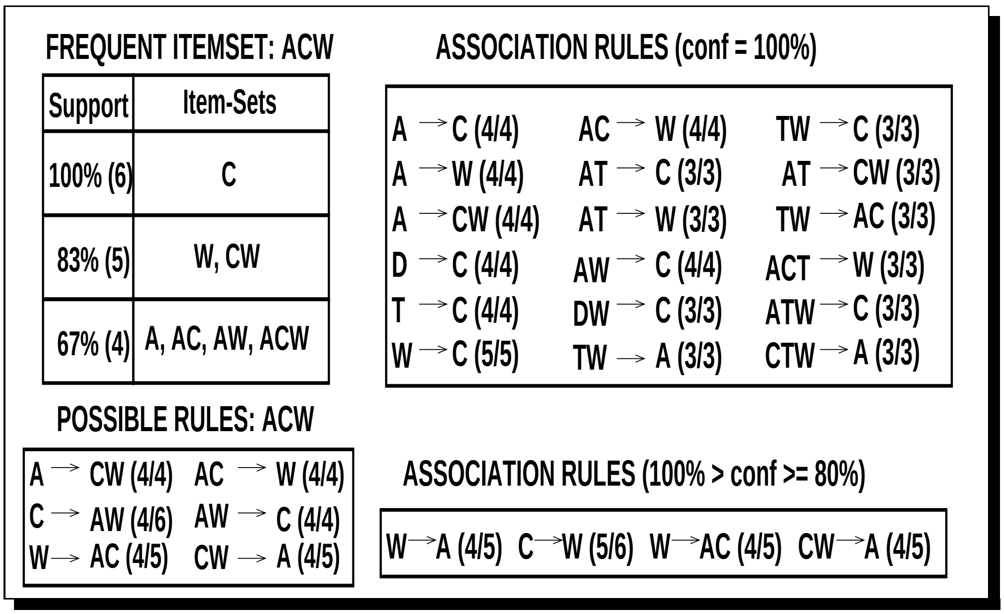

Figure 2: Generating Confident Rules

Generating confident rules This step is relatively straightforward; rules of the form $Y \xrightarrow{p} X - Y$, are generated for all frequent itemsets X, for all $Y \subset X$, $Y \neq \emptyset$, and provided $p \geq minconf$. For example, from the frequent itemset ACW we can generate 6 possible rules (all of them have support of 4): $A \xrightarrow{1.0} CW, C \xrightarrow{0.67} AW, W \xrightarrow{0.8} AC, AC \xrightarrow{1.0} W, AW \xrightarrow{1.0} C$, and $CW \xrightarrow{0.8} A$. This process is also shown pictorially in Figure 2. Notice that we need access to the support of all subsets of ACW to generate rules from it. To obtain all possible rules we need to examine each frequent itemset and repeat the rule generation process shown above for ACW. Figure 2 shows the set of all other association rules with confidence above or equal to *minconf* = 80%.

For an itemset of size k there are $2^k - 2$ potentially confident rules that can be generated. This follows from the fact that we must consider each subset of the itemset as an antecedent, except for the empty and the full itemset. The complexity of the rule generation step is thus $O(f \cdot 2^l)$, where f is the number of frequent itemsets, and l is the longest frequent itemset.

3 Closed Frequent Itemsets

In this section we develop the concept of closed frequent itemsets, and show that this set is necessary and sufficient to capture all the information about frequent itemsets, and has smaller cardinality than the set of all frequent itemsets.

3.1 Partial Order and Lattices

We first introduce some lattice theory concepts (see [5] for a good introduction).

Let P be a set. A *partial order* on P is a binary relation $\leq$, such that for all $x, y, z \in P$, the relation is: 1) Reflexive: $x \leq x$. 2) Anti-Symmetric: $x \leq y$ and $y \leq x$, implies $x = y$. 3) Transitive: $x \leq y$ and $y \leq z$, implies $x \leq z$. The set P with the relation $\leq$ is called an *ordered set*, and it is denoted as a pair $(P, \leq)$. We write $x < y$ if $x \leq y$ and $x \neq y$.

Let $(P, \leq)$ be an ordered set, and let S be a subset of P. An element $u \in P$ is an *upper bound* of S if $s \leq u$ for all $s \in S$. An element $l \in P$ is a *lower bound* of S if $s \geq l$ for all $s \in S$. The least upper bound is called the **join** of S, and is denoted as $\bigvee S$, and the greatest lower bound is called the **meet** of S, and is denoted as $\bigwedge S$. If $S = \{x, y\}$, we also write $x \vee y$ for the join, and $x \wedge y$ for the meet.

An ordered set $(L, \leq)$ is a *lattice*, if for any two elements x and y in L the join $x \vee y$ and meet $x \wedge y$ always exist. L is a *complete lattice* if $\bigvee S$ and $\bigwedge S$ exist for all $S \subseteq L$. Any finite lattice is complete. L is called a *join semilattice* if only the join exists. L is called a *meet semilattice* if only the meet exists.

Let $\mathcal{P}$ denote the power set of S (i.e., the set of all subsets of S). The ordered set $(\mathcal{P}(S), \subseteq)$ is a complete lattice, where the meet is given by set intersection, and the join is given by set union. For example, the partial orders $(\mathcal{P}(\mathcal{I}), \subseteq)$, the set of all possible itemsets, and $(\mathcal{P}(\mathcal{T}), \subseteq)$, the set of all possible tidsets are both complete lattices.

The set of all frequent itemsets, on the other hand, is only a meet-semilattice. For example, consider Figure 3, which shows the semilattice of all frequent itemsets we found in our example database (from Figure 1). For any two itemsets, only their meet is guaranteed to be frequent, while their join may or may not be frequent. This follows from the well known principle in association mining that, if an itemset is frequent, then all its subsets are also frequent. For example, $AC \wedge AT = AC \cap AT = A$ is frequent. For the join, while $AC \vee AT = AC \cup AT = ACT$ is frequent, $AC \cup DW = ACDW$ is not frequent.

3.2 Closed Itemsets

Let the binary relation $\delta \subseteq \mathcal{I} \times \mathcal{T}$ be the input database for association mining. Let $X \subseteq \mathcal{I}$, and $Y \subseteq \mathcal{T}$. Then the mappings

$$t : \mathcal{I} \mapsto \mathcal{T}, \quad t(X) = \{y \in \mathcal{T} \mid \ \forall x \in X, \ x\delta y\}$$
$$i : \mathcal{T} \mapsto \mathcal{I}, \quad i(Y) = \{x \in \mathcal{I} \mid \ \forall y \in Y, \ x\delta y\}$$

define a *Galois connection* between the partial orders $(\mathcal{P}(\mathcal{I}), \subseteq)$ and $(\mathcal{P}(\mathcal{T}), \subseteq)$, the power sets of $\mathcal{I}$ and $\mathcal{T}$, respectively. We denote a $X, t(X)$ pair as $X \times t(X)$, and a $i(Y), Y$ pair as $i(Y) \times Y$. Figure 4 illustrates the two mappings. The mapping $t(X)$ is the set of all transactions (tidset) which contain the itemset X, similarly $i(Y)$ is the itemset that is contained in all the transactions in Y. For example, $t(ACW) = 1345$, and $i(245) = CDW$. In terms of individual elements $t(X) = \bigcap_{x \in X} t(x)$, and $i(Y) = \bigcap_{y \in Y} i(y)$. For example $t(ACW) = t(A) \cap t(C) \cap t(W) = 1345 \cap 123456 \cap 12345 = 1345$.

The Galois connection satisfies the following properties (where $X, X_1, X_2 \in \mathcal{P}(\mathcal{I})$ and $Y, Y_1, Y_2 \in \mathcal{P}(\mathcal{T})$):

1) $X_1 \subseteq X_2 \Rightarrow t(X_1) \supseteq t(X_2)$. E.g., for $ACW \subseteq ACTW$, $t(ACW) = 1345 \supseteq 135 = t(ACTW)$.

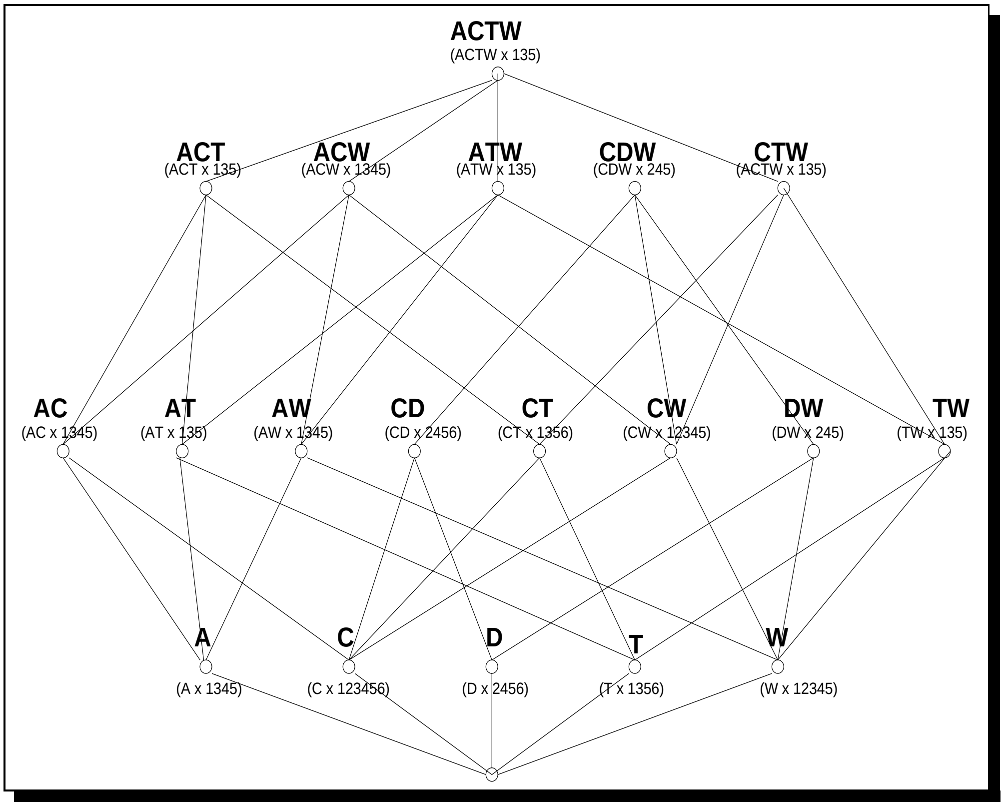

Figure 3: Meet Semi-lattice of Frequent Itemsets

2) $Y_1 \subseteq Y_2 \Rightarrow i(Y_1) \supseteq i(Y_2)$. For example, for $245 \subseteq 2456$, we have $i(245) = CDW \supseteq CD = i(2456)$.
3) $X \subseteq i(t(X))$ and $Y \subseteq t(i(Y))$. For example, $AC \subseteq i(t(AC)) = i(1345) = ACW$.

Let S be a set. A function $c : \mathcal{P}(S) \mapsto \mathcal{P}(S)$ is a *closure operator* on S if, for all $X, Y \subseteq S$, c satisfies the following properties: 1) Extension: $X \subseteq c(X)$. 2) Monotonicity: if $X \subseteq Y$, then $c(X) \subseteq c(Y)$. 3) Idempotency: $c(c(X)) = c(X)$. A subset X of S is called *closed* if $c(X) = X$.

Theorem 1 *Let $X \subseteq \mathcal{I}$ and $Y \subseteq \mathcal{T}$. Let $c_{it}(X)$ denote the composition of the two mappings $i \circ t(X) = i(t(X))$. Dually, let $c_{ti}(Y) = t \circ i(Y) = t(i(Y))$. Then $c_{it} : \mathcal{P}(\mathcal{I}) \mapsto \mathcal{P}(\mathcal{I})$ and $c_{ti} : \mathcal{P}(\mathcal{T}) \mapsto \mathcal{P}(\mathcal{T})$ are both closure operators on itemsets and tidsets respectively.*

We define a *closed itemset* as an itemset X that is that same as its closure, i.e., $X = c_{it}(X)$. For example the itemset ACW is closed. A *closed tidset* is a tidset $Y = c_{ti}(Y)$. For example, the tidset 1345 is closed.

The mappings c_{it} and c_{ti}, being closure operators, satisfy the three properties of extension, monotonicity, and idempotency. We also call the application of $i \circ t$ or $t \circ i$ a *round-trip*. Figure 4 illustrates this round-trip starting with an itemset X. For example, let $X = AC$, then the extension property say that X is a subset of its closure, since $c_{it}(AC) = i(t(AC)) = i(1345) = ACW$. Since $AC \neq c_{it}(AC) = ACW$, we conclude that AC is not closed. On the other hand, the idempotency property say that once we map an itemset to the tidset that contains it, and then map that tidset back to the set of items common to all tids in the tidset, we obtain a closed itemset. After this no matter how many such round-trips we make we cannot extend a closed itemset. For example, after one round-trip for AC we obtain the closed itemset ACW. If we perform another round-trip on ACW, we get $c_{it}(ACW) = i(t(ACW)) = i(1345) = ACW$.

For any closed itemset X, there exists a closed tidset given by Y, with the property that $Y = t(X)$ and $X = i(Y)$ (conversely, for any closed tidset there exists a closed itemset). We can see that X is closed by

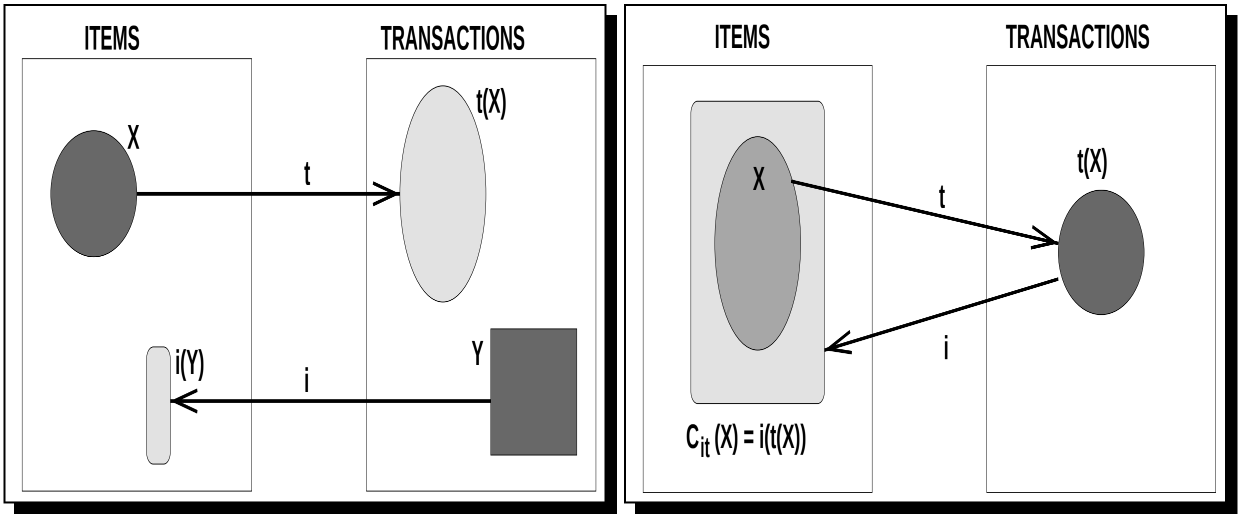

Figure 4: A) Galois Connection: Mappings between Items and Transactions, B) Closed Itemset: Round-Trip

the fact that $X = i(Y)$, then plugging $Y = t(X)$, we get $X = i(Y) = i(t(X)) = c_{it}(X)$, thus X is closed. Dually, Y is closed. For example, we have seen above that for the closed itemset ACW the associated closed tidset is 1345. Such a closed itemset and closed tidset pair $X \times Y$ is called a *concept*.

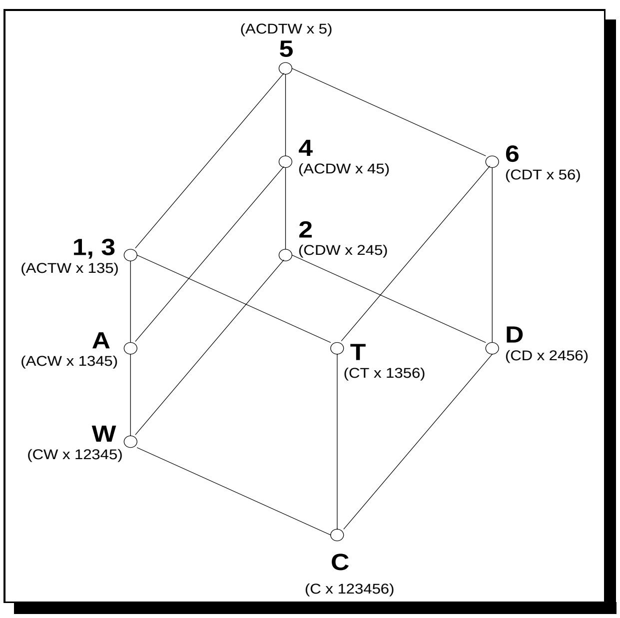

Figure 5: Galois Lattice of Concepts

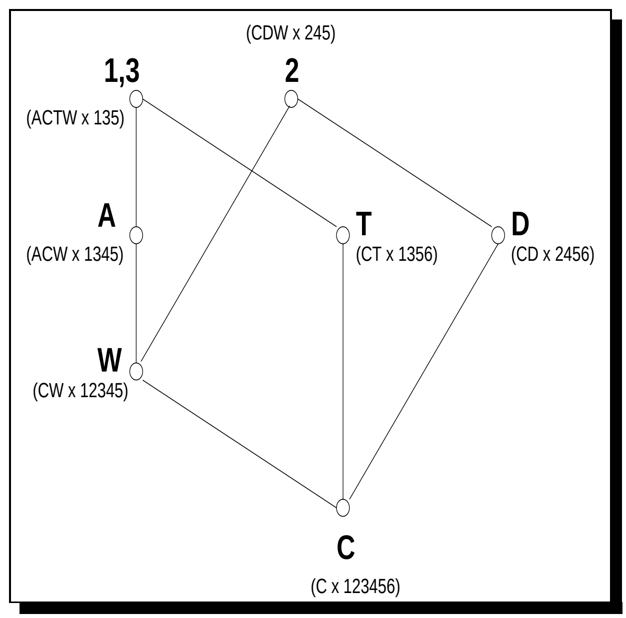

Figure 6: Frequent Concepts

A concept $X_1 \times Y_1$ is a *subconcept* of $X_2 \times Y_2$, denoted as $X_1 \times Y_1 \leq X_2 \times Y_2$, iff $X_1 \subseteq X_2$ (iff $Y_2 \subseteq Y_1$). Let $\mathcal{B}(\delta)$ denote the set of all possible concepts in the database. Then the ordered set $(\mathcal{B}(\delta), \leq)$ is a complete lattice, called the *Galois* lattice. For example, Figure 5 shows the Galois lattice for our example database, which has a total of 10 concepts. The least element is the concept $C \times 123456$ and the greatest element is the concept $ACDTW \times 5$. Notice that the mappings between the closed pairs of itemsets and tidsets are anti-isomorphic, i.e., concepts with large cardinality itemsets have small tidsets, and vice versa.

The concept generated by a single item $x \in \mathcal{I}$ is called an *item concept*, and is given as $\mathcal{C}_i(x) = c_{it}(x) \times t(x)$. Similarly, the concept generated by a single transaction $y \in \mathcal{T}$ is called a *tid concept*, and is given as $\mathcal{C}_t(y) = i(y) \times c_{ti}(y)$. For example, the item concept $\mathcal{C}_i(A) = i(t(A)) \times t(A) = i(1345) \times 1345 = ACW \times 1345$. Further, the tid concept $\mathcal{C}_t(2) = i(2) \times t(i(2)) = CDW \times t(CDW) = CDW \times 245$.

In Figure 5 if we relabel each concept with the item concept or tid concept that it is equivalent to, then we obtain a lattice with *minimal labelling*, with item or tid labels, as shown in the figure in bold letters. Such a relabelling reduces clutter in the lattice diagram, which provides an excellent way of visualizing the structure of the patterns and relationships that exist between items. We shall see its benefit in the next section when we talk about high confidence rules extraction.

It is easy to reconstruct the concepts from the minimal labeling. For example, consider the tid concept $\mathcal{C}_t(2) = X \times Y$. To obtain the closed itemset X, we append all item labels reachable below it. Conversely, to obtain the closed tidset Y we append all labels reachable above $\mathcal{C}_t(2)$. We see that W, D and C are all the labels reachable by a path below it. Thus $X = CDW$ forms the closed itemset. We also see that 4 and 5 are the only labels reachable above $\mathcal{C}_t(2)$. Thus $Y = 245$, giving the concept $CDW \times 245$, which matches the concept shown in the figure.

3.3 Frequent Closed Itemsets vs. Frequent Itemsets

We begin this section by defining the join and meet operation on the concept lattice (see [6] for the formal proof): The set of all concepts in the database relation δ, given by $(\mathcal{B}(\delta), \leq)$ is a (complete) lattice with join and meet given by

$$\textbf{join: } (X_1 \times Y_1) \ \vee \ (X_2 \times Y_2) = c_{it}(X_1 \cup X_2) \times (Y_1 \cap Y_2)$$
$$\textbf{meet: } (X_1 \times Y_1) \ \wedge \ (X_2 \times Y_2) = (X_1 \cap X_2) \times c_{ti}(Y_1 \cup Y_2)$$

For the join and meet of multiple concepts, we simply take the unions and joins over all of them. For example, consider the join of two concepts, $(ACDW \times 45) \vee (CDT \times 56) = c_{it}(ACDW \cup CDT) \times (45 \cap 56) = ACDTW \times 5$. On the other hand their meet is given as, $(ACDW \times 45) \wedge (CDT \times 56) = (ACDW \cap CDT) \times c_{ti}(45 \cup 56) = CD \times c_{ti}(456) = CD \times 2456$. Similarly, we can perform multiple concept joins or meets; for example, $(CT \times 1356) \vee (CD \times 2456) \vee (CDW \times 245) = c_{it}(CT \cup CD \cup CDW) \times (1356 \cap 2456 \cap 245) = c_{it}(CDTW) \times 5 = ACDTW \times 5$.

We define the support of a closed itemset X or a concept $X \times Y$ as the cardinality of the closed tidset $Y = t(X)$, i.e, $\sigma(X) = |Y| = |t(X)|$. A closed itemset or a concept is *frequent* if its support is at least *minsup*. Figure 6 shows all the frequent concepts with *minsup* = 50% (i.e., with tidset cardinality at least 3). The frequent concepts form a meet-semilattice, where the meet is guaranteed to exist, while the join may not.

All frequent itemsets can be determined by the join operation on the frequent item concepts in Figure 6. For example, since join of item concepts D and T, $\mathcal{C}_i(D) \vee \mathcal{C}_i(T)$, doesn't exist, DT is not frequent. On the other hand, $\mathcal{C}_i(A) \vee \mathcal{C}_i(T) = ACTW \times 135$, thus AT is frequent. Furthermore, the support of AT is given by the cardinality of the resulting concept's tidset, i.e., $\sigma(AT) = |t(AT)| = |135| = 3$.

Theorem 2 *For any itemset X, its support is equal to the support of its closure, i.e., $\sigma(X) = \sigma(c_{it}(X))$.*

This theorem states that all frequent itemsets are uniquely determined by the frequent closed itemsets (or frequent concepts). Furthermore, the set of frequent closed itemsets is bounded above by the set of frequent itemsets, and is typically much smaller, especially for dense datasets. For very sparse datasets, in the worst case, the two sets may be equal. To illustrate the benefits of closed itemset mining, contrast Figure 3, showing the set of all frequent itemsets, with Figure 6, showing the set of all closed frequent itemsets (or concepts). We see that while there are only 7 closed frequent itemsets, in contrast there are 19 frequent itemsets. This example clearly illustrates the benefits of mining the closed frequent itemsets.

4 Rule Generation

Recall that an association rule is of the form $X_1 \xrightarrow{p} X_2$, where $X_1, X_2 \subseteq \mathcal{I}$. Its support equals $|t(X_1 \cup X_2)|$, and its confidence is given as $p = P(X_2|X_1) = |t(X_1 \cup X_2)|/|t(X_1)|$. We are interested in finding all high support (at least *minsup*) and high confidence rules (at least *minconf*).

It is widely recognized that the set of such association rules can rapidly grow to be unwieldy. In this section we will show how the closed frequent itemsets help us form a generating set of rules, from which all other association rules can be inferred. Thus, only a small and easily understandable set of rules can be presented to the user, who can later selectively derive other rules of interest.

In the last section, we showed that the support of an itemset X equals the support of its closure $c_{it}(X)$. Thus it suffices to consider rules *only* among the frequent concepts. In other words the rule $X_1 \xrightarrow{p} X_2$ is exactly the same as the rule $c_{it}(X_1) \xrightarrow{p} c_{it}(X_2)$.

Another observation that follows from the concept lattice is that it is sufficient to consider rules among adjacent concepts, since other rules can be inferred by transitivity, that is:

Theorem 3 Transitivity: *Let X_1, X_2, X_3 be frequent closed itemsets, with $X_1 \subseteq X_2 \subseteq X_3$. If $X_1 \xrightarrow{p} X_2$ and $X_2 \xrightarrow{q} X_3$, then $X_1 \xrightarrow{pq} X_3$.*

In the discussion below, we consider two cases of association rules, those with 100% confidence, i.e., with $p = 1.0$, and those with $p < 1.0$.

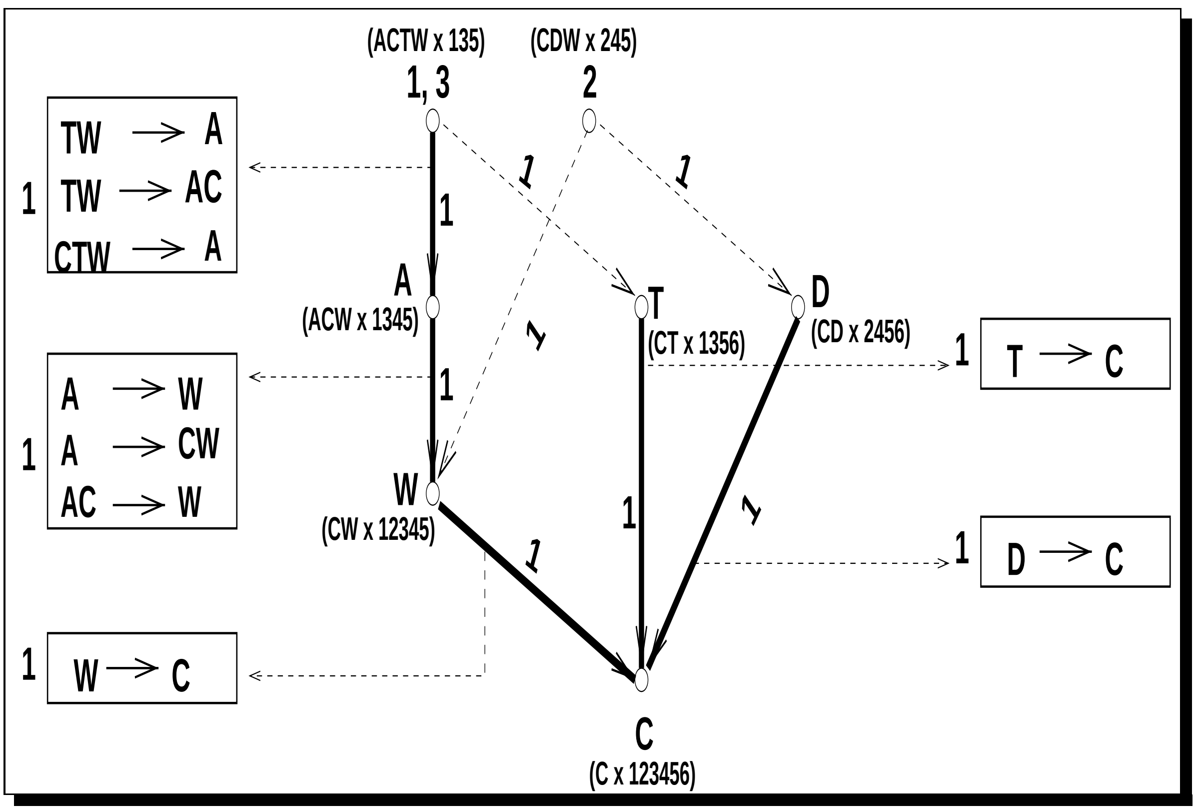

Figure 7: Rules with 100% Confidence

4.1 Rules with 100% Confidence

Theorem 4 *An association rule $X_1 \xrightarrow{1.0} X_2$ has confidence $p = 1.0$ if and only if $t(X_1) \subseteq t(X_2)$.*

This theorem says that all 100% confidence rules are those that are directed from a super-concept ($X_1 \times t(X_1)$) to a sub-concept ($X_2 \times t(X_2)$),i.e., down-arcs, since it is in precisely these cases that $t(X_1) \subseteq t(X_2)$ (or $X_1 \subseteq X_2$). For example, consider the item concepts $\mathcal{C}_i(W) = CW \times 12345$ and $\mathcal{C}_i(C) = C \times 123456$. The rule $W \xrightarrow{1.0} C$ is a 100% confidence rule. Note that if we take the itemset closure on both sides of the rule, we obtain $CW \xrightarrow{1.0} C$, i.e., a rule between closed itemsets, but since the antecedent and consequent are not disjoint in this case, we prefer to write the rule as $W \xrightarrow{1.0} C$, although both rules are exactly the same. Figure 7 shows some of the other rules among adjacent concepts with 100% confidence.

We notice that some down-arcs are labeled with more than one rule. In such cases, all rules within a box are equivalent, and we prefer the rule that is most general. For example, consider the rules $TW \xrightarrow{1.0} A, TW \xrightarrow{1.0} AC$, and $CTW \xrightarrow{1.0} A$. We prefer the rule $TW \xrightarrow{1.0} A$ since the latter two are obtained by adding one (or more) items to either the antecedent or consequent of $TW \xrightarrow{1.0} A$. In other words $TW \xrightarrow{1.0} A$ is more general than the latter two rules. In fact, we can say that the addition of C to either the antecedent or the consequent has no effect on the support or confidence of the rule. In this case we also call the other two rules redundant.

Theorem 5 *Let R_i stand for a 100% confidence rule $X_1^i \xrightarrow{1.0} X_2^i$, and let $\mathcal{R} = \{R_1, \cdots, R_n\}$ be a set of rules such that $I_1 = c_{it}(X_1^i \cup X_2^i)$, and $I_2 = c_{it}(X_2^i)$ for all rules R_i. Then all the rules are equivalent to the 100% confidence rule $I_1 \xrightarrow{1.0} I_2$, and thus are* redundant.

Let's apply this theorem to the three rules we considered above. We find that for the first rule that $c_{it}(TW \cup A) = c_{it}(ATW) = ACTW$. Similarly for the other two rules we see that $c_{it}(TW \cup AC) = c_{it}(ACTW) = ACTW$, and $c_{it}(CTW \cup A) = c_{it}(ACTW) = ACTW$. Thus for these three rules we get the closed itemset $I_1 = ACTW$. By the same process we obtain $I_2 = ACW$. All three rules correspond to the arc between the tid concept $\mathcal{C}_t(1,3)$ and the item concept $\mathcal{C}_i(A)$. Finally $TW \xrightarrow{1.0} A$ is the most general rule, and so the other two are redundant.

A set of such general rules constitutes a *generating set*, i.e., a rule set, from which all other 100% confidence rules can inferred. Note that in this paper we do not address the question of eliminating self-redundancy within this generating set, i.e., there may still exist rules in the generating set that can be derived from other rules in the set. In other words we do not claim anything about the minimality of the generating set; that is the topic of a forthcoming paper.

Figure 7 shows the generating set in bold arcs, which includes the 5 most general rules $\{TW \xrightarrow{1.0} A, A \xrightarrow{1.0} W, W \xrightarrow{1.0} C, T \xrightarrow{1.0} C, D \xrightarrow{1.0} C\}$ (the down-arcs that have been left out produce rules that cannot be written with disjoint antecedent and consequent. For example, between $\mathcal{C}_t(2)$ and $\mathcal{C}_i(D)$, the most general rule is $DW \xrightarrow{1.0} D$. Since the antecedent and consequent are not disjoint, as required by definition, we discard such rules). All other 100% confidence rules can be derived from this generating set by application of simple inference rules. For example, we can obtain the rule $A \xrightarrow{1.0} C$ by transitivity from the two rules $A \xrightarrow{1.0} W$ and $W \xrightarrow{1.0} C$. The rule $DW \xrightarrow{1.0} C$ can be obtained by augmentation of the two rules $W \xrightarrow{1.0} C$ and $D \xrightarrow{1.0} C$, etc. One can easily verify that all the 18 100% confidence rules produced by using frequent itemsets, as shown in Figure 2, can be generated from this set of 5 rules, produced using the closed frequent itemsets!

4.2 Rules with Confidence less than 100%

We now turn to the problem of finding a generating set for association rules with confidence less than 100%. As before, we need to consider only the rules between adjacent concepts. But this time the rules correspond to the up-arcs, instead of the down-arcs for the 100% confidence rules, i.e., the rules go from sub-concepts to super-concepts.

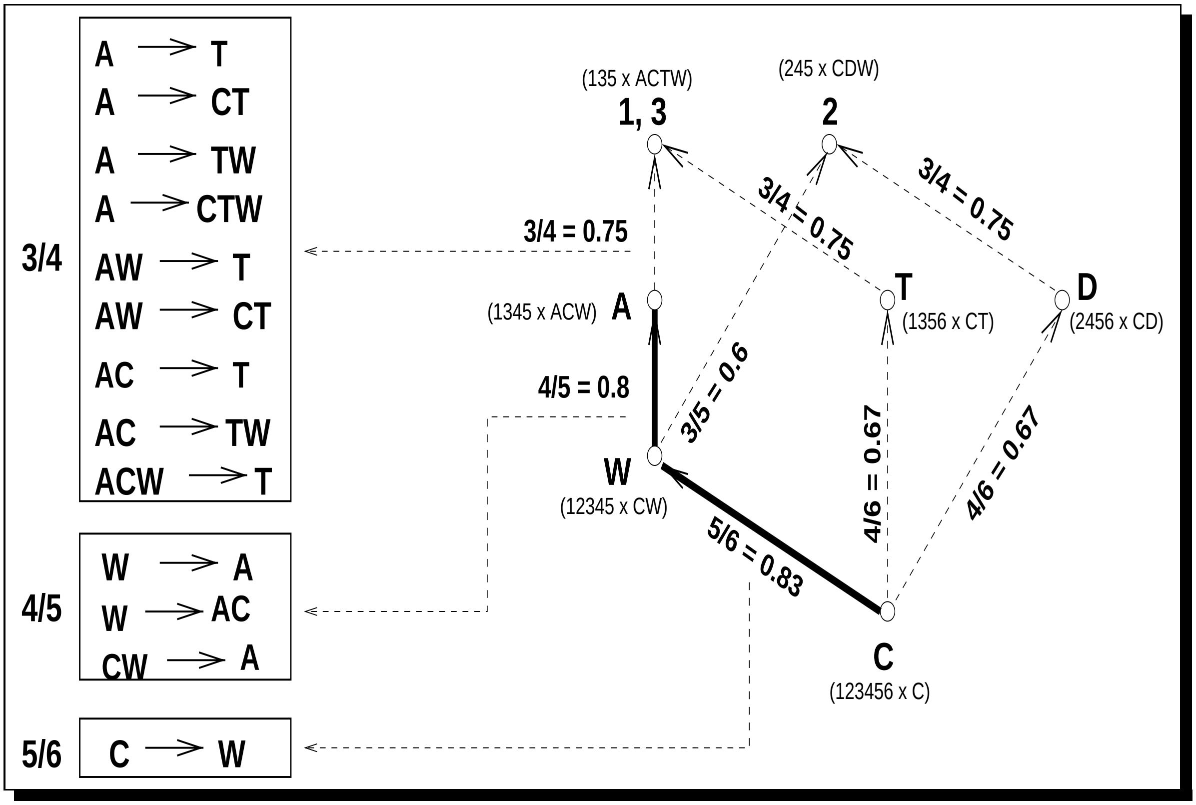

Figure 8: Rules with Confidence $< 100\%$

Consider Figure 8. The edge between item concepts $\mathcal{C}_i(C)$ and $\mathcal{C}_i(W)$ corresponds to $C \xrightarrow{0.83} W$. Rules between non-adjacent concepts can be derived by transitivity. For example, for $C \xrightarrow{p} A$ we can obtain the value of p using the rules $C \xrightarrow{q=5/6} W$ and $W \xrightarrow{r=4/5} A$. We have $p = qr = 5/6 \cdot 4/5 = 4/6 = 0.67$.

Theorem 6 *Let R_i stand for a $p < 1.0$ confidence rule $X_1^i \xrightarrow{p} X_2^i$, and let $\mathcal{R} = \{R_1, \cdots, R_n\}$ be a set of rules such that $I_1 = c_{it}(X_1^i)$, and $I_2 = c_{it}(X_1^i \cup X_2^i)$ for all rules R_i. Then all the rules are equivalent to the rule $I_1 \xrightarrow{p} I_2$, and thus are* redundant.

This theorem differs from that of the 100% confidence rules to account for the up-arcs. Consider the rules produced by the up-arc between item concepts $\mathcal{C}_i(W)$ and $\mathcal{C}_i(A)$. We find that for all three rules, $I_1 = c_{it}(W) = c_{it}(CW) = CW$, and $I_2 = c_{it}(W \cup A) = c_{it}(W \cup AC) = c_{it}(CW \cup A) = ACW$. The support of the rule is given by $|t(I_1 \cup I_2)| = |t(ACW)| = 4$, and the confidence given as $|t(I_1 \cup I_2)|/|t(I_1)| = 4/5 = 0.8$. Finally, since $W \xrightarrow{0.8} A$ is the most general rule, the other two are redundant. Similarly for the up-arc between $\mathcal{C}_i(A)$ and $\mathcal{C}_t(1,3)$, we get the general rule $A \xrightarrow{0.75} T$. The other 8 rules in the box are redundant!

The set of all such general rules forms a generating set of rules from which other rules can be inferred. The two bold arrows in Figure 8 constitute a generating set for all rules with $0.8 \leq p < 1.0$. Due to the transitivity property, we only have to consider arcs with confidence at least *minconf* = 0.8. No other rules can be confident at this level.

By combining the generating set for rules with $p = 1.0$, shown in Figure 7 and the generating set for rules with $1.0 > p \geq 0.8$, shown in Figure 8, we obtain a generating set for all association rules with *minsup* = 50%, and *minconf* = 80%: $\{TW \xrightarrow{1.0} A, A \xrightarrow{1.0} W, W \xrightarrow{1.0} C, T \xrightarrow{1.0} C, D \xrightarrow{1.0} C, W \xrightarrow{0.8} A, C \xrightarrow{0.83} W\}$.

It can be easily verified that all the association rules shown in Figure 2, for our example database from Figure 1, can be derived from this set. Using the closed itemset approach we produce 7 rules versus the 22 rules produced in traditional association mining. To see the contrast further, consider the set of all possible association rules we can mine. With *minsup* = 50%, the least value of confidence can be 50% (since the maximum support of an itemset can be 100%, but any frequent subset must have at least 50% support; the least confidence value is thus 50/100 = 0.5). There are 60 possible association rules versus only 13 in the generating set (5 rules with $p = 1.0$ in Figure 7, and 8 rules with $p < 1.0$ in Figure 8)

4.3 Complexity of Rule Generation: Traditional vs. Closed Itemset Framework

The complexity of rule generation in the traditional framework is $O(f \cdot 2^l)$, which is exponential in the length l of the longest frequent itemset (f is the total number of frequent itemsets). On the other hand using the closed itemset framework, the number of non-redundant rules is linear in the number of closed itemsets. To see how much savings are possible using closed frequent itemsets, lets consider the case where the longest frequent itemset has length l; with all 2^l subsets also being frequent.

In the traditional association rule framework, we would have to consider for each frequent itemset all its subsets as rule antecedents. The total number of rules generated in this approach is given as $\sum_{i=0}^{l} \binom{l}{i} \cdot 2^{l-i} \leq \sum_{i=0}^{l} \binom{l}{i} \cdot 2^l = 2^l \sum_{i=0}^{l} \binom{l}{i} = 2^l \cdot 2^l = 0(2^{2l})$.

On the other hand the number of non-redundant rules produced using closed itemsets is given as follows. Let's consider two extreme cases: In the best case, there is only one closed itemset, i.e., all 2^l subsets have the same support as the longest frequent itemset. Thus all rules between itemsets must have 100% confidence. The closed itemset approach doesn't produce any rule; it just lists the closed itemset with its frequency, with the implicit assumption that all possible rules from this itemset have 100% confidence. This corresponds to a reduction in the number of rules by a factor of $O(2^{2l})$.

On the other hand, in the worst case, all 2^l frequent itemsets are also closed. In this case there can be no 100% confidence rules and all ($<$ 100% confidence) rules point upwards, i.e., from subsets to their immediate supersets. For each subset of length k we have k rules from each of its $k - 1$ length subsets to that set. The total number of rules generated is thus $\sum_{i=0}^{l} \binom{l}{i} \cdot (l - i) \leq \sum_{i=0}^{l} \binom{l}{i} \cdot l = O(l \cdot 2^l)$. Thus we get a reduction in the number of rules by of a factor of $O(2^l/l)$, i.e., asymptotically exponential in the length of the longest frequent itemset.

5 Related Work

There has been a lot of research in developing efficient algorithms for mining frequent itemsets [1, 2, 4, 7, 8, 10, 14]. Most of these algorithms enumerate all frequent itemsets. Using these for rule generation produces many redundant rules. Some methods only generate maximal frequent itemsets [2, 8]. Maximal itemsets cannot be used for rule generation, since support of subsets is required for confidence computation. While it is easy to make one more data scan to gather the supports of all subsets, we still have the problem of many redundant rules. Further, for all these methods it is simply not possible to find rules in dense datasets which may easily have frequent itemsets of length 20 and more [2]. In contrast the set of *closed* frequent itemsets can be orders of magnitude smaller than the set of all frequent itemsets, and they can be used to generate rules even in dense domains. We use the recently proposed CHARM algorithm [12] for mining all closed frequent itemsets, in a fraction of the time it takes to mine all frequent itemsets using the Apriori [1] method.

There has been some work in pruning discovered association rules by forming rule covers [11]. However, the problem of constructing a generating set has not been studied previously. The recent work in [3] addresses the problem of mining the most interesting rules. They do not address the issue of rule redundancy, however their work is complimentary to ours.

A number of algorithms have been proposed for generating the Galois lattice of concepts [6]. These algorithms will have to be adapted to enumerate only the frequent concepts. Further, they have only been studied on small datasets. Our framework builds upon and adapts the work in [9]. However our characterization of the generating set is different, and we also present an experimental verification. An early version of this paper appeared in [13], but no experiments were shown.

6 Experimental Evaluation

All experiments described below were performed on a 400MHz Pentium PC with 256MB of memory, running RedHat Linux 6.0. Algorithms were coded in C++.

Database	# Items	Avg. Record Length	# Records
chess	76	37	3,196
connect	130	43	67,557
mushroom	120	23	8,124
pumsb*	7117	50	49,046
pumsb	7117	74	49,046
T20I12D100K	1000	20	100,000
T40I8D400K	1000	40	100,000

Table 1: Database Characteristics

Table 1 shows the characteristics of the real and synthetic datasets used in our evaluation. The real datasets were obtained from IBM Almaden (www.almaden.ibm.com/cs/quest/demos.html). All datasets except the PUMS (pumsb and pumsb*) sets, are taken from the UC Irvine Machine Learning Database Repository. The PUMS datasets contain census data. pumsb* is the same as pumsb without items with 80% or more support. The mushroom database contains characteristics of various species of mushrooms. Finally the connect and chess datasets are derived from their respective game steps. Typically, these real datasets are very dense, i.e., they produce many long frequent itemsets even for very high values of support.

We also chose a few synthetic datasets (also available from IBM Almaden), which have been used as benchmarks for testing previous association mining algorithms. These datasets mimic the transactions in a retailing environment. Usually the synthetic datasets are sparse when compared to the real sets, but we modified the generator to produce longer frequent itemsets.

6.1 Traditional vs. Closed Itemset Framework

Consider Tables 2, 4 and 3, which compares the traditional rule generation framework with the new closed itemset approach proposed in this paper. The tables shows the experimental results along a number of dimensions: 1) total number of frequent itemsets vs. closed frequent itemsets, 2) total number of rules in the

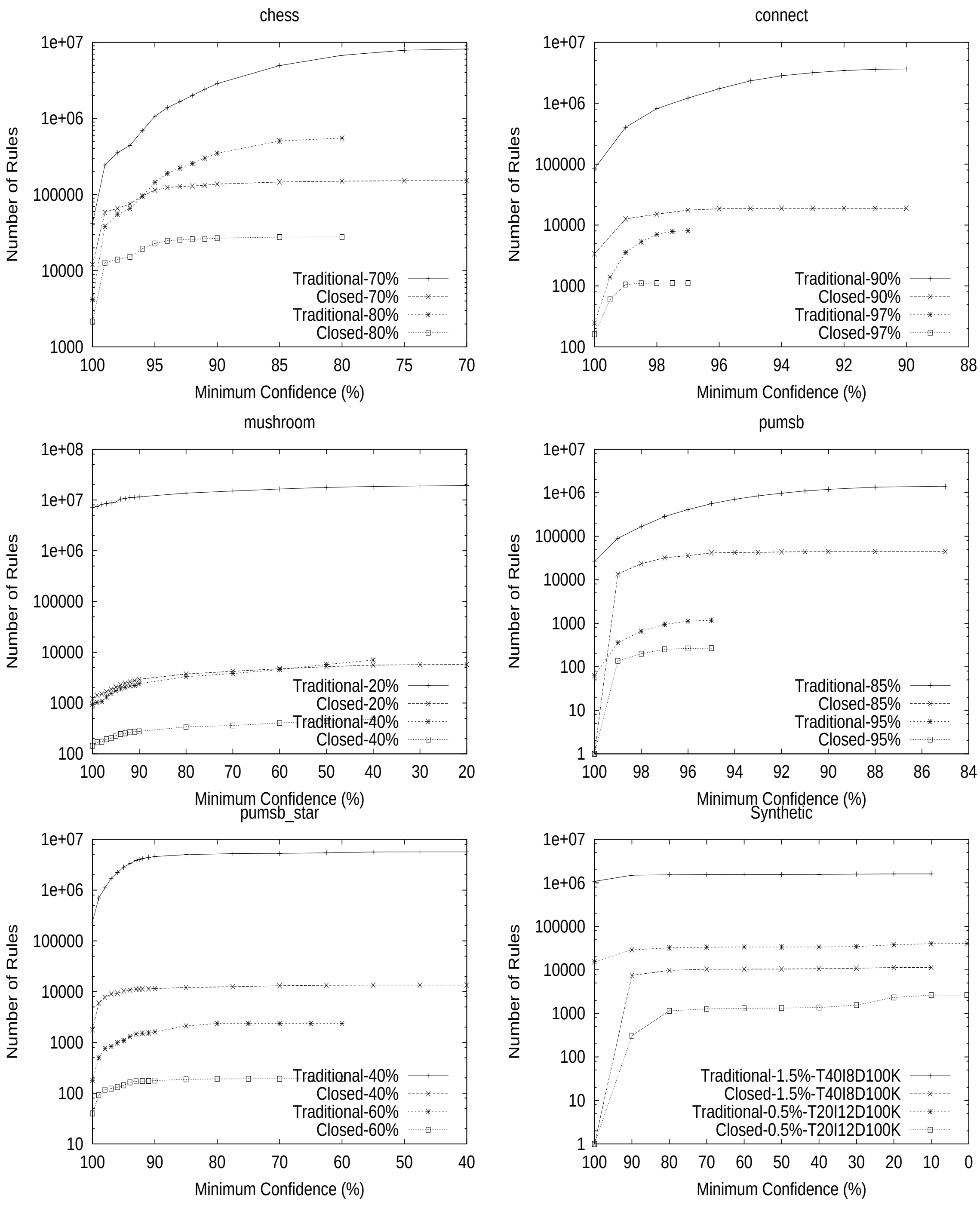

Figure 9: Number of Rules: Traditional vs. Closed Itemset Framework

Database	Sup	Len	Number of Itemsets #Freq	#Closed	Ratio
chess	80%	10	8227	5083	1.6
chess	70%	13	48969	23991	2.0
connect	97%	6	487	284	1.7
connect	90%	12	27127	3486	7.8
mushroom	40%	7	565	140	4.0
mushroom	20%	15	53583	1197	44.7
pumsb*	60%	7	167	68	2.5
pumsb*	40%	13	27354	2610	10.5
pumsb	95%	5	172	110	1.6
pumsb	85%	10	20533	8513	2.4
T20I12D100K	0.5%	9	2890	2067	1.4
T40I8D400K	1.5%	13	12088	4218	2.9

Table 2: Number of Itemset (Sup=*minsup*, Len=longest frequent itemset)

traditional vs. new approach, and 3) total time taken for mining all frequent itemsets (using Apriori) and the closed frequent itemsets (using CHARM).

Table 2 shows that the number of closed frequent itemsets can be much smaller than the set of all frequent itemsets. For the support values we look at here, we got reductions (shown in the Ratio column) in the cardinality anywhere from a factor of 1.4 to 44.7. For lower support values the gap widens rapidly [12]. It is noteworthy, that CHARM finds these closed sets in a fraction of the time it takes Apriori to mine all frequent itemsets as shown in Table 3. The reduction in running time ranges from a factor of 1.2 to more than 100 times (again the gap widens with lower support).

Table 4 shows that the reduction in the number of rules (with all possible consequent lengths) generated is drastic, ranging from a factor of 4 to more than 3000 times! Incidentally, these ratios are in agreement with the complexity formula we presented in Section 4.3. For example, consider the mushroom dataset. At 40% support, the longest frequent itemset has length 7. The complexity figure predicts a reduction in the number of rules by a factor of $2^7/7 = 128/7 = 18$, which is close to the ratio of 15 we got empirically. Similarly for 20% support, we expect a reduction of $2^{15}/15 = 2185$, and empirically it is 3343. The table also shows that even if we restrict the traditional rule generation to a single item consequent, the reduction with the closed itemset approach is still substantial, ranging from a factor of 2 to a factor of 66 reduction (once again, the reduction is more for lower supports).

The results above present all possible rules that are obtained by setting *minconf* equal to the *minsup*. Figure 9 shows the effect of *minconf* on the number of rules generated. It shows that a majority of the rules have very high confidence, a particularly distressing result for the traditional rule generation framework. The new approach produces a rule set that can be orders of magnitude smaller. In general it is possible to mine closed sets using CHARM for low values of support, where it is infeasible to find all frequent itemsets. Thus, even for dense datasets we can generate rules, which may not be possible in the traditional approach.

7 Conclusions

This paper has demonstrated in a formal way, supported with experiments on several datasets, the well known fact that the traditional association rule framework produces too many rules, most of which are

Database	Sup	Len	Running Time		
			Apriori	ChARM	Ratio
chess	80%	10	18.54	1.92	9.7
chess	70%	13	213.03	8.17	26.1
connect	97%	6	19.7	4.15	4.7
connect	90%	12	2084.3	43.8	47.6
mushroom	40%	7	1.56	0.28	5.6
mushroom	20%	15	167.5	1.2	144.4
pumsb*	60%	7	11.4	1.0	11.1
pumsb*	40%	13	847.9	17.1	49.6
pumsb	95%	5	19.7	1.7	11.7
pumsb	85%	10	1379.8	76.1	18.1
T20I12D100K	0.5%	9	6.3	5.1	1.2
T40I8D400K	1.5%	13	41.6	15.8	2.6

Table 3: Running Time (Sup=*minsup*, Len=longest frequent itemset)

redundant. We proposed a new framework based on closed itemsets that can drastically reduce the rule set, and that can be presented to the user in a succinct manner.

This paper opens a lot of interesting directions for future work. For example we plan to use the concept lattice for interactive visualization and exploration of a large set of mined associations. Keep in mind that the frequent concept lattice is a very concise representation of all the frequent itemsets and the rules that can be generated from them. Instead of generating all possible rules, we plan to generate the rules on-demand, based on the user's interests. Finally, there is the issue of developing a theory for extracting a base, or a minimal generating set, for all the rules.

References

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. Inkeri Verkamo. Fast discovery of association rules. In U. Fayyad and et al, editors, *Advances in Knowledge Discovery and Data Mining*, pages 307–328. AAAI Press, Menlo Park, CA, 1996.

[2] R. J. Bayardo. Efficiently mining long patterns from databases. In *ACM SIGMOD Conf. Management of Data*, June 1998.

[3] R. J. Bayardo and R. Agrawal. Mining the most interesting rules. In *5th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, August 1999.

[4] S. Brin, R. Motwani, J. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In *ACM SIGMOD Conf. Management of Data*, May 1997.

[5] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 1990.

[6] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag, 1999.

[7] D. Gunopulos, H. Mannila, and S. Saluja. Discovering all the most specific sentences by randomized algorithms. In *Intl. Conf. on Database Theory*, January 1997.

Database	Sup	Len	All Possible Rules #Traditional	#Closed	Ratio	Rules with one Consequent #Traditional	Ratio
chess	80%	10	552564	27711	20	44637	2
chess	70%	13	8171198	152074	54	318248	2
connect	97%	6	8092	1116	7	1846	1.7
connect	90%	12	3640704	18848	193	170067	9
mushroom	40%	7	7020	475	15	1906	4.0
mushroom	20%	15	19191656	5741	3343	380999	66
pumsb*	60%	7	2358	192	12	556	3
pumsb*	40%	13	5659536	13479	420	179638	13
pumsb	95%	5	1170	267	4	473	2
pumsb	85%	10	1408950	44483	32	113089	3
T20I12D100K	9	0.5%	40356	2642	15	6681	3
T40I8D400K	13	1.5%	1609678	11379	142	63622	6

Table 4: Number of Rules (all vs. rules with consequent of length 1) (Sup=*minsup*, Len=longest frequent itemset)

[8] D-I. Lin and Z. M. Kedem. Pincer-search: A new algorithm for discovering the maximum frequent set. In *6th Intl. Conf. Extending Database Technology*, March 1998.

[9] M. Luxenburger. Implications partielles dans un contexte. *Math. Inf. Sci. hum.*, 29(113):35–55, 1991.

[10] A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. In *21st VLDB Conf.*, 1995.

[11] H. Toivonen, M. Klemettinen, P. Ronkainen, K. Hätönen, and H. Mannila. Pruning and grouping discovered association rules. In *MLnet Wkshp. on Statistics, Machine Learning, and Discovery in Databases*, April 1995.

[12] M. J. Zaki and C.-J. Hsiao. CHARM: An efficient algorithm for closed association rule mining. Technical Report 99-10, Computer Science Dept., Rensselaer Polytechnic Institute, October 1999.

[13] M. J. Zaki and M. Ogihara. Theoretical foundations of association rules. In *3rd ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, June 1998.

[14] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New algorithms for fast discovery of association rules. In *3rd Intl. Conf. on Knowledge Discovery and Data Mining*, August 1997.

8 APPENDIX (Theorem Proofs)

Theorem 1 *Let* $X \subseteq \mathcal{I}$ *and* $Y \subseteq \mathcal{T}$. *Let* $c_{it}(X)$ *denote the composition of the two mappings* $i \circ t(X) = i(t(X))$. *Dually, let* $c_{ti}(Y) = t \circ i(Y) = t(i(Y))$. *Then* $c_{it} : \mathcal{P}(\mathcal{I}) \mapsto \mathcal{P}(\mathcal{I})$ *and* $c_{ti} : \mathcal{P}(\mathcal{T}) \mapsto \mathcal{P}(\mathcal{T})$ *are both closure operators on itemsets and tidsets respectively.*

PROOF: This is a well established result; see [6]. ■

Theorem 2 *For any itemset X, its support is equal to the support of its closure, i.e., $\sigma(X) = \sigma(c_{it}(X))$.*

PROOF: The support of an itemset X is the number of transactions where it appears, which is exactly the cardinality of the tidset $t(X)$, i.e., $\sigma(X) = |t(X)|$. Since $\sigma(c_{it}(X)) = |t(c_{it}(X))|$, to prove the theorem, we have to show that $t(X) = t(c_{it}(X))$.

Since c_{ti} is closure operator, it satisfies the extension property, i.e., $t(X) \subseteq c_{ti}(t(X)) = t(i(t(X))) = t(c_{it}(X))$. Thus $t(X) \subseteq t(c_{it}(X))$. On the other hand since c_{it} is also a closure operator, $X \subseteq c_{it}(X)$, which in turn implies that $t(X) \supseteq t(c_{it}(X))$, due to property 1) of Galois connections. Thus $t(X) = t(c_{it}(X))$. ■

Theorem 3 **Transitivity:** *Let X_1, X_2, X_3 be frequent closed itemsets, with $X_1 \subseteq X_2 \subseteq X_3$. If $X_1 \xrightarrow{p} X_2$ and $X_2 \xrightarrow{q} X_3$, then $X_1 \xrightarrow{pq} X_3$.*

PROOF: From the three rules we have $p = |t(X_1 \cup X_2)|/|t(X_1)|$, $q = |t(X_2 \cup X_3)|/|t(X_2)|$, and $pq = |t(X_1 \cup X_3)|/|t(X_1)|$. Since $X_1 \subseteq X_2$, we have $p = |t(X_2)|/|t(X_1)|$. Similarly, using other subset relationships, we get $q = |t(X_3)|/|t(X_2)|$, and $pq = |t(X_3)|/|t(X_1)|$. Now consider the product of the first two confidences, i.e., $p \cdot q = |t(X_2)|/|t(X_1)| \cdot |t(X_3)|/|t(X_2)| = |t(X_3)|/|t(X_1)|$, which matches the confidence of the third rule. ■

Theorem 4 *An association rule $X_1 \xrightarrow{1.0} X_2$ has confidence $p = 1.0$ if and only if $t(X_1) \subseteq t(X_2)$.*

PROOF: If $X_1 \xrightarrow{1.0} X_2$, it means that X_2 always occurs in a transaction, whenever X_1 occurs in that transaction. Put another way, the tidset where X_1 occurs must be a subset of the tidset where X_2 occurs. But this is precisely given as $t(X_1) \subseteq t(X_2)$.

The confidence of the rule $X_1 \xrightarrow{p} X_2$ is given as $p = |t(X_1 \cup X_2)|/|t(X_1) = |t(X_1) \cap t(X_2)|/|t(X_1)|$. If $t(X_1) \subseteq t(X_2)$, then $p = |t(X_1)|/|t(X_1)| = 1.0$. ■

Theorem 5 *Let R_i stand for a 100% confidence rule $X_1^i \xrightarrow{1.0} X_2^i$, and let $\mathcal{R} = \{R_1, \cdots, R_n\}$ be a set of rules such that $I_1 = c_{it}(X_1^i \cup X_2^i)$, and $I_2 = c_{it}(X_2^i)$ for all rules R_i. Then all the rules are equivalent to the 100% confidence rule $I_1 \xrightarrow{1.0} I_2$. Further, all rules other than the most general ones are* redundant.

PROOF: Consider any rule $R_i = X_1^i \xrightarrow{1.0} X_2^i$. Then the support of the rule is given as s and its confidence as s/r, where $s = |t(X_1^i \cup X_2^i)|$ and $r = |t(X_1^i)|$. Also according to Theorem 4 we have $t(X_1^i) \subseteq t(X_2^i)$. Then according to property 2 of Galois connections, we have $i(t(X_1^i)) \supseteq i(t(X_2^i))$, i.e, $c_{it}(X_1^i) \supseteq c_{it}(X_2^i)$.

Now consider the rule $I_1 \xrightarrow{1.0} I_2$. Its support is $|t(I_1 \cup I_2)| = |t(c_{it}(X_1^i \cup X_2^i) \cup c_{it}(X_2^i))| = |t(c_{it}(X_1^i \cup X_2^i))| = |t(X_1^i \cup X_2^i)| = s$. The last step follows from the fact that the support of an itemset equals the support of its closure.

Now we need to show that the denominator in the confidence formula equals r. The denominator is given as $|t(I_1)| = |t(c_{it}(X_1^i \cup X_2^i))| = |t(X_1^i \cup X_2^i)| = |t(X_1^i) \cap t(X_2^i)| = |t(X_1^i)| = r$. The last step follows from the fact that $t(X_1^i) \subseteq t(X_2^i)$. ■

Theorem 6 *Let R_i stand for a $p < 1.0$ confidence rule $X_1^i \xrightarrow{p} X_2^i$, and let $\mathcal{R} = \{R_1, \cdots, R_n\}$ be a set of rules such that $I_1 = c_{it}(X_1^i)$, and $I_2 = c_{it}(X_1^i \cup X_2^i)$ for all rules R_i. Then all the rules are equivalent to the rule $I_1 \xrightarrow{p} I_2$. Further, all rules other than the most general ones are* redundant.

PROOF: Consider any rule $R_i = X_1^i \xrightarrow{p} X_2^i$. Then the support of the rule is given as s and its confidence as $p = s/r$, where $s = |t(X_1^i \cup X_2^i)|$ and $r = |t(X_1^i)|$.

We will show that the $I_1 \xrightarrow{p} I_2$ also has confidence $p = s/r$. Let's consider the denominator first. We have $|t(I_1)| = |t(c_{it}(X_1^i))| = |t(X_1^i)| = r$.

Now consider the numerator. We have $|t(I_1 \cup I_2)| = |t(c_{it}(X_1^i) \cup c_{it}(X_1^i \cup X_2^i))|$. Since $X_1^i \subseteq (X_1^i \cup X_2^i)$, we have, from the property of closure operator, $c_{it}(X_1^i) \subseteq c_{it}(X_1^i \cup X_2^i)$. Thus, $|t(I_1 \cup I_2)| = |t(c_{it}(X_1^i \cup X_2^i))| = |t(X_1^i \cup X_2^i)| = s$. ∎