MINING ASSOCIATION RULES FROM MARKET BASKET DATA USING SHARE MEASURES AND CHARACTERIZED ITEMSETS

ROBERT J. HILDERMAN, COLIN L. CARTER[†],
HOWARD J. HAMILTON, and NICK CERCONE[‡]
Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada S4S 0A2
{hilder, hamilton}@cs.uregina.ca

[†] COLIN L. CARTER
Shaw Pipeline
Calgary, Alberta, Canada

[‡] NICK CERCONE
Department of Computer Science
Faculty of Mathematics
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
ncercone@math.uwaterloo.ca

We propose the *share-confidence framework* for knowledge discovery from databases which addresses the problem of mining characterized association rules from market basket data (i.e., itemsets). Our goal is to not only discover the buying patterns of customers, but also to discover customer profiles by partitioning customers into distinct classes. We present a new algorithm for classifying itemsets based upon characteristic attributes extracted from census or lifestyle data. Our algorithm combines the *Apriori* algorithm for discovering association rules between items in large databases, and the *AOG* algorithm for attribute-oriented generalization in large databases. We show how characterized itemsets can be generalized according to concept hierarchies associated with the characteristic attributes. Finally, we present experimental results that demonstrate the utility of the share-confidence framework.

Keywords: Data Mining, Knowledge Discovery, Machine Learning, Itemsets, Association Rules.

1 Introduction

The problem of mining association rules from market basket data has recently been an important research topic in the area of knowledge discovery from databases. It was originally introduced in [2] and studied extensively in [1, 5, 25, 26, 31, 19, 23, 29, 30, 3, 4, 33, 14]. The problem is typically examined in the context of discovering

buying patterns from retail sales transactions. Although there are many similar data mining applications which can be modelled in this way, we again study the problem using the retail store example because of its intuitive nature and clarity.

Consider a retail sales operation with a large inventory consisting of many different products. The operation is situated in a location where the customer base is socio-economically diverse, with annual household incomes ranging from very low to very high, and demographically ranging from young families to the elderly. The sales manager has used data mining to search for association rules in market basket data He has determined those products that are typically purchased together and those that are most likely to be purchased given that particular products have already been selected (called *itemsets*). Analysis of the itemsets has enabled him to strategically arrange store displays and plan advertising campaigns to increase sales. He now wonders whether there are any more subtle socio-economic buying patterns that could be helpful in guiding the distribution of flyers during the next advertising campaign. For example, he would like to know which itemsets are more likely to be purchased by those with higher incomes or by those with children. He would also like to know which itemsets are more likely to be purchased by those living in particular sales territories. He believes that characterizing itemsets with classificatory information available from credit card or cheque transactions will allow him to answer queries of this kind.

Using typical itemset methodologies, the sales manager is able to discover buying patterns through the generation of association rules which result in statements such as "90% of transactions that purchase bread also purchase butter." The sales manager's new goal is two-fold: to discover buying patterns which more accurately reflect the financial implications of an itemset, and to develop a profile of the purchasers of the itemset. For example, he wants the ability to generate association rules which result in statements such as "The purchase of the bread and butter itemset comprises a 40% share of the quantity of all items sold." He then wants the ability to characterize the previous statement with a qualifier such as "65% of the bread and butter itemset purchases are by customers in eastern areas, 25% are by customers in southern areas, and 10% are by customers in northern and western areas."

In this paper, we propose the *share-confidence framework* for association rules that looks beyond the simple frequency with which two or more items are bought together. Our framework addresses functionality and versatility issues of market basket analysis by providing share measures which more accurately indicate the financial impact of an itemset. We improve versatility by not only considering the co-occurrence of items in a market basket, but by also considering the quantity and value of the items purchased. We extend functionality with a new algorithm that characterizes itemsets with classificatory information extracted from external databases (i.e., customer, census, or lifestyle data). The algorithm, called *CI*, integrates the *Apriori* algorithm for discovering association rules between items in large databases [5, 31, 3, 4], and the *AOG* algorithm for attribute-oriented generalization

in large databases [9, 18, 15, 12, 13, 11, 20, 10]. We show how association rules can be mined from market basket data by using share measures and characterized itemsets which have been generalized according to concept hierarchies associated with characteristic attributes. However, it should be noted that our methods are not limited to the discovery of customer profiles based upon market basket data, the method is more widely applicable to any problem where taxonomic hierarchies can be associated with characterized data.

The remainder of this paper is organized as follows. In the following section, we present a formal description of the market basket analysis problem and describe a well-known itemset generation algorithm. In Section 3, we introduce the share-confidence framework for association rules. In Section 4, we compare share, the primary metric in the share-confidence framework, with support, the primary metric in the support-confidence framework. In Section 5, we describe characterized itemsets and an algorithm for generating characterized itemsets from market basket data. In Section 6, we review attribute-oriented generalization and show how this summarization technique can be useful in a characterized itemset application. In Section 7, we combine all of the techniques discussed to discover association rules in an extended example. In Section 8, we present experimental results obtained using the share-confidence framework on a database supplied by a commercial partner. We conclude in Section 9 with a summary of this work and suggest areas for future research.

2 Background

We now provide background information by reviewing significant previous work on association rules and the market basket analysis problem.

This section is organized as follows. In Section 2.1, we present a formal description of the market basket analysis problem. In Section 2.2, we describe *Apriori*, one of the most well-known algorithms for finding itemsets from market basket data. In Section 2.3, we present an example to demonstrate the operation of *Apriori*.

2.1 Statement of Problem

The problem of discovering association rules form market basket data has been formally defined as follows [2, 5, 3]. Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of literals, called *items*. Let D be a set of *transactions*, where each transaction T is an *itemset* such that $T \subseteq I$. Transaction T contains X, a set of some items in I, if $X \subseteq T$. An *association rule* is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. The association rule $X \Rightarrow Y$ holds in transaction set D with *confidence* c, if $c\%$ of transactions in D that contain X, also contain Y. The association rule $X \Rightarrow Y$ has *support* s in transaction set D, if $s\%$ of transactions in D contain $X \cup Y$. This formalism is known as the *support-confidence framework* for association rules [7].

There is a subtle feature of the support-confidence framework that should be

recognized before we proceed with our discussion. That is, that the implication symbol (i.e., $\Rightarrow$) used in association rules does not correspond to the logical notion of implication. Instead, the confidence of an association rule $X \Rightarrow Y$ actually measures the conditional probability of Y given X, denoted $P(Y \mid X)$. However, we will use the implication symbol in our discussion to remain consistent with previous work in this area.

To demonstrate the support and confidence measures, consider the transaction database containing eleven transactions, shown in Table 1. In Table 1, the *TID* column describes the transaction identifier and columns A to F describe the items (products) being sold. The values in columns A to F are binary, where a 1 indicates that at least one of the corresponding item has been purchased in the transaction and a 0 indicates the item was not purchased. For example, in transaction T_1, items A, C, and D have been purchased, while items B, E, and F have not.

Table 1: An example transaction database

TID	*A*	*B*	*C*	*D*	*E*	*F*
T_1	1	0	1	1	0	0
T_2	0	1	0	0	1	0
T_3	1	1	1	0	0	0
T_4	0	0	1	0	1	1
T_5	0	0	0	1	0	1
T_6	0	1	1	0	1	0
T_7	1	0	0	1	1	1
T_8	1	0	0	1	0	1
T_9	0	1	0	0	1	1
T_{10}	0	1	1	0	1	0
T_{11}	0	0	1	0	0	1

We acknowledge that the model used to represent the transaction database in Table 1 is inefficient in its use of space. In a real-world business application, a typical transaction contains only a few of many possible items being sold. So using this model, storage space would be wasted to record items that were not sold (e.g. consider the space that would be wasted for a retailer with an inventory of 25,000 unique products). However, for demonstration purposes, giving each item its own column in each transaction enables the table to be scanned quickly. Thus, the format shown was adopted for reader convenience.

The support for some of the possible itemsets that can be generated from Table 1 are shown in Table 2 (not an exhaustive list). In Table 2, the *Itemset* column describes the items in the itemset, the *TIDs* column describes the transaction identifiers that contain the corresponding itemset, the *No. of Transactions* column describes the number of transactions contained in the corresponding itemset, and the *Support* column describes the calculated support for the corresponding itemset. For example, the 1-itemset $\{B\}$ has 45% support because the sale of item B occurred in 5 of the 11 transactions, the 2-itemset $\{B, D\}$ has no support because the sale of items B and D did not occur together in any transactions, and the 3-itemset $\{B, C, E\}$ has 18% support because the sale of items B, C, and E occurred together in 2 of the 11 transactions.

The confidence for some of the possible association rules that can be generated from Table 1 are shown in Table 3 (again, not an exhaustive list). In Table 3, the

Table 2: Support for some of the possible itemsets

Itemset	*TIDs*	*No. of Transactions*	*Support (%)*
$\{A\}$	T_1, T_3, T_7, T_8	4	36
$\{B\}$	$T_2, T_3, T_6, T_9, T_{10}$	5	45
$\{E\}$	$T_2, T_4, T_6, T_7, T_9, T_{10}$	6	55
$\{A, C\}$	T_1, T_3	2	18
$\{A, B\}$	T_3	1	9
$\{B, D\}$		0	0
$\{C, E\}$	T_4, T_6, T_{10}	3	27
$\{A, B, C\}$	T_3	1	9
$\{A, B, E\}$		0	0
$\{B, C, E\}$	T_6, T_{10}	2	18
$\{A, D, E, F\}$	T_7	1	9

Association Rule column describes the association rules, the *No. of Transactions* (X) column describes the number of transactions that contain itemset X, the *No. of Transactions* $(X \cup Y)$ column describes the number of transactions that contain itemset $X \cup Y$, and the *Confidence* column describes the confidence in association rule $X \Rightarrow Y$. For example, the association rule $\{A\} \Rightarrow \{C\}$ has 50% confidence because item C occurs in two of the four transactions containing item A, $\{B\} \Rightarrow \{D\}$ has 0% confidence because item D occurs in none of the five transactions containing item B, and $\{A, B\} \Rightarrow \{C\}$ has 100% confidence because item C occurs in all of the transactions containing $\{A, B\}$.

Table 3: Confidence for some of the possible association rules

Association Rule $(X \Rightarrow Y)$	*No. of Transactions* (X)	*No. of Transactions* $(X \cup Y)$	*Confidence* (%)
$\{A\} \Rightarrow \{C\}$	4	2	50
$\{A\} \Rightarrow \{B\}$	4	1	25
$\{B\} \Rightarrow \{D\}$	5	0	0
$\{C\} \Rightarrow \{E\}$	6	3	50
$\{A, B\} \Rightarrow \{C\}$	1	1	100
$\{A, B\} \Rightarrow \{E\}$	1	0	0
$\{B\} \Rightarrow \{C, E\}$	5	2	40
$\{A, D\} \Rightarrow \{E, F\}$	3	1	33

2.2 The Apriori Algorithm

The most studied and analyzed algorithm for generating itemsets in the support-confidence framework is *Apriori* [5, 31, 29, 3, 4, 8]. This algorithm extracts the set of frequent itemsets from the set of candidate itemsets generated. A *frequent itemset* is an itemset whose support is greater than some user-specified minimum and a *candidate itemset* is an itemset whose support has yet to be determined.

Apriori is a level-wise algorithm that combines the frequent itemsets from pass $k-1$ to create the candidate itemsets in pass k. It has the important property that if any subset of a candidate itemset is not a frequent itemset, then the candidate itemset is also not a frequent itemset. In the overview of *Apriori* that follows, let L_k and C_k be the set of frequent and candidate k-itemsets, respectively. The k-th pass of the algorithm works as follows (assume the items in each itemset are in lexicographic order).

1. Generate the candidate itemsets in C_k from the frequent itemsets in L_{k-1}.

a. Join L_{k-1} with L_{k-1} (using the method shown below in SQL pseudo-code).

SELECT $p.\text{item}_1$, $p.\text{item}_2$, ..., $p.\text{item}_{k-1}$, $q.\text{item}_{k-1}$
FROM L_{k-1} p, L_{k-1} q
WHERE $p.\text{item}_1 = q.\text{item}_1$, $p.\text{item}_2 = q.\text{item}_2$, ..., $p.\text{item}_{k-2} = q.\text{item}_{k-2}$, $p.\text{item}_{k-1} = q.\text{item}_{k-1}$

b. Generate all $(k-1)$-subsets from the candidate itemsets in C_k.

c. Prune all candidate itemsets from C_k where some $(k-1)$-subset of the candidate itemset is not in the frequent itemset L_{k-1}.

2. Scan the transaction database to determine the support for each candidate itemset in C_k.

3. Save the frequent itemsets in L_k.

The first pass of the algorithm is a special pass which determines the frequent 1-itemsets, as follows.

1. Generate the candidate itemsets in C_1.

2. Save the frequent itemsets in L_1.

2.3 Example

We now present an example to demonstrate *Apriori*. Assume we are given the transaction database shown in Table 4, and the frequent itemsets contained in L_3, shown in Table 5. Also, assume the user-specified minimum support is 35%. In Tables 4 and 5, the column descriptions have the same meaning as the like-named columns in Tables 1 and 2. Our task is to generate L_4.

Table 4: A smaller example transaction database

TID	*A*	*B*	*C*	*D*	*E*
T_1	1	1	1	0	0
T_2	1	1	1	1	1
T_3	1	0	1	1	0
T_4	1	0	1	1	1
T_5	1	1	1	1	0

Table 5: Frequent itemsets contained in L_3

Itemset	*Support* (%)
$\{A, B, C\}$	60
$\{A, B, D\}$	40
$\{A, C, D\}$	80
$\{A, C, E\}$	40
$\{A, D, E\}$	40
$\{B, C, D\}$	40
$\{C, D, E\}$	40

Following *Apriori* for the k-th pass, we join L_3 with L_3 to generate the candidate itemsets in C_4. Joining $\{A, B, C\}$ with $\{A, B, D\}$ yields $\{A, B, C, D\}$, and

joining $\{A, C, D\}$ with $\{A, C, E\}$ yields $\{A, C, D, E\}$. So before pruning, $C_4 = \langle\{A, B, C, D\}, -\rangle, \langle\{A, C, D, E\}, -\rangle$, where each element is a two-tuple containing the candidate itemset and its support (the $-$ symbol means that the support has yet to be determined). We then generate the 3-subsets from the candidate itemsets in C_4. The 3-subsets of $\{A, B, C, D\}$ are $\{A, B, C\}$, $\{A, B, D\}$, $\{A, C, D\}$, and $\{B, C, D\}$, and the 3-subsets of $\{A, C, D, E\}$ are $\{A, C, D\}$, $\{A, C, E\}$, $\{A, D, E\}$, and $\{C, D, E\}$. Since $\{A, D, E\}$ and $\{C, D, E\}$ (subsets of $\{A, C, D, E\}$) are not frequent itemsets in L_3, we prune $\{A, C, D, E\}$ from the candidate itemsets in C_4, yielding $C_4 = \langle\{A, B, C, D\}, -\rangle$. Finally, we scan the transaction database to determine the support for the remaining candidate itemset in C_4, and determine that the support is 40%, yielding $C_4 = \langle\{A, B, C, D\}, 40\%\rangle$. Since the support for the remaining candidate itemset $\{A, B, C, D\}$ is greater than 35%, it is a frequent itemset, so we save it in L_4, yielding $L_4 = \langle\{A, B, C, D\}, 40\%\rangle$.

3 The Share-Confidence Framework

In the support-confidence framework, the purchase of an item is indicated by a binary flag (i.e., the item is either purchased or not purchased). From this binary flag, we can determine the number of transactions containing an itemset, but not the number of items in the itemset. If we knew the number of items, we may find that an itemset is actually more frequent than support indicates, allowing for more accurate financial analysis, comparisons, and projections. We will now extend the formalization of the market basket problem from Section 2.1, enabling us to quantify the effect of selling more than one of the same item in a single transaction. The problem definition is identical to that for the support-confidence framework, except that we introduce the notion of share for itemsets, and redefine the notions of frequent itemsets and confidence. We refer to this extended formalism as the *share-confidence framework* for association rules and refer to the new itemset measures as simply *share measures*. In this framework, any of the algorithms presented in [2, 3, 16, 19, 22, 23, 29, 30, 31, 32, 33] can used to generate frequent itemsets using our new definition for frequent itemset. The definitions in this section have been implemented in a data mining system for analyzing market basket data. This system is an extension of *DB-Discover*, a software tool for knowledge discovery form databases [13, 11, 10].

This section is organized as follows. In Section 3.1, we discuss the limitations of support in the support-confidence framework. In Sections 3.2 and 3.3, we define functions that form the basis of the share-confidence framework. In Section 3.4, we introduce and define the notion of share. In Sections 3.5 and 3.6, we provide new definitions for the notions of frequent itemsets and confidence, respectively. In these sections, we present a two-part example following each definition. The first part is a natural language query demonstrating the practical application and utility of each definition. The second part is the result of the query. For these examples, refer to the transaction database shown in Table 6 and the item database shown in Table 7. Table 6 is identical to Table 1 except that the binary values have been replaced

with the actual number of items purchased in the corresponding transaction (i.e., the counts). In Table 7, the *Item* column describes the valid items and the *Retail Price* column describes the retailer's selling price to the customer.

Table 6: An example transaction database with counts

TID	*A*	*B*	*C*	*D*	*E*	*F*
T_1	1	0	2	2	0	0
T_2	0	3	0	0	1	0
T_3	4	1	2	0	0	0
T_4	0	0	3	0	1	2
T_5	0	0	0	4	0	1
T_6	0	3	2	0	1	0
T_7	3	0	0	1	2	4
T_8	2	0	0	4	0	2
T_9	0	1	0	0	2	1
T_{10}	0	4	1	0	1	0
T_{11}	0	0	3	0	0	2
	10	12	13	11	8	12

Table 7: The item database

Item	*Retail Price*
A	1.50
B	2.25
C	5.00
D	4.75
E	10.00
F	7.50

3.1 Motivation

Support in the support-confidence framework is the foundation measure for determining the interest of itemsets. It provides a stable method for comparing itemsets since the support of an itemset is relative to the number of transactions in the database. However, the notion of support has a primary limitation in that it assumes the number of items purchased in an itemset is irrelevant to meaningful analysis. If we knew the number of items, we may find that the sale of a particular combination of items is actually more frequent than support indicates, allowing for more accurate financial analysis, comparisions, and projections. For example, some items in a grocery store are typically purchased in multiples, such as frozen concentrated juices. If we assume that we are searching for itemsets whose support is above 2%, say, we may find that the itemset containing frozen concentrated juice has only 1.5% support, and consequently is considered infrequent. But if we were to count the number of cans purchased, we might find that frozen concentrated juice actually contributes a higher percentage of sales than 2%. To our sales manager, this could be interesting.

Support also does not allow for accurate financial calculations or comparisons. In [27], it is suggested that analysis of itemsets should consider both the frequency of an item contributing to a predictive rule and the value of the items in the prediction. Support allows for neither of these elements in its analysis, so its use is limited as a practical indicator for determining the effect an itemset has on gross sales, cost, or net profit. For example, an item with 2% support on which a 15 cent profit

is earned for each item sold, is not as interesting as an item with 2% support on which a 25 cent profit is earned for each item sold, unless the 15 cent profit item is typically sold in multiples. If that is the case, then the 15 cent profit item may actually make a more significant contribution to profitability than the 25 cent profit item.

3.2 Preliminaries

Definitions 1 to 6 are used to query the raw data as it is stored in the transaction database.

Definition 1. The *local item count* is the quantity of a particular item purchased in a particular transaction, denoted as $lic(i,t)$, where $i \in I$ and $t \in D$.

Query. *"Give the quantity of item D sold in transaction T_5."*
Result. The local item count for item D in transaction T_5 is $lic(D,T_5) = 4$ (i.e., the value at the intersection of row T_5 and column D).

Definition 2. The *local item amount* is the product of the local item count for a particular item purchased in a particular transaction and the item retail price, denoted as $lia(i,t)$, where $lia(i,t) = lic(i,t) * irp(i)$, $irp(i)$ is the item retail price, $i \in I$ and $t \in D$.

Query. *"Give the value of item D sold in transaction T_5."*
Result. The local item amount for item D in transaction T_5 is $lia(D,T_5) = lic(D,T_5) * irp(D) = 19.00$ (i.e., the value at the intersection of row T_5 and column D multiplied by the retail price of item D).

Definition 3. The *global item count* is the sum of the local item counts for a particular item purchased in all transactions, denoted as $gic(i)$, where $gic(i) = \sum lic(i,t_k)$, $i \in I$, and $t_k \in D$.

Query. *"Give the quantity of item D sold in all transactions."*
Result. The global item count for item D is $gic(D) = lic(D,T_1) + lic(D,T_5) + lic(D,T_7) + lic(D,T_8) = 11$ (i.e., the sum of all values in column D).

Definition 4. The *global item amount* is the sum of the local item amounts for a particular item purchased in all transactions, denoted as $gia(i)$, where $gia(i) = \sum lia(i,t_k)$, $i \in I$, and $t_k \in D$.

Query. *"Give the value of item D sold in all transactions."*
Result. The global item amount for item D is $gia(D) = lia(D,T_1) + lia(D,T_5) + lia(D,T_7) + lia(D,T_8) = 52.25$ (i.e., the value of item D in all transactions).

Definition 5. The *total item count* is the sum of the global item counts for all items purchased in all transactions, denoted as tic, where $tic = \sum gic(i_k)$ and $i_k \in I$, for all k.

Query. *"Give the quantity of all items sold in all transactions."*
Result. The total item count for all transactions is $tic = gic(A)+gic(B)+gic(C)+gic(D)+gic(E)+gic(F) = 66$ (i.e., the sum of all column totals in the transaction database).

Definition 6. The *total item amount* is the sum of the global item amounts for all items purchased in all transactions, denoted as tia, where $tia = \sum gia(i_k)$ and $i_k \in I$, for all k.

Query. *"Give the value of all items sold in all transactions."*
Result. The total item amount for all transactions is $tia = gia(A) + gia(B) + gia(C) + gia(D) + gia(E) + gia(F) = 329.25$ (i.e., the total value of all items in the transaction database).

3.3 Itemset Counting

Definitions 7 to 12 are used to query summary views containing discovered frequent itemsets.

Definition 7. The *local itemset count* is the sum of the local item counts for all transactions which contain a particular item in a particular itemset, denoted as $lisc(i, x)$, where $lisc(i, x) = \sum lic(i, t_k)$, $i \in I$, $x \subseteq I$, $x \in t_k$, and $t_k \in D$.

Query. *"Give the quantity of item C in itemset $\{B, C\}$."*
Result. The local itemset count for item C in itemset $\{B, C\}$ is $lisc(C, \{B, C\}) = lic(C, T_3) + lic(C, T_6) + lic(C, T_{10}) = 5$.

Definition 8. The *local itemset amount* is the sum of the local item amounts for all transactions which contain a particular item in a particular itemset, denoted as $lisa_1(i, x)$, where $lisa_1(i, x) = \sum lia(i, t_k)$, $i \in I$, $x \subseteq I$, $x \in t_k$, and $t_k \in D$. Alternatively, the local itemset amount is the product of the local itemset count for a particular item in a particular itemset and the item retail price, denoted as $lisa_2(i, x)$, where $lisa_2(i, x) = lisc(i, x) * irp(i)$, $irp(i)$ is the item retail price, $i \in I$, and $x \subseteq I$.

Query. *"Give the value of item C in itemset $\{B, C\}$."*
Result. The local itemset amount for item C in itemset $\{B, C\}$ is $lisa_1(C, \{B, C\}) = lia(C, T_3) + lia(C, T_6) + lia(C, T_{10}) = 25.00$, or alternatively, $lisa_2(C, \{B, C\}) = lisc(C, \{B, C\}) * irp(C) = 25.00$.

Definition 9. The *global itemset count* is the sum of the local itemset counts for all items in a particular itemset, denoted as $gisc(x)$, where $gisc(x) = \sum lisc(i_k, x)$, $x \subseteq I$, and $i_k \in x$, for all k.

Query. *"Give the quantity of all items in itemset $\{B, C\}$."*
Result. The global itemset count for itemset $\{B, C\}$ is $gisc(\{B, C\}) = lisc(B, \{B, C\}) + lisc(C, \{B, C\}) = 13$.

Definition 10. The *global itemset amount* is the sum of the local itemset amounts for all items in a particular itemset, denoted as $gisa(x)$, where $gisa(x) = \sum lisa_1(i_k, x)$, $x \subseteq I$, and $i_k \in x$, for all k, or alternatively, $gisa(x) = \sum lisa_2(i_k, x)$, $x \subseteq I$, and $i_k \in x$, for all k.

Query. *"Give the value of all items in itemset $\{B, C\}$."*
Result. The global itemset amount for itemset $\{B, C\}$ is $gisa(\{B, C\}) = lisa_2(B, \{B, C\}) + lisa_2(C, \{B, C\}) = 43.00$.

Definition 11. The *total itemset count* is the sum of the global item counts for all items in a particular itemset, denoted as $tisc(x)$, where $tisc(x) = \sum gic(i_k)$, $x \subseteq I$, and $i_k \in x$.

Query. *"Give the quantity of all items in the transaction database that are in itemset $\{B, C\}$."*
Result. The total itemset count for itemset $\{B, C\}$ is $tisc(\{B, C\}) = gic(B) + gic(C) = 25$.

Definition 12. The *total itemset amount* is the sum of the global item amounts for all items in a particular itemset, denoted as $tisa(x)$, where $tisa(x) = \sum gia(i_k)$, $x \subseteq I$, and $i_k \in x$.

Query. *"Give the value of all items in the transaction database that are in itemset $\{B, C\}$."*
Result. The total itemset amount for itemset $\{B, C\}$ is $tisa(\{B, C\} = gia(B) + gia(C) = 92.00$.

3.4 Share

We now define the notion of share in terms of the definitions from the previous two sections.

Definition 13. The *local share relative to the total item count* for a particular item in a particular itemset is the ratio of the local itemset count to the total item count, expressed as a percentage, denoted as $ls_{tic}(i, x)$, where $ls_{tic}(i, x) = (lisc(i, x)/tic) * 100$, $i \in I$, and $x \subseteq I$.

Query. *"Give the share of the quantity of item F in itemset $\{D, F\}$ in relation to the quantity of all items in the transaction database."*
Result. The local share for item F in itemset $\{D, F\}$ is $ls_{tic}(F, \{D, F\}) = (lisc(F, \{D, F\})/tic) * 100 = 10.6\%$.

Definition 14. The *local share relative to the total item amount* for a particular item in a particular itemset is the ratio of the local itemset amount to the total item amount expressed as a percentage, denoted as $ls_{tia}(i, x)$, where $ls_{tia}(i, x) = (lisa_v(i, x)/tia) * 100$, $i \in I$, $x \subseteq I$, and $v \in \{1, 2\}$.

Query. *"Give the share of the value of item F in itemset $\{D, F\}$ in relation to the value of all items in the transaction database."*
Result. The local share for item F in itemset $\{D, F\}$ is $ls_{tia}(F, \{D, F\}) = (lisa_1(F, \{D, F\})/tia) * 100 = 15.9\%$.

Definition 15. The *global share relative to the total item count* for a particular itemset is the ratio of the global itemset count to the total item count, expressed as a percentage, denoted as $gs_{tic}(x)$, where $gs_{tic}(x) = (gisc(x)/tic) * 100$, $x \subseteq I$.

Query. *"Give the share of the quantity of all items in itemset $\{D, F\}$ in relation to the quantity of all items in the transaction database."*
Result. The global share for itemset $\{D, F\}$ is $gs_{tic}(\{D, F\}) = (gisc(\{D, F\})/tic) * 100 = 24.2\%$.

Definition 16. The *global share relative to the total item amount* for a particular itemset is the ratio of the global itemset amount to the total item amount, expressed as a percentage, denoted as $gs_{tia}(x)$, where $gs_{tia}(x) = (gisa(x)/tia) * 100$, $x \subseteq I$.

Query. *"Give the share of the value of all items in itemset $\{D, F\}$ in relation to the value of all items in the transaction database."*
Result. The global share for itemset {D,F} is $gs_{tia}(\{D, F\}) = (gisa(\{D, F\})/tia) * 100 = 28.9\%$.

Definition 17. The *local share relative to the global itemset count* for a particular item in a particular itemset is the ratio of the local itemset count to the global itemset count, expressed as a percentage, denoted as $ls_{gisc}(i, x)$, where $ls_{gisc}(i, x) = (lisc(i, x)/gisc(x)) * 100$, $i \in I$, and $x \subseteq I$.

Query. *"Give the share of the quantity of item A in itemset $\{A, D\}$ in relation to the quantity of all items in the itemset."*
Result. The local share for item A in itemset $\{A, D\}$ is $ls_{gisc}(A, \{A, D\}) = (lisc(A, \{A, D\})/gisc(\{A, D\})) * 100 = 46.2\%$.

Definition 18. The *local share relative to the global itemset amount* for a particular item in a particular itemset is the ratio of the local itemset amount to the global itemset amount, expressed as a percentage, denoted as $ls_{gisa}(i, x)$, where $ls_{gisa}(i, x) = (lisa_v(i, x)/gisa(x)) * 100$, $i \in I$, $x \subseteq I$, and $v \in \{1, 2\}$.

Query. *"Give the share of the value of item A in itemset $\{A, D\}$ in relation to the value of all items in the itemset."*
Result. The local share for item A in itemset $\{A, D\}$ is $ls_{gisa}(A, \{A, D\}) = (lisa_1(A, \{A, D\})/gisa(\{A, D\})) * 100 = 21.3\%$.

3.5 Frequent Itemsets

A frequent itemset was previously defined as an itemset whose support is greater than some user-specified minimum [5, 31, 3, 4]. We now define frequent itemsets as

used in the share-confidence framework.

Definition 19. An itemset is *locally frequent* if there is an item in the itemset such that at least one of the following conditions holds:

1. The local share relative to the total item count is greater than some user-specified minimum. That is, $ls_{tic}(i_k, x) \geq minshare_1$, where $x \subseteq I$, $i_k \in x$, for some k, and $minshare_1$ is the user-specified minimum share.

2. The local share relative to the total item amount is greater than some user-specified minimum. That is, $ls_{tia}(i_k, x) \geq minshare_2$, where $x \subseteq I$, $i_k \in x$, for some k, and $minshare_2$ is the user-specified minimum share.

Query. *"Give the frequent 2-itemsets whose local share for at least one item is at least 8%."*
Result. The locally frequent 2-itemsets are shown in Table 8. In Table 8, the *Itemset* column describes the items in the itemset, the *TIDs* column describes the transaction identifiers that contain the corresponding itemset, the $ls_{tic}(i_1, x)$ and $ls_{tic}(i_2, x)$ columns describe the local share relative to the total item count for items one and two, respectively, and the $ls_{tia}(i_1, x)$ and $ls_{tia}(i_2, x)$ columns describe the local share relative to the total item amount for items one and two, respectively.

Table 8: Locally frequent 2-itemsets

Itemset (x)	*TIDs*	$ls_{tic}(i_1, x)$ (%)	$ls_{tic}(i_2, x)$ (%)	$ls_{tia}(i_1, x)$ (%)	$ls_{tia}(i_2, x)$ (%)
$\{A, D\}$	T_1, T_7, T_8	**9.09**	**10.60**	2.73	**10.10**
$\{B, E\}$	T_2, T_6, T_9, T_{10}	**16.67**	7.58	7.52	**15.19**
$\{B, C\}$	T_3, T_6, T_{10}	**12.12**	7.58	5.47	7.59
$\{C, E\}$	T_4, T_6, T_{10}	**9.09**	4.55	**9.11**	**9.11**
$\{C, F\}$	T_4, T_{11}	**9.09**	6.06	**9.11**	**9.11**
$\{E, F\}$	T_4, T_7, T_9	7.58	**10.60**	**15.19**	**15.95**
$\{D, F\}$	T_5, T_7, T_8	**13.64**	**10.60**	**12.98**	**15.95**
$\{A, F\}$	T_7, T_8	7.58	**9.09**	2.27	**13.67**
$\{D, E\}$	T_7	1.52	6.06	1.44	**12.15**

Definition 20. An itemset is *globally frequent* if every item in the itemset is locally frequent.

Query. *"Give the frequent 2-itemsets whose local share for all items is at least 8%."*
Result. The globally frequent 2-itemsets are shown in Table 9. The columns in Table 9 have the same meaning as in Table 8.

Table 9: Globally frequent 2-itemsets

Itemset (x)	*TIDs*	$ls_{tic}(i_1, x)$ (%)	$ls_{tic}(i_2, x)$ (%)	$ls_{tia}(i_1, x)$ (%)	$ls_{tia}(i_2, x)$ (%)
$\{A, D\}$	T_1, T_7, T_8	**9.09**	**10.60**	2.73	10.10
$\{C, E\}$	T_4, T_6, T_{10}	9.09	4.55	**9.11**	**9.11**
$\{C, F\}$	T_4, T_{11}	9.09	6.06	**9.11**	**9.11**
$\{E, F\}$	T_4, T_7, T_9	7.58	10.6	**15.19**	**15.95**
$\{D, F\}$	T_5, T_7, T_8	**13.64**	**10.60**	**12.98**	**15.95**

3.6 Confidence

Confidence in an association rule $X \Rightarrow Y$ was previously defined as the ratio of the number of transactions containing itemset $X \cup Y$ to the number of transactions containing itemset X [5, 31, 3, 4]. We now define confidence as used in the share-confidence framework.

Definition 21. The *count confidence* in an association rule $X \Rightarrow Y$ is the ratio of the sum of the local itemset counts for all items in itemset X contained in $X \cup Y$ to the global itemset count for itemset X, expressed as a percentage, denoted as $cc(x, x \cup y)$, where $cc(x, x \cup y) = (\sum lisc(i_k, x \cup y)/gisc(x)) * 100$, $x \subseteq I$, $x \cup y \subseteq I$, and $i_k \in x$, for all k.

Query. *"Give the count confidence for the association rule* $\{B, C\} \Rightarrow \{E\}$*."*
Result. The count confidence for the association rule $\{B, C\} \Rightarrow \{E\}$ is $cc(\{B, C\}, \{B, C, E\}) = ((lisc(B, \{B, C, E\}) + lisc(C, \{B, C, E\}))/gisc(\{B, C\})) * 100 = 76.9\%$.

Definition 22. The *amount confidence* in an association rule $X \Rightarrow Y$ is the ratio of the sum of the local itemset amounts for all items in itemset X contained in $X \cup Y$ to the global itemset amount for itemset X, expressed as a percentage, denoted as $ac(x, x \cup y)$, where $ac(x, x \cup y) = (\sum lisa_v(i_k, x \cup y)/gisa(x)) * 100$, $x \subseteq I$, $x \cup y \subseteq I$, $i_k \in x$, for all k, and $v \in \{1, 2\}$.

Query. *"Give the amount confidence for the association rule* $\{B, C\} \Rightarrow \{E\}$*."*
Result. The amount confidence for the association rule $\{B, C\} \Rightarrow \{E\}$ is $ac(\{B, C\}, \{B, C, E\}) = ((lisa_2(B, \{B, C, E\}) + lisa_2(C, \{B, C, E\}))/gisa(\{B, C\})) * 100 = 59.9\%$.

4 Share vs Support

We now compare share with support to show how the choice of metric can lead to different conclusions when analyzing the same transactions from the database. This is clearly demonstrated in Table 10, where the *Itemset* column describes the items in the itemset, the gs_{tic} and gs_{tia} columns describe the global share relative to the total item count and total item amount, respectively. The *Support* column has the same meaning as the like-named column in Table 2.

Table 10: Share and support for some 1-, 2-, and 3-itemsets

Itemset (x)	*TIDs*	$gs_{tic}(x)$ (%)	$gs_{tia}(x)$ (%)	*Support* (%)
$\{A\}$	T_1, T_3, T_7, T_8	15.15	4.56	36.36
$\{B\}$	$T_2, T_3, T_6, T_9, T_{10}$	18.18	8.20	45.45
$\{C\}$	$T_1, T_3, T_4, T_6, T_{10}, T_{11}$	19.70	19.74	54.55
$\{D\}$	T_1, T_5, T_7, T_8	16.67	15.87	36.36
$\{E\}$	$T_2, T_4, T_6, T_7, T_9, T_{10}$	12.12	24.30	54.55
$\{F\}$	$T_4, T_5, T_7, T_8, T_9, T_{11}$	18.18	27.33	54.55
$\{A, C\}$	T_1, T_3	13.64	8.35	18.18
$\{A, B\}$	T_3	7.58	2.51	9.09
$\{C, E\}$	T_4, T_6, T_{10}	13.64	18.22	27.27
$\{A, B, C\}$	T_3	10.61	5.54	9.09
$\{B, C, E\}$	T_6, T_{10}	18.18	15.41	18.18
$\{A, D, F\}$	T_7, T_8	24.24	23.16	18.18

Support can overstate the contribution of an itemset to total sales. For example, the global share relative to the total item count and total item amount for itemset $\{C\}$ is 19.70% and 19.74%, respectively. This says that itemset $\{C\}$ comprises approximately one-fifth of total sales in terms of both the quantity and value of items sold. However, support indicates that itemset $\{C\}$ has the support of over half of all transactions, significantly overstating its relative contribution to total sales. With support of over 36%, the contribution to total sales of itemset $\{A\}$ is also overstated. Itemset $\{A\}$ comprises approximately 15% of the quantity of items sold, but only approximately 5% of the value of items sold.

Support can also understate the contribution of an itemset to total sales. For example, the global share relative to the total item count and total item amount for itemset $\{A, D, F\}$ is 24.24% and 23.16%, respectively. Again, this says that itemset $\{A, D, F\}$ comprises almost one-quarter of total sales in terms of both the quantity and value of items sold. However, support indicates that itemset $\{A, D, F\}$ has the support of less than one-fifth of all transactions, significantly understating its relative contribution to total sales.

Finally, support can indicate that multiple itemsets have the same support, but the contribution to total sales of each itemset can be significantly different. For example, itemsets $\{A, C\}$, $\{B, C, E\}$, and $\{A, D, F\}$ have the same support. But analysis of the corresponding global shares shows that the contribution to total sales for each of these itemsets is different, with the global share relative to the total item count ranging from 13.64% for itemset $\{A, C\}$ to 24.24% for itemset $\{A, D, F\}$, and the global share relative to the total item amount ranging from 8.35% for itemset $\{A, C\}$ to 23.16% for itemset $\{A, D, F\}$. Clearly, these examples show that analysis based upon share would be more meaningful than those based upon support.

5 Characterized Itemsets

A characterized itemset is an itemset that has been partitioned into classes based upon attributes which define specific characteristics of the itemset. Characterizing information is typically obtained from external databases containing customer, census, or lifestyle data. Every itemset contained in a transaction can be partitioned into a specific class based upon the characteristic attributes. The *CI* algorithm is used in the share-confidence framework for generating characterized itemsets.

This section is organized as follows. In Section 5.1, we present the *CI* algorithm for generating characterized itemsets. In Section 5.2, we present an example to demonstrate the operation of *CI*. In Section 5.3, we analyze the running time and space requirements of *CI* and contrast it with *Apriori*.

5.1 The CI *Algorithm*

In the description of *CI* that follows, let L_k^* and C_k^* denote the set of characterized frequent itemsets from pass k and the set of characterized candidate itemsets from pass k, respectively, and let R^* denote the characteristic relation. The k-th pass of

the algorithm works as follows.

1. Repeat steps *2* to *5* until no new candidate itemsets are generated in pass $(k-1)$.
2. Generate the candidate k-itemsets in C_k^* from the frequent $(k-1)$-itemsets in L_{k-1}^* using the *Apriori* method described for C_k and L_{k-1} in Section 2.2.
3. Partition the frequent $(k-1)$-itemsets in L_{k-1}^* and update the candidate itemsets in C_k^*.
   - **a.** Repeat steps *3-b* to *3-f* until there are no more transactions to be retrieved from the database.
   - **b.** Retrieve the next transaction from the database.
   - **c.** Retrieve the associated characteristic tuple from R^*.
   - **d.** For each $(k-1)$-itemset in the transaction, if it is contained in L_{k-1}^*, update the characteristic tuple.
     - (*i*) If itemset summary attributes already exist for this $(k-1)$-itemset in the characteristic tuple, go to step *3-d-*(*ii*). Otherwise, create new itemset summary attributes in the characteristic tuple.
     - (*ii*) Increment the total quantity and total value attributes for this $(k-1)$-itemset in the characteristic tuple.
   - **e.** If the characteristic tuple has been updated, save it in R^*.
   - **f.** For each k-itemset in the transaction, if it is contained in C_k^*, increment the associated total quantity and total value attributes.
4. Save the frequent k-itemsets in L_k^*.
   - **a.** Repeat steps *4-b* and *4-c* until there are no more itemset tuples in C_k^*.
   - **b.** Retrieve the next itemset tuple from C_k^*.
   - **c.** If the share of this itemset tuple is greater than the minimum specified, copy the itemset tuple to L_k^*.
5. Delete C_k^*.
6. Save R^*.

The first pass of the algorithm is a special pass which generates the frequent 1-itemsets and the characteristic relation, as follows.

1. Generate the candidate 1-itemsets in C_1^* and the characteristic relation R^*.
   - **a.** Repeat steps *1-b* to *1-f* until there are no more transactions to be retrieved from the database.
   - **b.** Retrieve the next transaction from the database.

c. For each 1-itemset in the transaction, if an itemset tuple already exists in C_1^*, go step *1-d*. Otherwise, create a new itemset tuple in C_1^*.

d. For each 1-itemset in the transaction, increment the total quantity and total value attributes of the associated itemset tuple in C_1^*.

e. Using the appropriate key(s), retrieve the characterizing attributes for this transaction from the external database(s).

f. If a characteristic tuple containing these characteristics already exists in R^*, go step *1-b*. Otherwise, create a new characteristic tuple in R^*.

2. Save the frequent 1-itemsets in L_1^*.

   a. Repeat steps *2-b* and *2-c* until there are no more itemset tuples in C_1^*.

   b. Retrieve the next itemset tuple from C_1^*.

   c. If the share of this itemset tuple is greater than the minimum specified, copy the itemset tuple to L_1^*.

3. Delete C_1^*.

4. Save R^*.

5.2 Example

We now present an example to demonstrate *CI*. Assume we are given the transaction and external customer databases shown in Tables 11 and 12, respectively. Also assume the user-specified minimum share is 15%. In Table 11, the *CID* column describes the customer identifier and the other column descriptions have the same meaning as the like-named columns in Table 1. In Table 12, the *CID* column also describes the customer identifier and the *Char. 1*, *Char. 2*, and *Char. 3* columns describe some characteristics associated with the customer. Tables 11 and 12 are joined on the *CID* attribute. Our task is to trace through the first three passes of *CI*. For this example, we use only the local share relative to the total item count to determine share, and select only those itemsets from each pass that are globally frequent.

Table 11: A smaller example transaction database with counts

TID	*A*	*B*	*C*	*D*	*E*
T_1	1	2	5	0	0
T_2	4	1	1	3	2
T_3	3	0	2	1	0
T_4	5	0	4	2	1
T_5	2	3	3	4	0
	15	6	15	10	3

After the first pass, *CI* generates L_1^* and R^* as shown in Tables 13 and 14, respectively. In Table 13, the *Itemset* column describes the items in the itemset and the *Share* column describes the total item count local share. In Table 14, the *Char. 1* and *Char. 2* columns describe the characteristics retrieved from the external

Table 12: The external customer database

CID	*Char. 1*	*Char. 2*	*Char. 3*
1	R	X	K
2	S	Y	M
3	S	Z	N
4	R	X	M
5	S	Y	N

customer database in Table 12, and the *TIDs* column describes the transactions that share the corresponding characteristics (the *TIDs* are not actually stored in R^* and are merely shown here for demonstration purposes). The domain of the first and second characteristic is $\{R, S\}$ and $\{X, Y, Z\}$, respectively.

Table 13: Frequent itemsets contained in L_1^*

Itemset	*Share* (%)
$\{A\}$	30.6
$\{C\}$	30.6
$\{D\}$	20.4

Table 14: R^* after the first pass

Char. 1	*Char. 2*	*TIDs*
R	X	T_1, T_4
S	Y	T_2, T_5
S	Z	T_3

After the second pass, *CI* generates L_2^* and updates R^* as shown in Tables 15 and 16, respectively. In Tables 15 and 16, the column descriptions have the same meaning as the like-named columns in Table 13 and Table 14, respectively. Also in Table 16, the *Itemsets* column describes the frequent itemsets from the previous pass that share the identified characteristics.

Table 15: Frequent itemsets contained in L_2^*

Itemset $\{X, Y\}$	*Share (X)* (%)	*Share (Y)* (%)
$\{A, C\}$	30.6	30.6
$\{A, D\}$	28.6	20.4
$\{C, D\}$	20.4	20.4

Table 16: R^* after the second pass

Char. 1	*Char. 2*	*TIDs*	*Itemsets*
R	X	T_1, T_4	$\langle\{A\}, 6\rangle$, $\langle\{C\}, 9\rangle$, $\langle\{D\}, 2\rangle$
S	Y	T_2, T_5	$\langle\{A\}, 6\rangle$, $\langle\{C\}, 4\rangle$, $\langle\{D\}, 7\rangle$
S	Z	T_3	$\langle\{A\}, 3\rangle$, $\langle\{C\}, 2\rangle$, $\langle\{D\}, 1\rangle$

After the third pass, *CI* updates R^* as shown in Table 17. In Table 17, the column descriptions have the same meaning as the like-named columns in Table 16. No L_3^* is generated as there are no globally frequent itemsets, so we are done. The characterized itemsets are contained in R^* of Table 17.

Table 17: R^* after the third pass

Char. 1	*Char. 2*	*TIDs*	*Itemsets*
R	X	T_1, T_4	$\langle\{A\}, 6\rangle$, $\langle\{C\}, 9\rangle$, $\langle\{D\}, 2\rangle$, $\langle\{A, C\}, 15\rangle$, $\langle\{A, D\}, 7\rangle$, $\langle\{C, D\}, 6\rangle$
S	Y	T_2, T_5	$\langle\{A\}, 6\rangle$, $\langle\{C\}, 4\rangle$, $\langle\{D\}, 7\rangle$, $\langle\{A, C\}, 10\rangle$, $\langle\{A, D\}, 13\rangle$, $\langle\{C, D\}, 11\rangle$
S	Z	T_3	$\langle\{A\}, 3\rangle$, $\langle\{C\}, 2\rangle$, $\langle\{D\}, 1\rangle$, $\langle\{A, C\}, 5\rangle$, $\langle\{A, D\}, 4\rangle$, $\langle\{C, D\}, 7\rangle$

5.3 Time and Space Analysis

The running time of *Apriori* is given as $O(\|c\| * |d|)$, where $\|c\|$ is the number of candidate itemsets in all iterations of the algorithm and $|d|$ is the number of transactions [3]. *CI* is also $O(\|c\| * |d|)$ when an *Apriori*-like extraction of itemsets from each transaction is assumed. The primary differences between *CI* and *Apriori* are that *CI* extracts characterizing attributes when the transactions are read in the first pass and in the partitioning of the itemsets in pass k, $k > 1$. In the k-th pass, each transaction is scanned for frequent itemsets of size $k - 1$. Now the same transaction is scanned for all candidate itemsets of size $k - 1$ in the previous pass, so the size of the $k - 1$ frequent itemset is at most as large as the $k - 1$ candidate set. Once a frequent itemset has been identified, the only other operation necessary is inserting this itemset into the appropriate partition associated with the characteristic tuple. If we use a multi-dimensional array to store and access characteristic tuples as is done for *AOG* in the *FIGR* algorithm [10], then lookup time is constant and the time to insert into the partitioned set will be the same as the time to update the candidate summary. Thus, *CI* will take at most $2 * O(\|c\| * |d|)$, which is $O(\|c\| * |d|)$.

The space requirements of *Apriori* are approximately $O(c)$, where c denotes the size of the largest candidate itemset in any given pass. All other space requirements are constant, or minimal, if we assume that frequent itemsets are archived after each iteration of the algorithm (i.e., written to disk and not kept in memory). The space requirements of *CI* will directly depend on the number of partitions created by the characterization process. Given a characteristic relation with p distinct tuples, *CI* will require at most $p * O(c)$ space, since it must must record counts for p partitions of frequent itemsets, and the number of frequent itemsets in the frequent set is at most c, the number of candidates in the largest candidate set. In practice, however, the size of the frequent set is much smaller than the size of the candidate set. Since the size of the characteristic relation is bounded by a constant [13, 10], *CI* is also $O(c)$.

6 Generalizing Itemsets

We now describe how new knowledge can be discovered by generalizing itemsets. This section is organized as follows. In Section 6.1, we provide an overview of attribute-oriented generalization. In Section 6.2, we discuss our approach to generalizing characterized itemsets. In Section 6.3, we discuss related work on generalizing itemsets.

6.1 *Attribute-Oriented Generalization*

The data structures containing the characterized itemsets in L_k^*, generated by *CI*, form a relation. In a relation, transforming a specific data description into a more general one is called generalization. Generalization techniques include the dropping condition and climbing tree methods [28]. The climbing tree method transforms the data in a database by repeatedly replacing specific attribute values with more general concepts according to user-defined concept hierarchies. A concept hierarchy (CH) associated with an attribute in a database is defined as follows. Let A be an attribute in a database whose domain values are represented by $V = V_k \cup V_g$, where V_k is the set of values known to be present in the data and V_g is the set of generalized values, including the most general value ANY. A CH on A is a directed acyclic graph (tree) on V with a single source node corresponding to the value ANY, sink (leaf) nodes corresponding to the values from V_k, and internal (intermediate) nodes corresponding to the values from V_g. For example, CHs for the *Income* and *Territory* attributes in a sales database are shown in Figure 1. Knowledge about the higher level concepts (i.e., non-leaf nodes) can be discovered through a general-to-specific search beginning at the leaf nodes.

The dropping condition method transforms the data in a database by removing a condition from a conjunction of conditions, so that the remaining conjunction of conditions is more general. For example, assume the conjunction of conditions $(shape = round \wedge size = large \wedge colour = white)$ describes the concept *ball*. Removing the condition $colour = white$, which is equivalent to generalizing the *colour* attribute to ANY, yields the conjunction of conditions $(shape = round \wedge size = large)$. The concept *ball* is now more general because instances can now be large, round objects of any colour.

6.2 *Generalizing Characterized Itemsets*

In our approach, we use attributes from the transaction database to join on the key fields of external database(s). The required characteristic attributes from the external database(s) are then inserted into the characteristic relation R^*. Assuming the characteristic attributes of R^* represent low level concepts, we can associate CHs with the attributes to discover new knowledge about the corresponding frequent itemsets in terms of the higher level concepts in the CHs. We can then generalize R^* using the climbing tree and dropping condition methods described in the previous section. This approach provides basis for efficient summarization and drilling down in the transaction database.

Fast and efficient implementations of AOG [13, 10, 24] can be used to generate summaries where the characteristic attributes are generalized according to the CHs. If the CHs have relatively few levels (i.e., fewer than 10), and if multiple CHs are available for some attributes, then the *All_Gen* algorithm [17, 21] can be used to generate all possible summaries.

6.3 Related Work

Several algorithms have been proposed for finding specialized or generalized itemsets where CHs are used to classify items. Here we discuss those presented in [19] and [31]. In [19], a top-down progressive deepening method is proposed which discovers multi-level association rules. This method first discovers frequent itemsets at the top-most level of the CH associated with an attribute, and then progressively descends the CH discovering frequent itemsets for lower level concepts. For example, if an association rule is discovered from a frequent itemset, such as *milk*⇒*bread*, it may be that when we descend the corresponding CHs for milk and bread, we discover that *2% milk*⇒*whole wheat bread* is also a valid association rule contained in a frequent itemset. The proposed method is flexible because different thresholds can be assigned at each level of the associated CHs, yielding a high potential for discovering interesting association rules.

A similar idea is presented in [31], where the descriptions for items in the discovered association rules may come from any level of the associated CH. The method differs from that presented in [19] in that redundant rules are eliminated from further consideration. For example, if the association rule *milk*⇒*bread* has 8% support and 70% confidence, and the association rule *2% milk*⇒*bread* has 2% support and 70% confidence, then the latter association rule is considered redundant because it is less general and conveys no new information.

Both the related methods discussed here differ from our approach in that they use CHs to classify the items, while we use CHs to classify the characteristic attributes. Our approach allows knowledge discovery to be guided by customized, user-defined CHs associated with the characteristic attributes. By extracting the frequent itemsets and joining in the characteristic attributes, we have all the information that we need to discover interesting association rules. That is, by focusing on the frequent itemsets, we can determine the characteristic profiles of the customers that have purchased the itemsets. Or, by focusing on the characteristic profiles of customers, we can determine the itemsets that have been purchased, classified by customer profile. We can analyze the summary data in this way, without having to re-read the database or do any extra processing

7 Discovering Association Rules: An Extended Example

We now present an extended example of discovering association rules using share measures, characterized itemsets, and generalization. We will attempt to answer our sales manager's queries from Section 1, which we refine and state here for convenience. That is, "Give the profile, based upon income and territory, of those customers who purchase frequent itemsets. Set the minimum share at 10% and the minimum confidence at 60%." Since the query does not specify whether share is to be based upon the quantity or value of the itemset, we will assume that if the share for either the quantity or value is greater than 10%, then the itemset will be considered frequent. Also, the query does not state whether the locally or globally

frequent measure should be used, so we will select locally frequent itemsets. We restrict our discussion to 2- and 3-itemsets. Again, refer to the transaction database in Table 6.

The seven locally frequent 2-itemsets and two locally frequent 3-itemsets which satisfy our query, from a possible 14 2-itemsets and 8 3-itemsets, respectively, are shown in Table 18, where the share measures exceeding 10% are shown in bold. In Table 18, the column descriptions have the same meaning as the like-named columns in Table 8.

Table 18: 2- and 3-itemsets satisfying the query

Itemset (x)	*TIDs*	$ls_{tic}(i_1, x)$ (%)	$ls_{tic}(i_2, x)$ (%)	$ls_{tic}(i_3, x)$ (%)	$ls_{tia}(i_1, x)$ (%)	$ls_{tia}(i_2, x)$ (%)	$ls_{tia}(i_3, x)$ (%)
$\{A, D\}$	T_1,T_7,T_8	9.09	**10.60**	-	2.73	**10.10**	-
$\{B, E\}$	T_2,T_6,T_9,T_{10}	**16.67**	7.58	-	7.52	**15.19**	-
$\{B, C\}$	T_3,T_6,T_{10}	**12.12**	7.58	-	5.47	7.59	-
$\{E, F\}$	T_4,T_7,T_9	7.58	**10.60**	-	**15.19**	**15.95**	-
$\{D, F\}$	T_5,T_7,T_8	**13.64**	**10.60**	-	**12.98**	**15.95**	-
$\{A, F\}$	T_7,T_8	7.58	9.09	-	2.27	**13.67**	-
$\{D, E\}$	T_7	1.52	6.06	-	1.44	**12.15**	-
$\{B, C, E\}$	T_6,T_{10}	**10.60**	7.58	3.03	4.78	9.11	6.07
$\{A, D, F\}$	T_7,T_8	7.58	7.58	9.09	2.28	7.21	**13.60**

The 26 association rules which can be generated from the nine itemsets in Table 18 are shown in Table 19. In Table 19, the *Association Rule* column describes the association rules, the $lisc(x, x \cup y)$ and $gisc(x)$ columns describe the local itemset count and global itemset counts for itemset x, respectively, and the $cc(x, x \cup y)$ column describes the count confidence for itemset x. The count confidence is shown in bold for 12 of the 26 association rules which exceed 60% confidence. We prune the number of association rules by selecting only one where similar association rules satisfy the query. For example, for the association rules $\{A\} \Rightarrow \{D\}$ and $\{D\} \Rightarrow \{A\}$ derived from itemset $\{A, D\}$, and the association rules $\{B, C\} \Rightarrow \{E\}$ and $\{B\} \Rightarrow \{C, E\}$ derived from itemset $\{B, C, E\}$, we prune $\{A\} \Rightarrow \{D\}$ and $\{B\} \Rightarrow \{C, E\}$ because they have lower confidence.

The remaining association rules are shown in Table 20. In Table 20, the *Association Rule* column describes the association rule, the *TIDs* column describes the transaction identifiers that contain the corresponding association rule, the *Income* column describes the first characteristic attribute, the *Territory* column describes the second characteristic attribute, the $\sum lic(i_1)$, $\sum lic(i_2)$, and $\sum lic(i_3)$ columns describe the sum of the local item counts for items i_1, i_2, and i_3, respectively, in the transactions containing the itemset, the $gisc(i_1 \cup i_2 \cup i_3)$ describes the global itemset count for itemset $\{i_1 \cup i_2 \cup i_3\}$, and the *Partition Share* column describes the partition share. None of the characterizing attributes in Table 20 has been generalized, and using association rule $\{D\} \Rightarrow \{A\}$ as an example, the table can be interpreted as follows. The association rule $\{D\} \Rightarrow \{A\}$ applies to three transaction, T_1, T_7, and T_8. Transaction T_1 was initiated by a customer with an annual income of approximately \$12,000, who lives in the area near the store designated as territory 9.

The characteristic attributes describing the association rules in Table 20 may be generalized to any level in the associated concept hierarchies, shown in Figure 1. For

Table 19: 26 possible association rules

Association Rule $(x \Rightarrow y)$	$lisc(x, x \cup y)$	$gisc(x)$	$cc(x, x \cup y)$ (%)
$\{A\} \Rightarrow \{D\}$	6	10	**60.00**
$\{D\} \Rightarrow \{A\}$	7	11	**63.63**
$\{B\} \Rightarrow \{E\}$	11	12	**91.67**
$\{E\} \Rightarrow \{B\}$	5	8	**62.50**
$\{B\} \Rightarrow \{C\}$	8	12	**66.67**
$\{C\} \Rightarrow \{B\}$	5	13	38.46
$\{E\} \Rightarrow \{F\}$	5	8	**62.50**
$\{F\} \Rightarrow \{E\}$	7	12	58.33
$\{D\} \Rightarrow \{F\}$	9	11	**81.82**
$\{F\} \Rightarrow \{D\}$	7	12	58.33
$\{A\} \Rightarrow \{F\}$	5	10	50.00
$\{F\} \Rightarrow \{A\}$	6	12	50.00
$\{D\} \Rightarrow \{E\}$	1	11	9.09
$\{E\} \Rightarrow \{D\}$	2	8	25.00
$\{B, C\} \Rightarrow \{E\}$	10	13	**76.92**
$\{B, E\} \Rightarrow \{C\}$	9	16	56.25
$\{C, E\} \Rightarrow \{B\}$	5	9	55.56
$\{B\} \Rightarrow \{C, E\}$	9	12	**75.00**
$\{C\} \Rightarrow \{B, E\}$	3	13	23.08
$\{E\} \Rightarrow \{B, C\}$	2	8	25.00
$\{A, D\} \Rightarrow \{F\}$	10	13	**76.92**
$\{A, F\} \Rightarrow \{D\}$	11	11	**100.00**
$\{D, F\} \Rightarrow \{A\}$	11	16	**68.75**
$\{A\} \Rightarrow \{D, F\}$	5	10	50.00
$\{D\} \Rightarrow \{A, F\}$	5	11	45.45
$\{F\} \Rightarrow \{A, D\}$	6	12	50.00

Table 20: Ungeneralized association rules

Association Rule $(x \Rightarrow y)$	*TIDs*	*Income*	*Territory*	$\sum lic(i_1)$	$\sum lic(i_2)$	$\sum lic(i_3)$	$gisc$ $(x \cup y)$	*Partition Share* (%)
$\{D\} \Rightarrow \{A\}$	T_1	12K	9	1	2	-	13	23.08
	T_7	55K	8	3	1	-		30.77
	T_8	17K	7	2	4	-		46.15
$\{B\} \Rightarrow \{E\}$	T_2	48K	3	3	1	-	16	25.00
	T_6	32K	4	3	1	-		25.00
	T_9	36K	6	1	2	-		18.75
	T_{10}	64K	2	4	1	-		31.25
$\{B\} \Rightarrow \{C\}$	T_3	21K	10	1	2	-	13	23.08
	T_6	32K	4	3	2	-		38.46
	T_{10}	44K	2	4	1	-		38.46
$\{E\} \Rightarrow \{F\}$	T_4	70K	9	1	2	-	12	25.00
	T_7	55K	8	2	4	-		50.00
	T_9	36K	7	2	1	-		50.00
$\{D\} \Rightarrow \{F\}$	T_5	5K	6	4	1	-	16	31.25
	T_7	15K	8	1	4	-		31.25
	T_8	17K	5	4	2	-		37.50
$\{B, C\} \Rightarrow \{E\}$	T_6	55K	2	3	2	1	12	50.00
	T_{10}	75K	7	4	1	1		50.00
$\{A, F\} \Rightarrow \{D\}$	T_7	18K	5	3	1	4	16	50.00
	T_8	31K	9	2	4	2		50.00

example, two possible generalizations are shown in Tables 21 and 22. In Tables 21 and 22, the income and territory attributes have been generalized first, respectively, and the column descriptions have the same meaning as the like-named columns in Table 20.

The characteristic attribute to generalize first can be determined in accordance with the guidelines specified in [6], where lookahead and predictive strategies are suggested. Using the lookahead strategy, a relation with m attributes is used to generate m new generalized relations, each of which is created by generalizing a different attribute to the next highest level in its associated CH. Using the predictive strategy, the complexity of the CHs associated with the attributes is considered when determining the attributes to generalize next (i.e., which attribute has the greatest or least distinct values). The latter is faster, requires less space, and gives similar results.

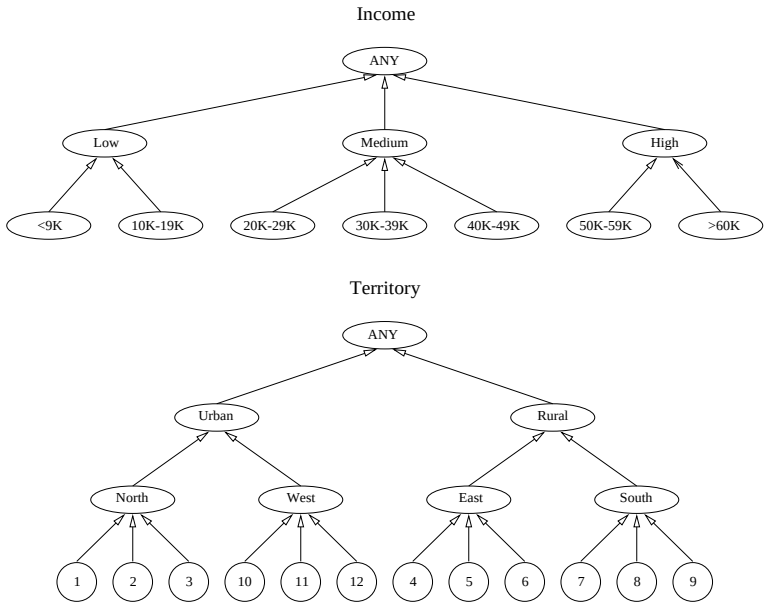

Figure 1: Concept hierarchies for characteristic attributes

In Tables 21 and 22, some buying patterns are beginning to become clear. For example, in Table 21, association rules $\{B\} \Rightarrow \{C\}$ and $\{D\} \Rightarrow \{F\}$ contain items that are purchased by medium and low income customers, respectively. Statements regarding these association rules can be made as follows.

"The purchase of itemset $\{B, C\}$ comprises a 19.7% share of the quantity of all items sold, where 100% of the purchases are by customers with a medium income."

"The purchase of itemset $\{D, F\}$ comprises a 24.24% share of the quantity of all items sold, where 100% of the purchases are by customers with a low income."

In Table 22, association rules $\{D\} \Rightarrow \{A\}$ and $\{E\} \Rightarrow \{F\}$ contain items that are purchased by customers living in territories south of the store. Statements regarding these association rules can be made as follows.

"The purchase of itemset $\{A, D\}$ comprises a 19.7% share of the quantity of all items sold, where 100% of the purchases are by customers in southern territories."

"The purchase of itemset $\{E, F\}$ comprises a 18.18% share of the quantity of all items sold, where 100% of the purchases are by customers in southern territories."

A summary statement can be made regarding the previous two statements as follows.

"Two itemsets comprise a 37.88% share of the quantity of all items sold, where 100% of the purchases are by customers in southern territories."

Of course, the statements may not always be so definitive. For example, using this same table, we can look at association rules $\{B\} \Rightarrow \{C\}$ and $\{D\} \Rightarrow \{F\}$ in a different way. So, alternative statements regarding these association rules can be made as follows.

"The purchase of itemset $\{B, C\}$ comprises a 19.7% share of the quantity of all items sold, where 23.08% of the purchases are by customers in western territories and 38.46% of the purchases are by customers in each of the eastern and northern territories, respectively."

"The purchase of itemset $\{D, F\}$ comprises a 24.24% share of the quantity of all items sold, where 68.75% of the purchases are by customers in eastern territories and 31.25% of the purchases are by customers in southern territories."

Table 21: Association rules with income generalized first

Association Rule $(x \Rightarrow y)$	*TIDs*	*Income*	*Territory*	$\sum lic(i_1)$	$\sum lic(i_2)$	$\sum lic(i_3)$	$gisc$ $(x \cup y)$	*Partition Share* (%)
$\{D\} \Rightarrow \{A\}$	T_1, T_8	L	S	3	6	-	13	69.23
	T_7	H	S	3	1	-		30.77
$\{B\} \Rightarrow \{E\}$	T_2, T_6, T_9	M	N,E	7	4	-	16	68.75
	T_{10}	H	N	4	1	-		31.25
$\{B\} \Rightarrow \{C\}$	T_3, T_6, T_{10}	M	N,E,W	8	5	-	13	100.00
$\{E\} \Rightarrow \{F\}$	T_4, T_7	H	S	3	6	-	12	75.00
	T_9	M	S	2	1	-		25.00
$\{D\} \Rightarrow \{F\}$	T_5, T_7, T_8	L	E,S	9	7	-	16	100.00
$\{B, C\} \Rightarrow \{E\}$	T_6, T_{10}	H	N,S	7	3	2	12	100.00
$\{A, F\} \Rightarrow \{D\}$	T_7	L	E	3	1	4	16	50.00
	T_8	M	S	2	4	2		50.00

Table 22: Association rules with territory generalized first

Association Rule $(x \Rightarrow y)$	*TIDs*	*Income*	*Territory*	$\sum lic(i_1)$	$\sum lic(i_2)$	$\sum lic(i_3)$	$gisc$ $(x \cup y)$	*Partition Share* (%)
$\{D\} \Rightarrow \{A\}$	T_1, T_7, T_8	L,H	S	6	7	-	13	100.00
$\{B\} \Rightarrow \{E\}$	T_2, T_{10}	M,H	N	7	2	-	16	56.25
	T_6, T_9	M	E	4	3	-		43.75
$\{B\} \Rightarrow \{C\}$	T_3	M	W	1	2	-	13	23.08
	T_6	M	E	3	2	-		38.46
	T_{10}	M	N	4	1	-		38.46
$\{E\} \Rightarrow \{F\}$	T_4, T_7, T_9	M,H	S	2	1	-	12	100.00
$\{D\} \Rightarrow \{F\}$	T_5, T_8	L	E	8	3	-	16	68.75
	T_7	L	S	1	4	-		31.25
$\{B, C\} \Rightarrow \{E\}$	T_6	H	N	3	2	1	12	50.00
	T_{10}	H	S	4	1	1		50.00
$\{A, F\} \Rightarrow \{D\}$	T_7	L	E	3	1	4	16	50.00
	T_8	M	S	2	4	2		50.00

Generalizing the territory attribute in Table 22 to the next higher level results in the generalization shown in Table 23. Using this table, we can look at association rules $\{B\} \Rightarrow \{C\}$ and $\{D\} \Rightarrow \{F\}$ in yet another way. For example, statements regarding these association rules can be made as follows.

"The purchase of itemset $\{B, C\}$ comprises a 19.7% share of the quantity of all items sold, where 61.54% of the purchases are by medium income customers in urban territories and 38.46% of the purchases are by medium income customers in rural territories."

"The purchase of itemset $\{D, F\}$ comprises a 24.24% share of the quantity of all items sold, where 100% of the purchases are made by low income customers in rural territories."

Table 23: Association rules with territory generalized to next higher level

Association Rule $(x \Rightarrow y)$	*TIDs*	*Income*	*Territory*	$\sum lic(i_1)$	$\sum lic(i_2)$	$\sum lic(i_3)$	$gisc$ $(x \cup y)$	*Partition Share* (%)
$\{D\} \Rightarrow \{A\}$	T_1, T_7, T_8	L,H	R	6	7	-	13	100.00
$\{B\} \Rightarrow \{E\}$	T_2, T_{10}	M,H	U	7	2	-	16	56.25
	T_6, T_9	M	R	4	3	-		43.75
$\{B\} \Rightarrow \{C\}$	T_3, T_{10}	M	U	5	3	-	13	61.54
	T_6	M	R	3	2	-		38.46
$\{E\} \Rightarrow \{F\}$	T_4, T_7, T_9	M,H	R	2	1	-	12	100.00
$\{D\} \Rightarrow \{F\}$	T_5, T_7, T_8	L	R	9	7	-	16	100.00
$\{B, C\} \Rightarrow \{E\}$	T_6	H	U	3	2	1	12	50.00
	T_{10}	H	R	4	1	1		50.00
$\{A, F\} \Rightarrow \{D\}$	T_7, T_8	L,M	R	5	5	6	16	100.00

The previous examples show that the complexity and completeness of the CHs is a primary factor determining the interestingness of the results. Also, if several CHs are available for the same attribute, which means knowledge about the attribute can be expressed in different ways, then many different summaries are possible.

8 Experimental Results

The primary distinction between the share-confidence framework and the support-confidence framework is that the former considers the quantity and value of the items purchased rather than simply the number of transactions which contain the item. We now present experimental results obtained using the *CI* algorithm which show that knowing the quantity and value of items can give informative feedback and insight about the relative importance of particular itemsets. We ran all of our experiments on an IBM AT-compatible personal computer, consisting of a Pentium P166 processor with 64 MB of memory running Windows NT Workstation version 4.0. Input data was from a database supplied by a commercial partner in the telecommunications industry. We ran the database under Oracle Release 7.3 and IRIX Release 5.3 on a Silicon Graphics Challenge L with 512 MB of memory and twelve 150 MHz MIPS R4400 CPUs. The database contained approximately 3.3 million tuples representing account activity for over 500 thousand customer accounts and 2200 unique items. Each tuple is either an equipment rental or service transaction containing the number of items and the cost of each item. An itemset was considered to be frequent if at least one of the following three conditions held:

1. The minimum support was greater then 0.25%.
2. The global share relative to the total item count (referred to in this discussion as *global share quantity* or *quantity*) was greater than 0.25%.
3. the global share relative to the total item amount (referred to in this discussion as *global share value* or *value*) was greater than 0.25%.

Figure 2: 20 most frequent 1-itemsets ranked by support

Figure 3 shows that 14 of the frequent 1-itemsets that were ranked highest by support (i.e., those identified by integers less than or equal to 20), also appear in the 20 most frequent 1-itemsets ranked by quantity. The remaining six 1-itemsets (i.e., 101, 81, 25, 107, 100, 34) are shown to have a higher ranking when ranked by quantity. The 1-itemsets that include items 100, 101, and 107 are especially noteworthy since there were only 109 frequent 1-itemsets ranked. The support measure considers these items to be among the least important, yet when ranked

Figure 4: 20 most frequent 1-itemsets ranked by global share value

Figures 3 and 4 show that twelve of the frequent 1-itemsets are common to the 20 most frequent 1-itemsets ranked by both quantity and value (i.e., 1-6, 8, 9, 34, 81, 100, 101). Thus, eight items that were highly ranked by value were not highly ranked by quantity.

Similar results to those shown in Figures 2 to 4 were obtained when ranking 3-, 4-, and 5-itemsets. We present the results for 2-itemsets, shown in Table 24. Table 24 shows three sets of rankings for 2-itemsets, where each set contains three columns. In Table 24, the *Support Rank* column in each set describes 20 itemsets ranked by support, the *Share Rank (Quantity)* column describes 20 itemsets ranked by quantity, and the *Share Rank (Value)* column describes 20 itemsets ranked by value. In the first set, the first column shows the 20 most frequent 2-itemsets ranked by support. The second and third columns show the corresponding rank for each itemset ranked by quantity and value, respectively. In the second set, the second column shows the 20 most frequent 2-itemsets ranked by quantity. The first and third columns show the corresponding rank for each itemset ranked by support and value, respectively. In the third set, the third column shows the 20 most frequent 2-itemsets ranked by value. The first and second columns show the corresponding rank for each itemset ranked by support and quantity. There were 351 frequent 2-itemsets.

Table 24: 2-itemsets ranked by support and share

Set 1			*Set 2*			*Set 3*		
Support Rank	*Share Rank (Quantity)*	*Share Rank (Value)*	*Support Rank*	*Share Rank (Quantity)*	*Share Rank (Value)*	*Support Rank*	*Share Rank (Quantity)*	*Share Rank (Value)*
1	**4**	**1**	**306**	**1**	**18**	1	4	1
2	13	3	341	2	38	293	8	2
3	17	9	324	3	27	2	13	3
4	19	12	1	4	1	305	45	4
5	20	5	294	5	23	5	20	5
6	22	11	316	6	32	288	121	6
7	27	28	291	7	24	75	80	7
8	35	33	293	8	2	287	206	8
9	47	59	307	9	29	3	17	9
10	**41**	**109**	301	10	31	**336**	**350**	**10**
11	49	41	339	11	22	6	22	11
12	51	37	328	12	69	4	19	12
13	54	42	2	13	3	318	37	13
14	55	58	343	14	77	304	348	14
15	62	97	349	15	76	300	248	15
16	61	90	340	16	67	314	108	16
17	67	101	3	17	9	337	31	17
18	68	132	323	18	44	306	1	18
19	69	98	4	19	12	138	57	19
20	65	154	5	20	5	54	60	20

The 2-itemset ranked as most frequent by support (refer to the first set) and value was ranked fourth by quantity. While this itemset does not represent the most frequent itemset sold in terms of the quantity of items, it was purchased in the greatest number of transactions and had the highest gross income of all 2-itemsets. In contrast, the 2-itemset ranked tenth by support, for instance, was ranked 41-st by quantity and 109-th by value. This itemset is ranked highly by support, yet its contribution to gross income is comparatively low.

The 2-itemset ranked as most frequent by quantity (refer to the second set) was ranked 306-th by support. This is an itemset where the items are typically purchased in multiples. Consequently, it is purchased more frequently than support seems to indicate. Similarly, 15 of the 20 most frequent 2-itemsets ranked highly by quantity are ranked below 291 by support.

The 2-itemset ranked tenth by value (refer to the third set) was ranked 336-th by support and 350-th by quantity. The items in this itemset are relatively expensive

items. Consequently, although not purchased as frequently as many other items, its contribution to gross income is comparatively high.

9 Conclusion and Future Research

We have introduced the share-confidence framework for knowledge discovery from databases. We defined practical itemset counting functions that measure both the quantity and value of the items in an itemset. These itemset counting functions were used as the basis for our notion of share. Share measures the contribution of an itemset relative to the total number of items sold or to the total value of items sold. We redefined the notion of frequent itemsets. A frequent itemset is an itemset where the quantity or value of the items in the itemset is greater than some user-specified minimum. We presented the *CI* algorithm which classifies itemsets based upon characteristic attributes extracted from customer, census, or lifestyle data. We suggested how characterized itemsets can be generalized according to concept hierarchies associated with the characteristics attributes. We showed how it is possible not only to discover the buying patterns of customers, but also to discover customer profiles by partitioning customers into distinct classes. Our experimental results demonstrated that the share-confidence framework can give more informative feedback than analysis based strictly upon support.

Future research will include an interface for navigating through the results generated by DB-Discover. For any combination of items, the interface will display the characteristics of the customers that purchase those items. Also, for any combination of customer characteristics, the interface will display any corresponding frequent itemsets.

Other future research will include rewriting DB-Discover to use domain generalization graphs [17, 21] as the primary data structure for guiding generalization, rather than concept hierarchies. A domain generalization graph defines a partial order which represents a set of generalization relations for a set of attributes. To generalize a set of attributes, the set can be considered a single attribute whose domain is the cross product of the individual attribute domains. A generalization from this domain is a combination of nodes, with one node from the DGG for each attribute.

Finally, DB-Discover will be enhanced to include the coincidence and dominance measures [14] and standard correlation measures [7] for itemsets.

References

[1] R. Agrawal, T. Imielinski, and A. Swami. Database mining: a performance perspective. *IEEE Transactions on Knowledge and Data Engineering*, 5(6):914–925, December 1993.

[2] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on the Management of Data (SIGMOD'93)*, pages 207–216, Washington, D.C., May 1993.

[3] R. Agrawal, H. Mannila, R.Srikant, H.Toivonen, and A.I. Verkamo. Fast discovery of association rules. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 307–328, Menlo Park, CA, 1996. AAAI Press/MIT Press.

[4] R. Agrawal and J.C. Schafer. Parallel mining of association rules. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):962–969, December 1996.

[5] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Databases (VLDB'94)*, pages 487–499, Santiago, Chile, September 1994.

[6] D.B. Barber and H.J. Hamilton. Comparison of attribute selection strategies for attribute-oriented generalization. In *Lecture Notes in Artificial Intelligence, The 11th International Symposium on Methodologies for Intelligent Systems (ISMIS'97)*, pages 106–116, Charlotte, North Carolina, 1997.

[7] S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: Generalizing association rules to correlations. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'97)*, pages 265–276, May 1997.

[8] S. Brin, R. Motwani, J.D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'97)*, pages 255–264, May 1997.

[9] Y. Cai, N. Cercone, and J. Han. Attribute-oriented induction in relational databases. In G. Piatetsky-Shapiro and W. Frawley, editors, *Knowledge Discovery in Databases*, pages 213–228, Cambridge, MA, 1991. AAAI/MIT Press.

[10] C.L. Carter and H.J. Hamilton. Efficient attribute-oriented algorithms for knowledge discovery from large databases. IEEE Transactions on Knowledge and Data Engineering. To appear.

[11] C.L. Carter and H.J. Hamilton. Fast, incremental generalization and regeneralization for knowledge discovery from databases. In *Proceedings of the 8th Florida Artificial Intelligence Symposium*, pages 319–323, Melbourne, Florida, April 1995.

[12] C.L. Carter and H.J. Hamilton. A fast, on-line generalization algorithm for knowledge discovery. *Applied Mathematics Letters*, 8(2):5–11, 1995.

[13] C.L. Carter and H.J. Hamilton. Performance evaluation of attribute-oriented algorithms for knowledge discovery from databases. In *Proceedings of the Seventh IEEE International Conference on Tools with Artificial Intelligence (ICTAI'95)*, pages 486–489, Washington, D.C., November 1995.

[14] C.L. Carter, H.J. Hamilton, and N. Cercone. Share-based measures for itemsets. In J. Komorowski and J. Zytkow, editors, *Proceedings of the First European Conference on the Principles of Data Mining and Knowledge Discovery (PKDD'96)*, pages 14–24, Trondheim, Norway, June 1997.

[15] D.W. Cheung, A.W. Fu, and J. Han. Knowledge discovery in databases: a rule-based attribute-oriented approach. In *Lecture Notes in Artificial Intelligence, The 8th International Symposium on Methodologies for Intelligent Systems (ISMIS'94)*, pages 164–173, Charlotte, North Carolina, 1994.

[16] T. Fukuda et al. Mining optimized association rules for numeric attributes. In *Proceedings of the Fifteenth ACM SIGACT-SIGMOD-SIGART Symposium on the Principles of Database Systems (PODS'96)*, pages 182–191, Montreal, Canada, June 1996.

[17] H.J. Hamilton, R.J. Hilderman, and N. Cercone. Attribute-oriented induction using domain generalization graphs. In *Proceedings of the Eighth IEEE International Conference on Tools with Artificial Intelligence (ICTAI'96)*, pages 246–253, Toulouse, France, November 1996.

[18] J. Han, Y. Cai, and N. Cercone. Knowledge discovery in databases: an attribute-oriented approach. In *Proceedings of the 18th International Conference on Very Large Data Bases*, pages 547–559, Vancouver, August 1992.

[19] J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. In *Proceedings of the 1995 International Conference on Very Large Data Bases (VLDB'95)*, pages 420–431, September 1995.

[20] J. Han and Y. Fu. Exploration of the power of attribute-oriented induction in data mining. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Adavances in Knowledge Discovery and Data Mining*, pages 399–421. AAAI/MIT Press, 1996.

[21] R.J. Hilderman, H.J. Hamilton, R.J. Kowalchuk, and N. Cercone. Parallel knowledge discovery using domain generalization graphs. In J. Komorowski and J. Zytkow, editors, *Proceedings of the First European Conference on the Principles of Data Mining and Knowledge Discovery (PKDD'96)*, pages 25–35, Trondheim, Norway, June 1997.

[22] M. Holsheimer, M. Kersten, H. Mannila, and H. Toivonen. A perspective on databases and data mining. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD'95)*, pages 150–155, August 1995.

[23] M. Houtsma and A. Swami. Set-oriented mining of association rules. In *Proceedings of the 11th International Conference on Data Engineering (ICDE'95)*, pages 25–34, 1995.

[24] H.-Y. Hwang and W.-C. Fu. Efficient algorithms for attribute-oriented induction. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD'95)*, pages 168–173, Montreal, August 1995.

[25] M. Klemettinen, H. Manilla, P. Ronkainen, H. Toivonen, and A.I. Verkamo. Finding interesting rules from large sets of discovered association rules. In *Proceedings of the 3rd International Conference on Information and Knowledge Management*, pages 401–407, 1994.

[26] H. Mannila, H. Toivonen, and A.I. Verkamo. Efficient algorithms for discovering association rules. In U.M. Fayyad and R. Uthurusamy, editors, *Proceedings of the 1994 AAAI Workshop on Knowledge Discovery in Databases*, pages 144–155, Seattle, Washington, July 1994.

[27] B. Masand and G. Piatetsky-Shapiro. A comparison of approaches for maximizing business payoff of prediction models. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*, pages 195–201, Portland, OR, August 1996.

[28] R.S. Michalski. A theory and methodology of inductive learning. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, pages 83–134. Tioga Publishing Company, 1983.

[29] J.S. Park, M.-S. Chen, and P.S. Yu. An effective hash-based algorithm for mining association rules. *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'95)*, pages 175–186, May 1995.

[30] A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. In *Proceedings of the 21th International Conference on Very Large Databases (VLDB'95)*, pages 432–444, Zurich, Switzerland, September 1995.

[31] R. Srikant and R. Agrawal. Mining generalized association rules. In *Proceedings of the 21th International Conference on Very Large Databases (VLDB'95)*, pages 407–419, Zurich, Switzerland, September 1995.

[32] R. Srikant and R. Agrawal. Mining quantitative association rules in large relational databases. In *Proceedings of the ACM SIGMOD Conference on Management of Data (SIGMOD'96)*, pages 1–12, Montreal, Canada, June 1996.

[33] H. Toivonen. Sampling large databases for finding association rules. In *Proceedings of the 22nd International Conference on Very Large Databases (VLDB'96)*, pages 134–145, Mumbay, India, September 1996.