Mining 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,carter,hamilton,nick}@cs.uregina.ca

Abstract. We propose the *share-confidence framework* for knowledge discovery from databases which addresses the problem of mining itemsets from market basket data. Our goal is two-fold: (1) to present new itemset measures which are practical and useful alternatives to the commonly used support measure; (2) 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 suggest 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.

1 Introduction

Consider a retail sales operation with a large inventory consisting of many distinct 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 determine 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* [2, 14]). 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 specific 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 neighborhoods. He believes that characterizing itemsets with classificatory information available from credit card or cheque transactions will allow him to answer queries of this kind.

In this paper, we propose the *share-confidence framework* that looks beyond the simple frequency with which two or more items are bought together. We introduce a new algorithm, called *CI*, which integrates the *Apriori* algorithm for discovering association rules between items in large databases [2, 1], and the *AOG* algorithm for attribute-oriented generalization in large databases [9, 11]. We also show how market basket data can be mined 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 Section 2, we present a formal description of the market basket analysis problem and introduce the share-confidence framework. In Section 3, we describe characterized itemsets and an algorithm for generating characterized itemsets from market basket data. In Section 4, we present experimental results obtained using the share-confidence framework on a database supplied by a commercial partner. We conclude in Section 5 with a summary of our work.

2 The Share-Confidence Framework

The problem of discovering association rules form market basket data has been formally defined as follows [2]. 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 the *support-confidence framework* [4].

The most studied and analyzed algorithm for generating itemsets in the support-confidence framework is *Apriori*, described in detail in [1, 2, 3]. 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* 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 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 know 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. Since

support does not consider quantity and value, its use is limited as a practical indicator for determining the financial implications of an itemset.

We will now extend the formalization of the market basket problem. 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*, introduced in [8] as *share measures*.

In the sections that follow, we define the functions upon which the share-confidence framework is based. For the examples, refer to the transaction database shown in Table 1 and the item database shown in Table 2. In Table 1, the *TID* column describes the transaction identifier and columns A to F describe the items (products) being sold. Note that binary values are not used to indicate the purchase of an item, instead the actual number of items purchased in the corresponding transaction (i.e., the counts) is used. In Table 2, the *Item* column describes the valid items and the *Retail Price* column describes the retailer's selling price to the customer.

Table 1. 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 2. The item database

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

2.1 Preliminary Definitions

The definitions in this section were 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 from databases [7, 6]. Definitions 1 to 6 are used to query summary views containing discovered frequent itemsets.

Definition 1. The *local itemset count* is the sum of the local item counts (i.e., the quantity of a particular item purchased in a particular transaction) for all

transactions which contain a particular item in a particular itemset, denoted as $lisc(i,x)$, where $lisc(i,x) = \sum lic(i,t_k)$, $lic(i,t)$ is the value at the intersection of row t and column i, $i \in I$, $x \subseteq I$, $x \in t_k$, 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 2. The *local itemset amount* is the sum of the local item amounts (i.e., the product of the local item count for a particular item purchased in a particular transaction and the item retail price) for all transactions which contain a particular item in a particular itemset, denoted as $lisa_1(i,x)$, where $lisa_1(i,x) = \sum lia(i,t_k)$, $lia(i,t)$ is the value at the intersection of row t and column i multiplied by the item retail price of item i, $i \in I$, $x \subseteq I$, $x \in t_k$, 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 $lisa_1(C,\{B,C\}) = lia(C,T_3) + lia(C,T_6) + lia(C,T_{10}) = 25.00$.

Definition 3. 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 4. 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 $gisa(\{B,C\}) = lisa_2(B,\{B,C\}) + lisa_2(B,\{B,C\}) = 43.00$.

Definition 5. The *total itemset count* is the sum of the global item counts (i.e., the sum of the local item counts for a particular item purchased in all transactions) for all items in a particular itemset, denoted as $tisc(x)$, where $tisc(x) = \sum gic(i_k)$, $gic(i)$ is the sum of all values in column i, $x \subseteq I$, 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 6. The *total itemset amount* is the sum of the global item amounts

(i.e., the sum of the local item amounts for a particular item purchased in all transactions) for all items in a particular itemset, denoted as $tisa(x)$, where $tisa(x) = \sum gia(i_k)$, $gia(i)$ is the value of item i in all transactions, $x \subseteq I$, 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 $tisa(\{B, C\} = gia(B) + gia(C) = 92.00$.

2.2 Share

We now introduce and define the notion of share in terms of the definitions from the previous section.

Definition 7. The *total item count local share* for a particular item in a particular itemset is the ratio of the local itemset count to the total item count (i.e., the sum of the global item counts for all items purchased in all transactions), expressed as a percentage, denoted as $ticls(i, x)$, where $ticls(i, x) = (lisc(i, x)/tic) * 100$, tic is the quantity of all items in the transaction database, $i \in I$, 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 total item count local share for item F in itemset $\{D, F\}$ is $ticls(F, \{D, F\}) = (lisc(F, \{D, F\})/tic) * 100 = 10.6\%$.

Definition 8. The *total item amount local share* for a particular item in a particular itemset is the ratio of the local itemset amount to the total item amount (i.e., the sum of the global item amounts for all items purchased in all transactions), expressed as a percentage, denoted as $tials(i, x)$, where $tials(i, x) = (lisa_v(i, x)/tia) * 100$, tia is the total value of all items in the transaction database, $i \in I$, $x \subseteq I$, 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 total item amount local share for item F in itemset $\{D, F\}$ is $tials(F, \{D, F\}) = (lisa_1(F, \{D, F\})/tia) * 100 = 15.9\%$.

Definition 9. The *total item count global share* for a particular itemset is the ratio of the global itemset count to the total item count, expressed as a percentage, denoted as $ticgs(x)$, where $ticgs(x) = (gisc(x)/tic) * 100$, $x \subseteq I$.
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 total item count global share for itemset $\{D, F\}$ is $tigcs(\{D, F\}) = (gisc(\{D, F\})/tic) * 100 = 24.2\%$.

Definition 10. The *total item amount global share* for a particular itemset is the ratio of the global itemset amount to the total item amount, expressed as a percentage, denoted as $tiags(x)$, where $tiags(x) = (gisa(x)/tia) * 100$, $x \subseteq I$.

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 transaction database."*
Result. The total item amount global share for itemset {D,F} is $tiags(\{D, F\}) = (gisa(\{D, F\})/tia) * 100 = 28.9\%$.

Definition 11. The *global itemset count local share* 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 $giscls(i, x)$, where $giscls(i, x) = (lisc(i, x)/gisc(x)) * 100$, $i \in I$, 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 global itemset count local share for item A in itemset $\{A, D\}$ is $giscls(A, \{A, D\}) = (lisc(A, \{A, D\})/gisc(\{A, D\})) * 100 = 46.2\%$.

Definition 12. The *global itemset amount local share* 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 $gisals(i, x)$, where $gisals(i, x) = (lisa_v(i, x)/gisa(x)) * 100$, $i \in I$, $x \subseteq I$, 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 global itemset amount local share for item A in itemset $\{A, D\}$ is $gisals(A, \{A, D\}) = (lisa_1(A, \{A, D\})/gisa(\{A, D\})) * 100 = 21.3\%$.

2.3 Frequent Itemsets

A frequent itemset was previously defined as an itemset whose support is greater than some user-specified minimum [2]. We now define frequent itemsets as used in the share-confidence framework.

Definition 13. 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 total item count local share is greater than some user-specified minimum. That is, $ticls(i_k, x) \geq minshare_1$, where $x \subseteq I$, $i_k \in x$, for some k, and $minshare_1$ is the user-specified minimum share.
2. The total item amount local share is greater than some user-specified minimum. That is, $tials(i_k, x) \geq minshare_2$, 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 3. In Table 3, the *Itemset* column describes the items in the itemset, the *TIDs* column describes the transaction identifiers that contain the corresponding itemset, the $ticls(i_1, x)$ and $ticls(i_2, x)$ columns describe the total item count local share for items one and two, respectively, and the $tials(i_1, x)$ and $tials(i_2, x)$ columns describe the total item amount local share for items one and two, respectively.

Table 3. Locally frequent 2-itemsets

$Itemset$	$TIDs$	$ticls(i_1, x)$ (%)	$ticls(i_2, x)$ (%)	$tials(i_1, x)$ (%)	$tials(i_2, x)$ (%)
$\{A, D\}$	T_1, T_7, T_8	**9.09**	**10.6**	2.73	**10.1**
$\{B, E\}$	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	**10.6**	**15.19**	**15.95**
$\{D, F\}$	T_5, T_7, T_8	**13.64**	**10.6**	**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 14. 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 4. The columns in Table 4 have the same meaning as in Table 3.

Table 4. Globally frequent 2-itemsets

$Itemset$	$TIDs$	$ticls(i_1, x)$ (%)	$ticls(i_2, x)$ (%)	$tials(i_1, x)$ (%)	$tials(i_2, x)$ (%)
$\{A, D\}$	T_1, T_7, T_8	**9.09**	**10.6**	2.73	10.1
$\{C, E\}$	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**	**10.6**	**12.98**	**15.95**

2.4 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 [2]. We now define confidence as used in the share-confidence framework.

Definition 15. 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 16. 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 $ac(\{B, C\}, \{B, C, E\}) = ((lisa_2(B, \{B, C, E\}) + lisa_2(C, \{B, C, E\}))/gisa(\{B, C\})) * 100 = 59.9\%$.

3 Characterized Itemsets

3.1 Example

We now present an example to demonstrate the *CI* algorithm and describe the primary data structures. In this example, let L_k^* and C_k^* denote the set of frequent itemsets from pass k and the set of candidate itemsets from pass k, respectively, and let R^* denote the relation containing the characterized itemsets. Each element of L_k^* and C_k^* contains three attributes: the itemset, the total item count local share, and the total item amount local share. Each element of R^* contains one attribute for each characteristic of interest and an attribute containing a list of all frequent itemsets sharing the corresponding characteristic attributes. Assume we are given the transaction database shown in Table 5. Also assume the user-specified minimum share is 15%. In Table 5, the column descriptions have the same meaning as the like-named columns in Table 1. Our task is to trace through the first three passes of *CI* to generate and store the characterized itemsets in R^*. For this example, we consider only the total item count local share to determine whether an itemset is frequent.

Table 5. 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 6 and 7, respectively. In Table 6, the *Itemset* column describes the items in each itemset and the *Share* column describes the total item count local share. In Table 7, the *Char. 1* and *Char. 2* columns describe the characteristics retrieved from the external database(s), 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 reader convenience). The domain of the first and second characteristic is $\{R, S\}$ and $\{X, Y, Z\}$, respectively.

After the second pass, *CI* generates L_2^* and updates R^* as shown in Tables 8 and 9, respectively. In Tables 8 and 9, the column descriptions have the same meaning as the like-named columns in Table 6 and Table 7, respectively. Also in

Table 6. Frequent itemsets contained in L_1^*

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

Table 7. 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

Table 9, the *Itemsets* column describes the frequent itemsets from the previous pass that share the identified characteristics.

Table 8. Frequent itemsets contained in L_2^*

Itemset	*Share* (%)
$\{A, C\}$	61.2
$\{A, D\}$	49.0
$\{A, E\}$	24.5

After the third pass, *CI* generates L_3^* and updates R^* as shown in Tables 10 and 11, respectively. In Tables 10 and 11, the column descriptions have the same meaning as the like-named columns in Table 8 and Table 9, respectively.

The characterized itemsets in R^*, generated by *CI*, form a relation. In a relation, transforming a specific data description into a more general one is called generalization. Several algorithms have been proposed for finding generalized itemsets where concept hierarchies are used to classify items [10, 1]. Our approach differs from these in that we use concept hierarchies to classify the characteristic attributes. Fast and efficient implementations of *AOG* [7, 6, 13] are used to generate summaries where the characteristic attributes are generalized according to the concept hierarchies. If the concept hierarchies have relatively few levels (i.e., fewer than 10), and if multiple hierarchies are available for some attributes, the *AllGen* algorithm [12] is used to generate all possible summaries.

3.2 The *CI* Algorithm

In the description of *CI* that follows, L_k^*, C_k^*, and R^* have the same meaning as in the example of the previous section. 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 $(k-1)$.

Table 9. 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	$\langle\{A\}, 3\rangle$, $\langle\{C\}, 2\rangle$, $\langle\{D\}, 1\rangle$

Table 10. Frequent itemsets contained in L_3^*

Itemset	*Share* (%)
$\{A, C, D\}$	69.4
$\{A, C, E\}$	34.7

2. Generate the candidate k-itemsets in C_k^* from the frequent $(k-1)$-itemsets in L_{k-1}^* using the *Apriori* method described in [2, 5].
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 corresponding 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.* step. 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.

Table 11. R^* after the third pass

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

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^*.

The running time and space requirements of *CI* are $O(|c| * |t|)$ and $O(|s|)$, respectively, where $|c|$ is the number of candidate itemsets in all iterations of the algorithm, $|t|$ is the number of transactions, and $|s|$ is the size of the largest candidate itemset in any pass.

4 Experimental Results

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 large database supplied by a commercial partner in the telecommunications industry. The database contained approximately 3.3 million tuples representing account activity for over 500 thousand customer accounts and 2200 unique items (identified by integers in the range $[1 \ldots 2200]$). 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: the minimum support was greater then 0.25%, the total item count global share was greater than 0.25%, or the total item amount global share was greater than 0.25%.

The 20 most frequent 1-itemsets ranked by support, total item count global share, and total item amount global share are shown in Figures 1, 2, and 3, respectively. In Figures 1 to 3, the first row of bars (i.e., those at the front of

Fig. 1. 20 most frequent 1-itemsets ranked by support

Figure 2 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 total item count global share. The remaining six 1-itemsets (i.e., 101, 81, 25, 107, 100, 34) are shown to have a higher ranking when ranked by total item count global share. 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 by total item count global share, they are ranked eleventh, first, and eighth, respectively.

Figure 3 shows that nine of the frequent 1-itemsets that were ranked highest by support, also appear in the 20 most frequent 1-itemsets ranked by total item amount global share. It also shows that nine of the most frequent 1-itemsets which were ranked in the bottom 50% by support, are shown to be among the 20 most frequent when ranked by total item amount global share.

Similar results to those shown in Figures 1 to 3 were obtained when ranking k-itemsets. We present the results for 2-itemsets, shown in Table 12. Table 12 shows three sets of rankings for 2-itemsets, where each set contains three columns. In Table 12, the *Support*, *Share (Quantity)*, and *Share (Value)* columns

Fig. 3. 20 most frequent 1-itemsets ranked by total item amount global share

describe 10 itemsets ranked by support, total item count global share, and total item amount global share, respectively. In the first set, the first column shows the 10 most frequent 2-itemsets ranked by support. The second and third columns show the corresponding rank for these itemsets ranked by total item count and total item amount global share, respectively. In the second set, the second column shows the 10 most frequent 2-itemsets ranked by total item count global share. The first and third columns show the corresponding rank for these itemsets ranked by support and total item amount global share, respectively. In the third set, the third column shows the 10 most frequent 2-itemsets ranked by total item amount global share. The first and second columns show the corresponding rank for these itemsets ranked by support and total item count global share. There were 351 frequent 2-itemsets.

The 2-itemset ranked as most frequent by support (refer to the first set) and total item amount global share was ranked fourth by total item count global share. 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

Table 12. 2-itemsets ranked by support and share

Set 1 Rankings			*Set 2 Rankings*			*Set 3 Rankings*		
Support	*Share (Quantity)*	*Share (Value)*	*Support*	*Share (Quantity)*	*Share (Value)*	*Support*	*Share (Quantity)*	*Share (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**

2-itemset ranked tenth by support, for instance, was ranked 41-st by total item count global share and 109-th by total item amount global share. This itemset is ranked highly by support, yet its contribution to gross income is comparatively low.

The 2-itemset ranked as most frequent by total item count global share (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, 13 of the 15 most frequent 2-itemsets ranked highly by total item count global share are ranked below 291 by support.

The 2-itemset ranked tenth by total item amount global share (refer to the third set) was ranked 336-th by support and 350-th by total item count global share. 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.

5 Conclusion

We have introduced the share-confidence framework for knowledge discovery from databases which classifies itemsets based upon characteristic attributes extracted from external databases. We suggested how characterized itemsets can be generalized according to concept hierarchies associated with the characteristic attributes. Experimental results demonstrated that the share-confidence framework can give more informative feedback than the support-confidence framework.

References

1. R. Agrawal, K. Lin, H.S. Sawhney, and K. Shim. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *Proceedings of the 21th International Conference on Very Large Databases (VLDB'95)*, Zurich, Switzerland, September 1995.

2. 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.
3. R. Agrawal and J.C. Schafer. Parallel mining of association rules. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):962–969, December 1996.
4. 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.
5. 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.
6. 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.
7. 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.
8. 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'97)*, pages 14–24, Trondheim, Norway, June 1997.
9. 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.
10. 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.
11. 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.
12. 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'97)*, pages 25–35, Trondheim, Norway, June 1997.
13. 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.
14. 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.