A New Framework For Itemset Generation

Charu C. Aggarwal and Philip S. Yu

IBM T.J. Watson Research Center
Yorktown Heights, NY 10598

Abstract

The problem of finding association rules in a large database of sales transactions has been widely studied in the literature. We discuss some of the weaknesses of the large itemset method for association rule generation. A different method for evaluating and finding itemsets referred to as *strongly collective itemsets* is proposed. The concepts of "support" of an itemset and correlation of the items within an itemset are related, though not quite the same. This criterion stresses the importance of the actual correlation of the items with one another rather than the absolute support. Previously proposed methods to provide correlated itemsets are not necessarily applicable to very large databases. We provide an algorithm which provides very good computational efficiency, while maintaining statistical robustness. The fact that this algorithm relies on *relative* measures rather than absolute measures such as support also implies that the method can be applied to find association rules in datasets in which items may appear in a sizeable percentage of the transactions (dense datasets), datasets in which the items have varying density, or even negative association rules.

1 Introduction

Association rules find the relationships between the different items in a database of sales transactions. Such rules track the buying patterns in consumer behavior eg. finding how the presence of one item in the transaction affects the presence of another and so forth. The problem of association rule generation has recently gained considerable prominence in the data mining community because of the capability of its being used as an important tool for knowledge discovery. Consequently, there has been a spurt of research activity in the recent years surrounding this problem.

Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of items. Each transaction T is a set of items, such that $T \subseteq I$. This corresponds to the set of items which a consumer may buy in a basket transaction.

An *association rule* is a condition of the form $X \Rightarrow Y$ where $X \subseteq I$ and $Y \subseteq I$ are two sets of items. The idea of an association rule is to develop a systematic method by which a user can figure out how to infer the presence of some sets of items, given the presence of other items in a transaction. Such information is useful in making decisions such as customer targeting, shelving, and sales promotions.

An important approach to the association rule problem was developed by Agrawal et. al. in [2]. This is a two-phase large itemset approach. Certain terminologies were defined in the same work which formalize these notions. In order to discuss the method further, we shall review some important definitions.

The *support* of a rule $X \Rightarrow Y$ is the fraction of transactions which contain both X and Y.

The *confidence* of a rule $X \Rightarrow Y$ is the fraction of transactions containing X, which also contain Y. Thus, if we say that a rule has 90% confidence then it means that 90% of the tuples containing X also contain Y.

The large itemset approach is as follows. Generate all combinations of items that have fractional transaction support above a certain user-defined threshold called *minsupport*. We call all such combinations *large itemsets*. Given an itemset $S = \{i_1, i_2, \ldots, i_k\}$, we can use it to generate at most k rules of the type $[S - \{i_r\}] \Rightarrow i_r$ for each $r \in \{1, \ldots, k\}$. Once these rules have been generated, only those rules above a certain user defined threshold called *minconfidence* may be retained.

Initially, the method was proposed only for the case of transactional data. A lot of further research has been devoted to speeding up the algorithm and extending the approach to other scenarios [3, 8, 11, 12]. An up-to-date survey on some of the work done in data mining may be found in [6].

1.1 Contributions of this paper

The main contributions of this work are as follows:

(1) We propose an alternative model (called "strongly collective itemset model") to the large itemset method for association rule generation.

(2) The large itemset technique is specially designed for sales transaction data which tends to be *sparse*. In other words, a given item appears only in a small fraction of the transactions. It is useful to study the performance of the large itemset method when the data is different from the typical binary sales transaction sets.[1]. We believe that the applicability of the con-

[1] An example is that of finding quantitative association rules by discretizing the quantitative attributes and binarizing the categorical

PODS 98 Seattle WA USA

X	1	1	1	1	0	0	0	0
Y	1	1	0	0	0	0	0	0
Z	0	1	1	1	1	1	1	1

Table 1: The base data

Rule	Support	Confidence
$X \Rightarrow Y$	25%	50%
$X \Rightarrow Z$	37.5%	75%

Table 2: Corresponding support and confidence

ventional technique to many kinds of data (such as discretized quantitative or categorical data) is not immediate. We provide a detailed criticism of the large itemset model .

(3) Some of the weaknesses of the support-confidence framework for association rule generation have been observed earlier by Brin et. al. [4]. A more accurate measure called the χ^2 measure was proposed in order to evaluate the quality of a itemset. However, this method, though statistically precise, is a little expensive for large databases. As we shall see, the strongly collective itemset generation model is applicable to very large databases. In fact, the method requires just one pass over the entire transaction database in order to closely approximate the strongly collective itemsets. If required, an additional pass can be performed in order to find the exact set.

2 A criticism of the large itemset method

Some of the criticisms associated with the traditional large itemset method are the following:

- **Spuriousness in itemset generation:** We shall explain this issue with the help of an example. Consider a retailer of breakfast cereal which surveys 5000 students on the activities they engage in the morning. The data shows that 3000 students play basketball, 3750 eat cereal, and 2000 students both play basketball and eat cereal. For a minimum support of 40%, and minimum confidence of 60%, we find the following association rule:
  play basketball $\Rightarrow$ *eat cereal*
  The association rule is misleading because the overall percentage of students eating cereal is 75% which is even larger than 60%. Thus, playing basketball and eating cereals are negatively associated: being involved in one decreases the chances of being involved in the other. Consider the following association:
  play basketball $\Rightarrow$ *(not) eat cereal*
  Although this rule has both lower support and lower confidence than the rule implying positive association, it is far more accurate. Thus, if we set the support and confidence sufficiently low, two contradictory rules would be generated; on the other hand if we set the parameters sufficiently high only the inaccurate rule would be generated. In other words, no combination of support and confidence can generate purely the correct association.

  Another example is illustrated in Table 1, in which we have three items X, Y, and Z. The coefficient of correlation between X and Y is 0.577, while that between X and Z is -0.378. Thus, X and Y are positively related to each other, while X and Z are negatively related. The support for the rules $X \Rightarrow Y$ and $X \Rightarrow Z$ are illustrated in Table 2. Interestingly, the support and confidence for the rule $X \Rightarrow Z$ strictly dominates the support and confidence for the rule $X \Rightarrow Y$. Thus, it is not possible to find any level of *minsupport* and *minconfidence* at which only the rule $X \Rightarrow Y$ would be generated, without a spurious rule $X \Rightarrow Z$ being generated as well. If we set *minsupport* and *minconfidence* too low, then in many cases (especially when different items have widely varying global density) an unacceptably large number of rules might be generated, (a great majority of which may be spurious) and this defeats the purpose of data mining in the first place.

- **Dealing with dense data sets:** For a k-dimensional database, there are 2^k possibilities for itemsets. Some data sets may be such that a large number of these 2^k possibilities may qualify above the minimum support. For such situations, it may be necessary to set the *minsupport* to an unacceptably high level. This may result in a number of important rules being lost. Often, the value of *minsupport* is decided based on computational constraints or in restricting the number of rules generated to a manageable number. Thus, it is impossible to ascertain when important rules are being lost and when they are not.

  Some important applications in which the data may be dense are the following:

  - **Negative association rules:** Suppose we wish to find negative correlations among items. In other words, we wish to mine association rules in which presence as well as absence of an item may be used. Although the large itemset approach can be directly extended to this problem by treating the absence of an item as a pseudo-item, the sparsity of item presence in real transaction data may result in considerable bias towards rules which are concerned only with finding rules corresponding to absence of items rather than their presence.
  - **Data in which attributes have widely varying densities:** Suppose that the categorical data corresponding to customer profile information is available along with the sales transaction data and we wish to find association rules concerning the demographic nature of people buying different items. In this case, directly applying the itemset method may be very difficult because of the different densities of the demographic and the sales transaction information. An example of such a demographic attribute is sex, which may take on either value 50% of the time. On the other hand a bit representing an item may take on the value of 1 only 5% of the time.

2.1 Does the use of an interest measure help?

The use of an interest measure has been presented as a solution in order to avoid spurious association rules. The interest level of a rule is the ratio of the actual strength to

attributes

the expected strength based upon the assumption of statistical independence. Past work has concentrated on using the interest measure as a pruning tool in order to remove the uninteresting rules in the output. However, (as the basketball-cereal example illustrates) as long as support is still the primary determining factor in the initial itemset generation, either the user has to set the initial support parameter low enough so as to not lose any interesting rules in the output or risk losing some important rules. In the former case, computational efficiency may be a problem, while the latter case has the problem of not being able to retain rules which may be interesting from the point of view of a user.

There is some other work which deals with providing alternative methods for viewing itemsets. Correlation rules have been discussed by Brin et. al. in [4]. In [5], the notion of developing implication rules instead of association rules has been discussed. The implication strength of a rule is a number between 0 and ∞. An implication strength of 1 indicates that the strength of the rule is exactly as it would be under the assumption of statistical independence. An implication strength which is larger than 1 indicates greater than expected presence. This measure is preferable to confidence because it deals with greater than expected measures for finding association rules. We shall define a measure for itemsets called the collective strength, which is somewhat similar in spirit to the notion of implication strength because it compares between actual and expected occurences. We show how the fundamental itemset generation algorithms discussed in [3] may be replaced by new algorithms so as to work with this new definition of how a itemset is defined.

3 The notion of strongly collective itemsets

In this section, we shall discuss how to redefine the notion of support in order to develop a itemset generation algorithm. One way of defining a itemset as strongly present would be to consider an "interest measure" discussed in Agrawal et. al. [12]. Thus we may wish to generate all itemsets whose support is at least r-times the expected support based on the assumption of statistical independence. Unfortunately, the use of the interest measure has the following problem.

Consider a situation, where two attributes have perfectly positive correlation. In other words, one item is bought only when the other is bought, and vice-versa. Let us also consider the case when this pair of attributes assume the value of one 90% of the time. In this case, the interest measure for this pair of items as an itemset is $0.9/(0.9 * 0.9) = 1.11$. This is only marginally above the break-even value of 1, even though the two attributes are perfectly correlated. A similar issue has been raised by Brin et. al. in [5] with regard to the use of the interest measure in judging the confidence value of a rule.

We shall first define a notion called *collective strength* of an itemset. The collective strength of an itemset is defined to be a number between 0 to ∞. A value of 0 indicates perfect negative correlation, while a value of ∞ indicates perfectly positive correlation. A value of 1 indicates the "break-even point", corresponding to an itemset present at expected value.

An itemset I is said to be in *violation* of a transaction, if some of the items are present in the transaction, and others are not. Thus, the concept of violation denotes how many times a customer may buy at least some of the items in the itemset, but may not buy the rest of the items.

The *violation rate* of an itemset I is denoted by $v(I)$ and is the fraction of violations of the itemset I over all transactions. This is also equal to the fraction of transactions which contain a proper non-null subset of I.

The *collective strength* of an itemset I is denoted by $C(I)$ and is defined as follows:

$$C(I) = \frac{1 - v(I)}{1 - E[v(I)]} \cdot \frac{E[v(I)]}{v(I)} \quad (1)$$

The expected value of of $v(I)$ is calculated assuming statistical independence. Let us pause here to understand the meaning of collective strength before proceeding further. We note that the violation of an itemset in a transaction is a "bad event" from the perspective of trying to establish high correlation among the corresponding items. Thus $v(I)$ is the fraction of bad events while $(1 - v(I))$ is the fraction of "good events". The above definition for collective strength can be recast as follows:

$$C(I) = \frac{\text{Good Events}}{\text{E[Good Events]}} \cdot \frac{\text{E[Bad Events]}}{\text{Bad Events}} \quad (2)$$

The following are some of the interesting properties of collective strength.

(1) The notion of collective strength treats the 0 and 1 attributes in a symmetric way. Thus, if we were to apply this to a problem in which the absence as well as the presence of an item is used to find the itemsets, then we can immediately perceive the benefits of this definition.

(2) It is instructive to examine how this definition for collective strength fares for the case of a 2-itemset $I = \{i_1, i_2\}$. If i_1 and i_2 items are perfectly positively correlated, then the collective strength for the corresponding 2-itemset is ∞. This is because in this case, the fraction of occurences of bad events is 0. On the other hand, when the items are perfectly negatively correlated, the fraction of good events is 0, and hence the collective strength for the corresponding itemset pair is 0. For items which are independent from one another, the collective strength is 1. We provide this example in light of the criticisms that we expounded for the use of the interest measure at the beginning of the section.

(3) The collective strength uses the *relative* number of times an itemset is present in the database. The itemsets which have an insignificant presence can always be pruned off at a later stage. The level of presence of an itemset I which constitutes useful information is not exactly the same problem as finding whether or not the items in I are related to each other.

Now, we shall proceed to describe the notion of a *strongly-collective itemset*, and then we shall discuss some algorithms for finding strongly collective itemsets. It is important to note that the notion of collective strength is designed to see how the value of one attribute affects the value of another.

Definition 1 *An itemset I is denoted to be strongly collective at level K, if it satisfies the following properties:*

(1) The collective strength $C(I)$ of the itemset I is at least K.

(2) **Closure property:** *The collective strength $C(J)$ of every subset J of I is at least K.*

Itemset	Support	Collective Strength
Play basketball, eat cereal	40%	0.67
Play basketball, (not)eat cereal	20%	1/0.67 = 1.49

Table 3: Comparing collective strengths with support

Itemset	Support	Correlation	Collective Str.
X, Y	25%	0.577	3
X, Z	37.5%	−0.378	0.6
Y, Z	12.5%	−0.655	0.31

Table 4:

It is necessary to force the closure property in order to ensure that unrelated items may not be present in an itemset. Consider, for example, the case when itemset I_1 is {Milk, Bread} and itemset I_2 is {Diaper, Beer}. If I_1 and I_2 each have high collective strength, then it may often be the case that the itemset $I_1 \cup I_2$ may also have a high collective strength, even though items such as milk and beer may be independent.

The notion of collective itemsets satisfies the *bit symmetric* property. Let $\overline{X_i}$ denote the negation of the attribute X_i.

Theorem 1 *The collective strength of itemset $I = \{X_1 \ldots X_k\}$ is the same as that of the collective strength of the itemset $\overline{I} = \{\overline{X_1}, \ldots, \overline{X_k}\}$.*

Proof: The collective strength of an itemset I is a function of $v(I)$. The result follows directly from the fact that $v(I) = v(\overline{I})$.

■

In other words, the notion is symmetric over the complement property. A statistical analogue would be the coefficient of correlation between an item pair, which is also symmetric over the complement property.

3.1 A few examples

It is instructive to measure how the notion of collective strength fares for the examples we discussed in our criticisms of the large itemset method. For the basketball-cereal example, we see that the collective strength follows a different pattern from support. We already know that eating cereal is negatively correlated with playing basketball. Correspondingly, as illustrated in Table 3, the collective strength of the itemset (play basketball, eat cereal) is less than 1. This example thus illustrates the better applicability of this method to situations in which we wish to find negative association rules.

Similarly, the collective strength for the 2-itemsets corresponding to the data in the Table 1 is illustrated in Table 4. It is easy to see that the notion of collective strength is more closely related to the statistical coefficient of correlation than the support value.

4 Generating the strongly collective itemsets

Most algorithms which have been developed in past work for large itemset generation can be modified in order to develop algorithms for finding strongly collective itemsets. We first discuss how the value $E[v(I)]$ for an itemset I is calculated.

Let I consist of the set of items $\{i_1, i_2, \ldots i_k\}$. Let p_r denote the frequency of the occurence of the item i_r in the transaction data base. Then, for a given transaction, the probability that the itemset I occurs in a transaction is $\Pi_{r=1}^{k} p_r$. The probability that *none* of the items occur in the transaction is $\Pi_{r=1}^{k}(1-p_r)$. Thus the expected fraction of transactions in which at least one of the items of I occurs in the transaction and at least one does not is given by $1 - \Pi_{r=1}^{k} p_r - \Pi_{r=1}^{k}(1-p_r)$.

The algorithm in Agrawal et. al. [3] can be used to find the strongly collective itemsets with the modification that the notion of collective strength instead of support may be used in order to evaluate the itemsets. An itemset is retained if and only if it is strongly collective at level K. However, in this case it is possible to develop an approximate algorithm which is far more efficient. We shall see that it is usually not necessary to rescan the database beyond the 2-itemset phase. Such an algorithm is based on the following conjecture:

Conjecture: *Let k_0 be a number larger than 1. Consider an itemset B of size $n \geq 2$. Suppose that all 2-subsets of B have collective strength larger than k_0. Then the itemset B is highly likely to have collective strength larger than k_0.*

We will provide some intuition as to why the conjecture discussed above should hold. We define the *violation ratio* as the ratio of the number of violations to the expected number of violations. In other words:

$$\text{Violation Ratio} = \frac{v(I)}{E[v(I)]} \tag{3}$$

We define the agreement ratio as the ratio of the number of non-violations to the expected number of non-violations. Thus, we have:

$$\text{Agreement Ratio} = \frac{1 - v(I)}{1 - E[v(I)]} \tag{4}$$

Thus the collective strength is equal to the agreement ratio divided by the violation ratio. We shall now prove some results with regard to the agreement ratio and the violation ratio.

Theorem 2 *Let $I = \{i_1, i_2, i_3\}$ be a 3-itemset. For every 2-subset J of I, let it be the case that the agreement ratio is at least $\alpha > 1$. Then, it must also be the case that the agreement ratio of I is at least α.*

Proof: Let i_1, i_2, i_3 be the set of items in I. Let p_1, p_2, and p_3 be the frequencies (in fractions) of occurences of these items over the entire transaction database. Then we have:

$$\text{Expected Fraction of Agreements} = 1 - E[v(I)] \tag{5}$$

$$= p_1 \cdot p_2 \cdot p_3 + (1-p_1) \cdot (1-p_2) \cdot (1-p_3) \tag{6}$$

$$= 1 - p_1 - p_2 - p_3 + p_1 \cdot p_2 + p_1 \cdot p_3 + p_2 \cdot p_3 \tag{7}$$

Since i_1, i_2, and i_3 are 0-1 attributes, it must be the case that at least 2 of these 3 attributes must be in agreement. Let us classify the attributes into three types:

Type	Description	Fraction
1	All three attributes have the same value	f_1
2	Only attributes 1 and 2 have same value	f_2
3	Only attributes 1 and 3 have same value	f_3
4	Only attributes 2 and 3 have same value	f_4

Thus, we must have $f_1 + f_2 + f_3 + f_4 = 1$. From the fact that all two itemsets have agreement ratio at least α, we deduce the following:

$$f_1 + f_2 \geq \alpha \cdot (p_1 \cdot p_2 + (1-p_1) \cdot (1-p_2))$$
$$f_1 + f_3 \geq \alpha \cdot (p_1 \cdot p_3 + (1-p_1) \cdot (1-p_3))$$
$$f_1 + f_4 \geq \alpha \cdot (p_2 \cdot p_3 + (1-p_2) \cdot (1-p_3))$$

Adding the equations above, we get:

$$2f_1 + (f_1 + f_2 + f_3 + f_4) \geq \alpha[3 - 2 \cdot (p_1 + p_2 + p_3) + 2 \cdot (p_1 \cdot p_2 + p_1 \cdot p_3 + p_2 \cdot p_3)]$$

Using the fact that $f_1+f_2+f_3+f_4 = 1$, we get the following:

$$f_1 \geq (\alpha-1)/2+\alpha[1-p_1-p_2-p_3+p_1 \cdot p_2+p_1 \cdot p_3+p_2 \cdot p_3] \quad (8)$$

Since $\alpha \geq 1$, it follows that:

$$f_1 \geq \alpha \cdot (1 - p_1 - p_2 - p_3 + p_1 \cdot p_2 + p_1 \cdot p_3 + p_2 \cdot p_3) \quad (9)$$

Using Equations 7 and 9, we get:

$$f_1 \geq \alpha \cdot (1 - E[v(I)])$$
$$\Rightarrow 1 - v(I) \geq \alpha(1 - E[v(I)])$$

Thus, the result. ■

A similar result is true for the violation ratio as well.

Theorem 3 *Let $I = \{i_1, i_2, i_3\}$ be a 3-itemset. For every 2-subset J of I, let it be the case that the violation ratio is at most $\beta < 1$. Then, it must also be the case that the violation ratio of I is at most β.*

Proof: This proof is exactly similar to that of the previous theorem (with the same set of cases).

■

The collective strength of an itemset I can be expressed as α/β, where α is the agreement ratio, and β is the violation ratio. Further, when the collective strength is larger than 1, then it must be the case that $\alpha > 1$, and $\beta < 1$. The above results show that the agreement ratio of an 3-itemset is at least as large as the minimum of the agreement ratios of the subsets of that itemset. It is an open question as to whether the above results hold for k-itemsets, where is k is arbitrary. The results also show that the violation ratio of a 3-itemset is at most as large as the violation ratio of the 2-subsets of the itemset. This provides intuition as to why the results often hold for the case of collective strength as well. We will also support it with hard empirical results in the last section. We know from the closure property in the definition of a strongly collective itemset that if every $n-1$ subset of an itemset B is strongly collective at level k_0, then every 2-subset of the itemset B must also be strongly collective at the level k_0. This means that a join can be performed on the $(n-1)$-itemsets in order to generate a very close (superset) approximation of the n-itemsets.

This leads to a simple algorithm for itemset generation. We first explicitly generate all 2-itemsets. Then, for $j \geq 2$, we use repeated joins on j-itemsets to find $(j+1)$-itemsets, until no more itemset can be generated. A $(j+1)$-itemset is generated only if all j-subsets of it were also generated. Because of the conjecture above, this will result in a very close (superset) approximation of the itemsets. If necessary, an optional pass can be performed over the transaction database in order to prune those itemsets which were spuriously generated.

```
Algorithm GenerateItemsets(CollectiveStrength: k0);
begin
  Generate all 2-itemsets at collective strength k0;
  j = 2;
  Bj = Set of all j-itemsets with collective strength at least k0;
  while Bj ≠ φ do
  begin
    Perform a join on Bj to generate Bj+1;
    for each itemset I ∈ Bj+1 do
    begin
      for each j-subset I' of I do
      if I' ∉ Bj then prune I from Bj+1
    end;
    j = j + 1;
  end;
  O = ∪_{i=2}^{j} Bi;
  { O is the approximate output}
  Perform (an optional) pass over the transaction database to
          prune any false itemsets in O
end
```

Figure 1: The itemset generation algorithm

Note that this algorithm requires only to scan the database to generate the 2-itemsets. After that generation of all other itemsets is automatic, and it is no longer necessary to look at the transaction database again.

The itemset generation algorithm is illustrated in Figure 1. In the next section, we shall also discuss the quality of the itemsets generated, and the validity of the empirical observation upon which the correctness of the above algorithm depends. It is interesting to see that the above algorithm depends upon just a single pass over the entire transaction database. This results in considerable computational improvements for the itemset generation algorithm.

4.1 Using support as a constraint in the algorithm

The importance of support cannot be ignored. For example, itemsets with low support values but high collective strength may correspond to occurences in the database which do not have sufficient volume to convey any interesting information to a decision maker. However, this is an orthogonal issue to whether or not itemsets are closely correlated to each other. It is easy enough to add support as a constraint in the algorithm by a final postprocessing phase in which itemsets with low support are removed. Note that in this case, the user's choice of minimum support is decided by an opinion as to what constitutes high enough presence rather than computational considerations. This may not seem like a big advantage in typical market transaction data, but in dense datasets, or in artificial binary datasets created by discretizing quantitative data, this feature may offer a substantial advantage.

5 Empirical Results

The synthetic data sets were generated using a method similar to that discussed in Agrawal et. al. [3]. However, we added in an extra set of items in order to test how the data behaved when some of the items in the transaction data were of high density. Generating the data sets was a three stage process:

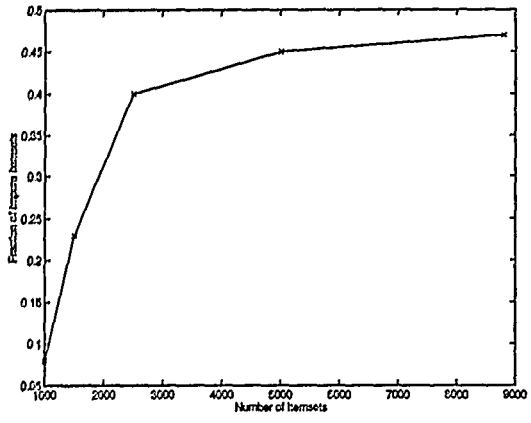

Figure 2: Variation of impurity with support

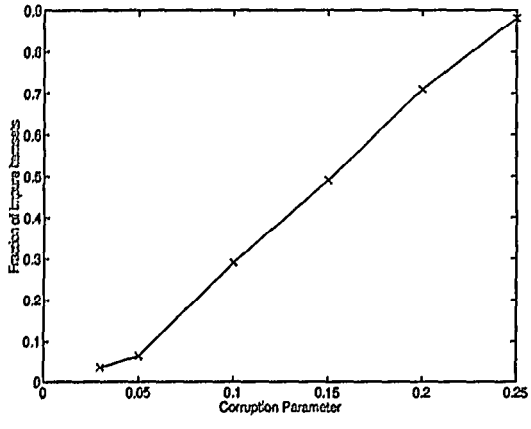

Figure 3: Variation of impurity with corruption level

(1) **Generating maximal potentially large itemsets:** The first step was to generate $L = 2000$ maximal "potentially large itemsets". These potentially large itemsets capture the consumer tendencies of buying certain items together. We first picked the size of a maximal potentially large itemset as a random variable from a poisson distribution with mean μ_L. Each successive itemset was generated by picking half of its items from the current itemset, and generating the other half randomly. This method ensures that large itemsets often have common items. Each itemset I has a weight w_I associated with it, which is chosen from an exponential distribution with unit mean.

(2) **Generating the transaction data:** The large itemsets were then used in order to generate the transaction data. First, the size S_T of a transaction was chosen as a poisson random variable with mean μ_T. Each transaction was generated by assigning maximal potentially large itemsets to it in succession. The itemset to be assigned to a transaction was chosen by rolling an L sided weighted die depending upon the weight w_I assigned to the corresponding itemset I. If an itemset did not fit exactly, it was assigned to the current transaction half the time, and moved to the next transaction the rest of the time. In order to capture the fact that customers may not often buy all the items in a potentially large itemset together, we added some noise to the process by corrupting some of the added itemsets. For each itemset I, we decide a noise level $n_I \in (0, 1)$. We generated a geometric random variable G with parameter n_I. While adding a potentially large itemset to a transaction, we dropped $\min\{G, |I|\}$ random items from the transaction. The noise level n_I for each itemset I was chosen from a normal distribution with mean 0.5 and variance 0.1.

(3) We consider an extra set of K "corrupt" items. The purpose of adding these corrupt items is to introduce some noise in the transaction data, and also create variation the density of the different items. Each of these K corrupted items may occur independently in a given transaction with probability of p_c. This probability p_c is denoted as the *corruption probability*. Thus, these items are statistically unrelated to any other item in the transaction data. Ideally, such an item ought not occur in an itemset of related items. For the purpose of this data, we always use $K = 10$, while p_c may vary between 0 and 0.5.

The data set which we used had $\mu_T = 10$, $\mu_L = 6$, and $100K$ transactions. We denote this data set by T'10.I6.D100K.[2]

We define an itemset to be *impure* if it contains at least one corrupt item. Our benchmark for comparing the two methods of itemset generation would be to find the percentage of itemsets generated which are impure. The percentage of impure itemsets generated is indicative of how the presence of a few independent items of high density may affect the generated itemsets.

In Figure 3, we show how the percentage of itemsets which are impure vary with the corruption level. In this case, we are using the conventional large itemset method [2] in order to generate the itemsets. For each corruption level p_c, we generate the 10000 itemsets with highest support

[2]Note that the data set T'10.I6.D100K is not exactly the same as the data set T10.I6.D100K in Agrawal et. al. because of the presence of the additional corrupt items.

Collective Strength	With final pass	Without final pass
1.19	1053	1053
1.1	2200	2200
1.05	11464	11464
1.033	56908	56908

Table 5: Number of itemsets found by *GenerateItemsets*

value. It is not surprising to see that the number of impure itemsets increases rapidly with the corruption level. In fact, at a corruption level of $p_c = 0.25$, we see that the number of itemsets which are impure is almost 90% of the total number of itemsets generated. Even at low corruption levels of $p_c = 0.03$, we see that 4% of the itemsets are impure.

In Figure 2, we have illustrated the variation of the impurity with the number of itemsets for a fixed corruption level p_c of 0.15. As we see, the level of impurity increases with the number of itemsets generated.

We also generated the same number of itemsets with the highest strongly collective strength using the algorithm *GenerateItemsets*. We observed the following:

The collective itemset generation algorithm never generated even a single impure itemset at any value of the corruption level p_c.

Another aspect which we would like to empirical substantiate is the conjecture upon which the optionality of the second pass in the algorithm *GenerateItemsets* relies. In the Table 5, we have illustrated the level of inaccuracy created by making this empirical approximation in the algorithm. As we see, the number of itemsets generated is exactly the same whether or not the (optional) second pass over the transaction database is made. Thus, to all intents and purposes, the algorithm requires just one pass in order to find the strongly collective itemsets at any level.

6 Conclusions and Summary

This paper provided an alternative to the large itemset model. We illustrated the greater robustness and accuracy of baskets found by using the strongly collective model. The closure properties of this method also helped in restricting the method to just one pass over the entire transaction database. In future research, we intend to explore methods for improving the computational complexity of the basic algorithm for generating collective baskets. We will also study how methods discussed in [1] may be used in order to provide online generation of the rules.

References

[1] Aggarwal C. C., and Yu P. S. Online Generation of Association Rules. *Proceedings of the Fourteenth International Conference on Data Engineering*, pages 402-411, Orlando, Florida, February 1998.

[2] Agrawal R., Imielinski T., and Swami A. Mining association rules between sets of items in very large databases. *Proceedings of the ACM SIGMOD Conference on Management of data*, pages 207-216, 1993.

[3] Agrawal R., and Srikant R. Fast Algorithms for Mining Association Rules in Large Databases. *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 478-499, 1994.

[4] Brin S., Motwani R. and Silverstein C. Beyond Market Baskets: Generalizing Association Rules to Correlations. *Proceedings of the ACM SIGMOD, 1997.* pages 265-276.

[5] Brin S., Motwani R. Ullman J. D. and Tsur S. Dynamic Itemset Counting and implication rules for Market Basket Data. *Proceedings of the ACM SIGMOD, 1997.* pages 255-264.

[6] Chen M. S., Han J., and Yu P. S. Data Mining: An Overview from Database Perspective. *IEEE Transactions on Knowledge and Data Engineering.* Volume 8, Number 6, December 1996, 866-883.

[7] Klementtinen M., Mannila H., Ronkainen P., Toivonen H., and Verkamo A. I. Finding interesting rules from large sets of discovered association rules. *Proceedings of the CIKM* 1994.

[8] Lent B., Swami A., and Widom J. Clustering Association Rules. *Proceedings of the Thirteenth International Conference on Data Engineering.* pages 220-231, Birmingham, UK, April 1997.

[9] Park J. S., Chen M. S., and Yu P. S. Using a Hash Based Method with Transaction Trimming for Mining Association Rules. *IEEE Transactions on Knowledge and Data Engineering.* Volume 9, Number 5, September 1997, 813-825.

[10] Mannila H., Toivonen H., and Verkamo A. I. Efficient algorithms for discovering association rules. *AAAI Workshop on Knowledge Discovery in Databases,* 1994, pages 181-192.

[11] Srikant R., and Agrawal R. Mining Generalized Association Rules. *Proceedings of the 21st International Conference on Very Large Data Bases,*1995, pages 407-419.

[12] Srikant R., and Agrawal R. Mining quantitative association rules in large relational tables. *Proceedings of the ACM SIGMOD Conference on Management of Data, 1996.*