Feasible Itemset Distributions in Data Mining: Theory and Application

Ganesh Ramesh
Dept. of Computer Science
University at Albany, SUNY
Albany, NY 12222, USA
ganesh@cs.albany.edu

William A. Maniatty
Dept. of Computer Science
University at Albany, SUNY
Albany, NY 12222, USA
maniatty@cs.albany.edu

Mohammed J. Zaki *
Dept. of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180, USA
zaki@cs.rpi.edu

ABSTRACT

Computing frequent itemsets and maximally frequent itemsets in a database are classic problems in data mining. The resource requirements of all extant algorithms for both problems depend on the distribution of frequent patterns, a topic that has not been formally investigated. In this paper, we study properties of length distributions of frequent and maximal frequent itemset collections and provide novel solutions for computing tight lower bounds for feasible distributions. We show how these bounding distributions can help in generating realistic synthetic datasets, which can be used for algorithm benchmarking.

1. INTRODUCTION

Mining frequent patterns or itemsets is a fundamental task in many data mining applications. These include the discovery of association rules, correlations, sequential rules, episodes, multi-dimensional patterns, and many other important discovery tasks [11]. The problem is formulated as follows: Given a large database of item transactions, find the frequent itemsets, i.e., itemsets that occurs in at least a user-specified percentage of the database.

Over the past decade many interesting algorithms have been proposed for mining frequent itemsets [2, 16, 15, 6, 12]. Typically these methods show good performance with sparse datasets, where the frequent patterns are relatively short. However, in dense datasets with long frequent patterns, which arise in many real world domains (e.g., DNA, Protein sequences) mining all frequent sets quickly becomes infeasible due to the combinatorial explosion; a frequent pattern of length k implies the presence of $2^k - 2$ frequent subsets. One solution is to mine the maximal frequent itemsets [4, 14, 1, 7, 9], which can be orders of magnitude fewer than all frequent patterns.

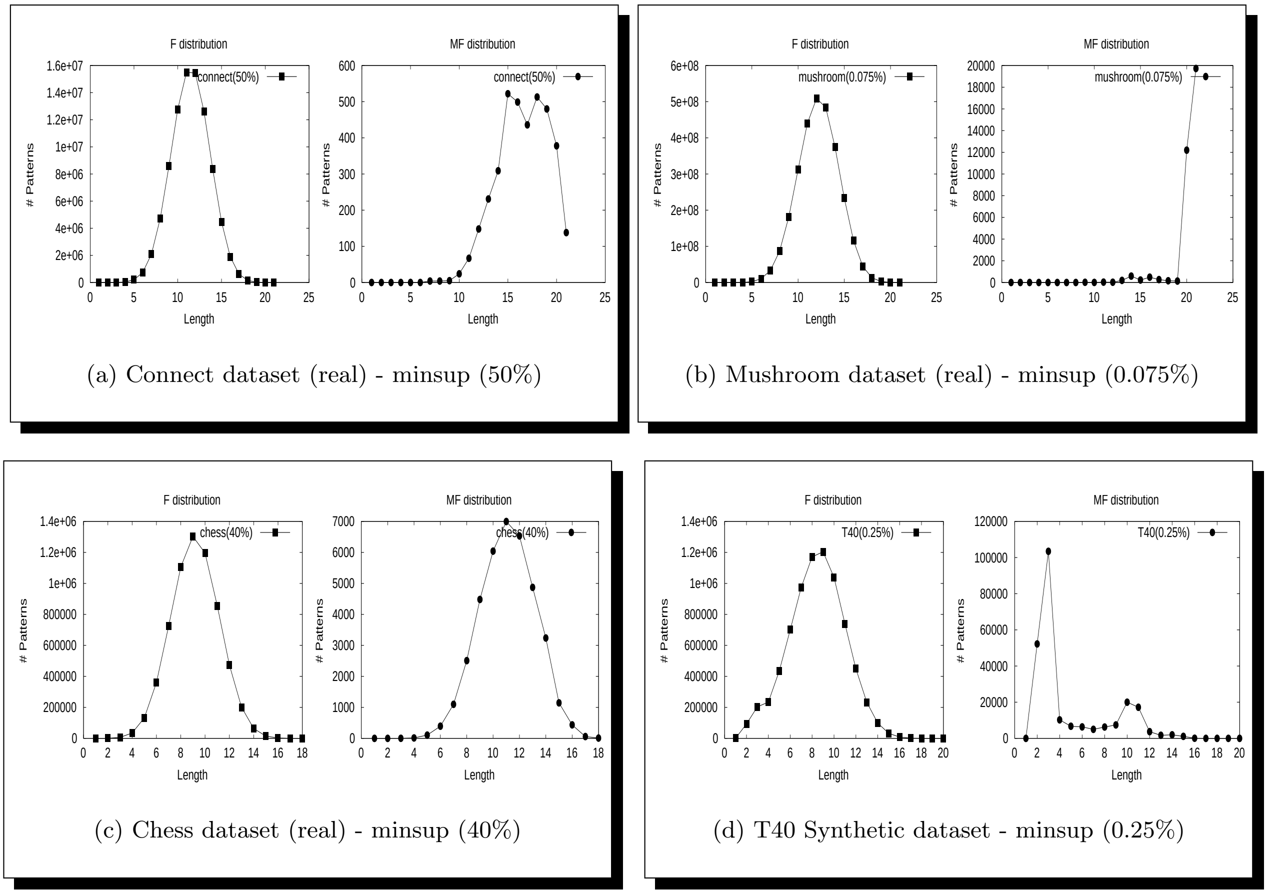
In the final analysis, the performance of methods that mine frequent or maximal patterns depends on the length distribution of mined patterns. A natural question arises: what are the feasible distributions of frequent and maximal frequent itemsets? Put another way, what kinds of distributions can one expect for sparse or dense, and synthetic or real datasets? To the best of our knowledge this fundamental question has not been formally addressed. In a seminal work on applying bounds, Goethals et. al. [8] gave a tight upper bound on the number of candidate patterns that can arise while mining in a level-wise fashion (e.g., in Apriori [2]). Our work is motivated by a different problem which uses related combinatorial results. Gunopulos et. al. [10] give lower bounds on the number of queries to the database for computing support.

Given the multitude of algorithms for mining itemsets, there has arisen a serious need for benchmarking [17]. It was noted that performance on real world data did not reflect the same trends as synthetic data (generated using the method proposed in [2]). This is because the frequent/maximal itemset distributions in real data ([3]) differ significantly from those in synthetic data as depicted in figure 1. The reader is referred to [9] for more details. Furthermore, datasets may be sparse or dense depending on the length of the longest pattern. Thus, there are two crucial problems that research needs to address: 1) to formally characterize the kinds of pattern distributions that may arise, and 2) to generate a variety of benchmark datasets to compare the different algorithms.

In this paper, we address both the problems: First, we characterize properties of length distributions of frequent and maximal frequent itemset collections and provide novel solutions for computing tight lower bounds for feasible distributions. In particular, given a sequence S of non-negative integers, $\langle s_1, s_2, \cdots, s_l \rangle$, we answer the question whether there exists a frequent or maximal frequent itemset collection that has s_i frequent itemsets of length i, for $1 \le i \le l$. Second, we show how these bounding distributions can help in generating realistic synthetic datasets, which can be used for algorithm benchmarking. In particular, given a list of sequences $\mathbf{S} = \mathsf{S}_1, \mathsf{S}_2, \cdots, \mathsf{S}_k$, we show how to construct a database (if it is feasible), such that, if one were to mine frequent or maximal frequent patterns at k successively in-

[*] Supported by NSF Career Award IIS-0092978, NSF grant EIA-0103708, and DOE Career Award DE-FG02-02ER25538.

PODS 2003, June 9-12, 2003, San Diego, CA.
Copyright 2003 ACM 1-58113-670-6/03/06 ...$5.00.



(a) Connect dataset (real) - minsup (50%)

(b) Mushroom dataset (real) - minsup (0.075%)

(c) Chess dataset (real) - minsup (40%)

(d) T40 Synthetic dataset - minsup (0.25%)

Figure 1: Frequent (F) and Maximal Frequent (MF) Itemset Distributions

creasing values of user-specified minimum frequency thresholds, one would get **S** as the sequence of pattern distributions.

2. PRELIMINARIES

DEFINITION 1. [Itemsets] *A positive integer is also called a* natural number. *For any set X, its* size (or length) *is the number of elements in X. Let $(\mathbf{n})$ denote the set of natural numbers $\{1, 2, \cdots, n\}$. Each $x \in (\mathbf{n})$ is called an* item, *and we use $\mathcal{I} = (\mathbf{n})$ to denote a universal set of n items. A non-empty subset of $\mathcal{I}$ is called an* itemset. *The power set of $\mathcal{I}$, denoted $\mathcal{P}(\mathcal{I})$, is the set of all possible itemsets of $\mathcal{I}$. An itemset of size k, $X = \{x_1, x_2, \cdots, x_k\}$ is called a k-itemset (for convenience we drop set notation and denote X as $x_1 x_2 \cdots x_k$). The set of all possible k-itemsets of $\mathcal{I}$, i.e., itemsets of size (or* length*) k, is denoted by $\mathcal{I}^{(k)}$. For $X, Y \in \mathcal{P}(\mathcal{I})$ we say that X* contains *Y if $Y \subseteq X$.*

DEFINITION 2. [Itemset Collections] *A set $\mathcal{F} \subseteq \mathcal{P}(\mathcal{I})$ (with $\emptyset \notin \mathcal{F}$) is called an* itemset collection. *An itemset collection $\mathcal{F}$ is called a* Sperner collection *if no itemset in it contains another: $X, Y \in \mathcal{F}$ and $X \neq Y$, implies $X \not\subset Y$. The* k-collection *of $\mathcal{F}$, denoted $\mathcal{F}_k$, is the collection of all k-itemsets in $\mathcal{F}$, i.e., $\mathcal{F}_k = \mathcal{F} \cap \mathcal{I}^{(k)}$ (or equivalently $\mathcal{F}_k = \{X \in \mathcal{I}^{(k)} \mid X \in \mathcal{F}\}$). On the other hand, the* induced *$k-$collection of $\mathcal{F}$, denoted $[\mathcal{F}]_k$, is the set of k-itemsets contained in some element of $\mathcal{F}$. Formally, $[\mathcal{F}]_k = \{X \in \mathcal{I}^{(k)} \mid X \subseteq Y$ for some $Y \in \mathcal{F}\}$, and $[\mathcal{F}] = \bigcup_k [\mathcal{F}]_k$. Let $f_k = |\mathcal{F}_k|$ denote the size of $\mathcal{F}_k$. Let $l \leq n$ be the length of the longest itemset in $\mathcal{F}$, then the* sequence representation *of $\mathcal{F}$ is the length distribution of itemsets in $\mathcal{F}$, given as $\langle \mathcal{F} \rangle = \langle f_1, f_2, \cdots, f_l \rangle$.*

DEFINITION 3. [Transactions] *A* transaction *T_i is an itemset, where i is a natural number called the* transaction identifier *or* tid. *A* transaction database, *$\mathsf{DB} = \{T_1, \cdots, T_N\}$, is a finite, nonempty multi-set of transactions, with size $|\mathsf{DB}| = N$. The* absolute support *of an itemset X in DB is the number of transactions in DB that contain X, given as $\pi^A(X, \mathsf{DB}) = |\{T_i \in \mathsf{DB} | X \subseteq T_i\}|$. The* (relative) support *of an itemset X in DB is the fraction of transactions in DB that contain X, given as, $\pi(X, \mathsf{DB}) = \frac{\pi^A(X, \mathsf{DB})}{N}$.*

DEFINITION 4. [Frequent and Maximal Frequent Itemsets] *An itemset X is said to be* frequent *if $\pi(X, \mathsf{DB}) \geq \pi^{\min}$, where $\pi^{\min}$ is a user-specified minimum support threshold, with $0 < \pi^{\min} \leq 1$. A collection of frequent itemsets is denoted as $\mathbf{F}(\pi^{\min}, \mathsf{DB})$ (or simply $\mathbf{F}$). A frequent itemset $X \in \mathbf{F}$ is* maximal *if it has no frequent superset, i.e., $(\nexists Y \mid (X \subset Y) \wedge (Y \in \mathbf{F}))$. A collection of maximal frequent itemsets is denoted as $\mathbf{MF}(\pi^{\min}, \mathsf{DB})$ (or simply $\mathbf{MF}$).*

Frequent itemsets are closed under $\subseteq$, as given by the following lemma.

LEMMA 5. [2] *Any subset of a frequent itemset is frequent: $X \in \mathbf{F}$ and $Y \subseteq X$ implies $Y \in \mathbf{F}$.* ■

By definition, $\mathbf{F}$ is a set system and $\mathbf{MF}$ is a Sperner system on $\mathcal{I}$. Also $\mathbf{F}_k = \mathbf{F} \cap \mathcal{I}^{(k)}$ and $\mathbf{MF}_k = \mathbf{MF} \cap \mathcal{I}^{(k)}$ define the frequent and maximal frequent k-collections of $\mathbf{F}$ and $\mathbf{MF}$.

LEMMA 6. *Given* DB *and* $\pi^{\min}$*,* $\mathbf{F} = [\mathbf{MF}]$*. Furthermore,* $\mathbf{MF}$ *is the smallest collection of frequent itemsets from which* $\mathbf{F}$ *can be inferred (provided only the frequent itemsets are required and not their supports).*

PROOF. By definition $\forall X \in \mathbf{F}$, $\exists Y \in \mathbf{MF}$ such that $X \subseteq Y$. It follows that $\mathbf{F}_k = [\mathbf{MF}]_k$, and $\mathbf{F} = \bigcup_k \mathbf{F}_k = \bigcup_k [\mathbf{MF}]_k$. Now assume that there is a frequent itemset collection M', such that $\mathbf{F} = [M']$, with $|M'| < |\mathbf{MF}|$. If $X \in \mathbf{MF}$, then $X \in \mathbf{F}$. So there exists $Y \in M'$ such that $X \subseteq Y$. But X is maximal means $X = Y$, so $X \in M'$. Thus $\forall X, X \in \mathbf{MF} \Rightarrow X \in M'$, contradicting the assumption that $|M'| < |\mathbf{MF}|$. Thus $\mathbf{MF}$ has smallest cardinality. □

DEFINITION 7. [Lex and Colex Order] *Let $X, Y \in \mathcal{F} \cap \mathcal{I}^{(k)}$ be any two distinct k-itemsets in $\mathcal{F}$, with $X = x_1 x_2 \cdots x_k$ and $Y = y_1 y_2 \cdots y_k$. The* lexicographic *(or* lex*) ordering $\preceq_l$ is given as: $X \preceq_l Y$ if and only if $\exists z < k$ such that $\forall i : 1 \leq i < z, x_i = y_i$ and $x_z < y_z$. In contrast the* colex [1] *or* squashed *ordering $\preceq_c$ is given as: $X \preceq_c Y$ if and only if $\exists z < k$ such that $\forall i : z < i \leq k, x_i = y_i$ and $x_z < y_z$. Both lex and colex ordering are total orders on k-itemsets. We define the* rank *of a k-itemset as its position in the ordering, the first element having a rank of* 1*. We denote by $\mathcal{C}^{(k)}(m)$ the first m itemsets in $\mathcal{I}^{(k)}$ in colex order.*

Intuitively, the colex order "uses" as few elements from $\mathcal{I}$ as possible to construct the elements of $\mathcal{F}$. Let $\mathcal{I} = (\mathbf{5})$. The colex order on $\mathcal{I}^{(2)}$ is 12,13,23,14,24,34,15,25,35,45. $rank(12) = 1$ and $rank(24) = 5$. Contrast this with the lex order: 12,13,14,15,23,24,25,34,35,45. Notice that the rank of itemsets in colex order is independent of $|\mathcal{I}|$ (with $|\mathcal{I}| \geq 5$). This, however, is not true of lex order since $rank(24)$ is 6 if $\mathcal{I} = (\mathbf{5})$, but $rank(24)$ is 11 if $\mathcal{I} = (\mathbf{10})$.

3. PROBLEM STATEMENT

In this paper, we address two main questions regarding the feasibility of itemset collections. The *unconstrained* feasibility question assumes no prior knowledge of $\mathcal{I}$. The *constrained* feasibility question deals with the case when the number of items in $\mathcal{I}$ is known. Furthermore, we address the question whether one can construct a database which would produce k given itemset distributions if mined at k distinct (increasing) values of minimum support.

PROBLEM 8. [Feasibility Problems for Itemset Collections] *Let $\mathcal{F}$ be an itemset collection over $\mathcal{I}$, and let S be a sequence of nonnegative integers, $\langle s_1, s_2, \ldots, s_l \rangle$. We address the following two existential questions for collections of frequent ($\mathcal{F} = \mathbf{F}$) and maximal frequent ($\mathcal{F} = \mathbf{MF}$) itemsets:*

1. **Unconstrained Problem***: Does there exist $\mathcal{F}$ such that $\langle \mathcal{F} \rangle = \mathsf{S}$, i.e., $|\mathcal{F} \cap \mathcal{I}^{(k)}| = s_k$, for $1 \leq k \leq l$?*
2. **$\mathcal{I}$-Constrained Problem** *Let $\mathcal{I} = (\mathbf{n})$, does there exist $\mathcal{F}$ such that $\langle \mathcal{F} \rangle = \mathsf{S}$?*

PROBLEM 9. [Feasibility Problem for Database Generation] *Let $\mathsf{S}_1, \mathsf{S}_2, \ldots, \mathsf{S}_k$ be k sequences of nonnegative integers such that $\mathsf{S}_j = \langle s_{j,1}, s_{j,2}, \ldots, s_{j,n_j} \rangle$ i.e, there are n_j nonnegative integers in S_j. Does there exist (and if so, can we construct) a database* DB *and k minimum support levels $\pi_1^{\min}, \ldots, \pi_k^{\min}$ such that*

1. $0 < \pi_1^{\min} \leq \pi_2^{\min} \cdots \leq \pi_k^{\min} \leq 1$
2. *S_j is the sequence representation of $\mathbf{MF}(\pi_j^{\min}, \mathsf{DB})$ for all $1 \leq j \leq k$.*
3. *$\mathbf{MF}(\pi_j^{\min}, \mathsf{DB})$ uses the minimum number of items, for all $1 \leq j \leq k$.*

4. RELATED COMBINATORIAL RESULTS

Not all sequences of nonnegative integers can represent distributions of frequent itemset collections. Lemma 5 which states that frequent k-itemsets induce frequent itemsets of lower cardinality, also implies that frequent k-itemsets impose constraints on the number of frequent itemsets of lower cardinality. Hence, to motivate the solution to feasibility problems for itemset collections, it is necessary to study induced subsets of itemset collections.

For this section, we assume h, l are natural numbers, $\mathcal{I} = (\mathbf{n})$, and $\mathcal{F} = \{F_1, \ldots F_h\}$ is an l-collection of size h, i.e, $\mathcal{F} \subseteq \mathcal{I}^{(l)}$. We use the notation $\partial^k(\mathcal{F})$ to denote the $(l-k)$-itemsets induced by $\mathcal{F}$, i.e., $\partial^k(\mathcal{F}) = [\mathcal{F}]_{l-k}$, where $1 \leq k < l$. We write $\partial^1(\mathcal{F})$ as $\partial(\mathcal{F})$.

4.1 Previously Known Results

Let's consider the following problem: Given l, h, find an $l-$collection $\mathcal{F}$ of size h which induces the smallest number of $(l-1)$-itemsets, i.e., find the minimum value $\min |\partial(\mathcal{F})|$ over all possible collections $\mathcal{F}$.

LEMMA 10. [13] *Given h and l, h can be uniquely written in the form, $h = \sum_{i=t}^{l} \binom{a_i}{i} = \binom{a_l}{l} + \binom{a_{l-1}}{l-1} + \binom{a_{l-2}}{l-2} + \cdots + \binom{a_t}{t}$, where $t \geq 1$, $a_l > a_{l-1} > \cdots > a_t$ are natural numbers and $\forall i : a_i \geq i$.* ■

Lemma 10 says that any integer h can be uniquely written as a sum of binomial coefficients, called the $l-$canonical representation of h in [8]. Let $\mathsf{LB}_l(h) = \sum_{i=t}^{l} \binom{a_i}{i-1} = \binom{a_l}{l-1} + \binom{a_{l-1}}{l-2} + \cdots + \binom{a_t}{t-1}$. The following theorem gives the lower bound for $|\partial(\mathcal{F})|$.

THEOREM 11. [13] *Given $1 \leq l \leq n$, and $1 \leq h \leq \binom{n}{l}$. Then* $\min |\partial(\mathcal{F})| = \mathsf{LB}_l(h)$ *over all systems $\mathcal{F}$.* ■

The numbers a_i uniquely determined by lemma 10 can be computed as follows. The integer a_l satisfies $\binom{a_l}{l} \leq h < \binom{a_l+1}{l}$. The integer a_{l-1} satisfies $\binom{a_{l-1}}{l-1} \leq h - \binom{a_l}{l} < \binom{a_{l-1}+1}{l-1}$, and so on. In general, a_i satisfies $\binom{a_i}{i} \leq h - \sum_{j=i+1}^{l} \binom{a_j}{j} < \binom{a_i+1}{i}$.

What do these numbers a_i signify? These are the numbers used in the construction of $\mathcal{F}$ that achieves the lower bound given by theorem 11. Two cases need consideration:

[1] An alternate definition is given as: $X \preceq_c Y$ if and only if the largest item in symmetric difference of X and Y is in Y only.

1. $\binom{a_l}{l} = h$: here $t = l$ and exactly a_l elements are needed to construct the itemsets in $\mathcal{F}$, i.e., $\mathcal{F} = (\mathbf{a_l})^{(l)}$.
2. $\binom{a_l}{l} < h$: here $a_l + 1$ elements are needed to construct $\mathcal{F}$ as follows. First, construct all the possible l-subsets of $(\mathbf{a_l})$. This accounts for $\binom{a_l}{l}$ subsets in $\mathcal{F}$. The remaining $h - \binom{a_l}{l}$ itemsets must contain the element $a_l + 1$. For these sets to induce the minimum number of subsets, the construction argument proceeds recursively. The problem now is to construct an $(l-1)$-collection of size $h - \binom{a_l}{l}$ such that their induced subsets are minimum. Hence, a_{l-1} is the minimum number of elements needed to construct this $(l-1)$-collection. The recursion proceeds until either a 1-collection needs to be constructed or when there are no sets to be constructed. An example is provided later in this section.

LEMMA 12. [5] *Let $\nabla = \mathcal{C}^{(l)}(|\mathcal{F}|)$ be the $l-$collection of the first $|\mathcal{F}|$ itemsets in colex order. Then the $(l-1)$-itemsets in $\partial(\nabla)$ are the first $|\partial(\nabla)|$ itemsets in colex order and $|\partial(\nabla)| \leq |\partial(\mathcal{F})|$.* ■

THEOREM 13. [13] *Given $1 \leq l \leq n$, and $\binom{n}{l} \leq h \leq 2\binom{n}{l}$. Let G and H be disjoint sets of n items. If $\mathcal{F} = \{F_1, \ldots, F_h\}$ with $F_i \subset G$ or $F_i \subset H$ and $|F_i| = l$, $(1 \leq i \leq h)$, then $\min|\partial(\mathcal{F})| = \binom{n}{l-1} + \mathsf{LB}_l(h - \binom{n}{l})$.* ■

Theorem 11 gives a lower bound on the size of induced $(l-1)$-itemsets of $\mathcal{F}$. A natural generalization is to deduce a lower bound for the induced $(l-k)$-itemsets, with $1 \leq k < l$. Let $\mathsf{LB}_l^k(h) = \sum_{i=t}^{l} \binom{a_i}{i-k} = \binom{a_l}{l-k} + \binom{a_{l-1}}{l-1-k} + \cdots + \binom{a_t}{t-k}$, where $\binom{a}{b} = 0$ if $b < 0$.

THEOREM 14. [13] *Given $1 \leq l \leq n$ and $l \leq h \leq \binom{n}{l}$. Then $\min|\partial^k(\mathcal{F})| = \mathsf{LB}_l^k(h)$ over all such $\mathcal{F}$.* ■

EXAMPLE 15. *To illustrate an example, consider the problem of constructing a 4-collection of size 10 that induces the smallest 3-collection. Let $10 = \sum_{i=0}^{4} \binom{a_i}{i}$. From the canonical representation of 10, we obtain $a_4 = 5, a_3 = 4, a_2 = 2, a_1 = a_0 = 0$. To construct the 4-collection (say A), at least 6 items are needed as $\binom{5}{4} < 10 < \binom{6}{4}$. Hence $a_4 = 5$ and we can set $\mathcal{I} = (\mathbf{6})$. With $a_4 = 5$ elements, it is possible to construct $\binom{5}{4} = 5$ elements of A. These are all the 4-subsets of $(\mathbf{5})$. There are $10 - 5 = 5$ remaining 4-itemsets in A that need construction. In all the remaining 4-itemsets the item 6 is present (the combinations of $(\mathbf{5})$ having been exhausted). The remaining 4-itemsets induce the smallest number of 3-itemsets if the itemsets are constructed by adding the item 6 to the 3-collection of size 5 which induce the smallest number of subsets. This construction recursively proceeds and is illustrated in figure 2. Also note that the collection A that contains the first 10 4-itemsets in colex order, induces the smallest number of j-itemsets for $1 \leq j < 4$.*

4.2 New Results

LEMMA 16. *Let $\nabla = \mathcal{C}^{(l)}(|\mathcal{F}|)$ be the $l-$collection of the first $|\mathcal{F}|$ itemsets in colex order, and $1 \leq k < l$. Then the $(l-k)$-itemsets in $\partial^k(\nabla)$ are the first $|\partial^k(\nabla)|$ itemsets in colex order, and $|\partial^k(\nabla)| \leq |\partial^k(\mathcal{F})|$.*

PROOF. By induction on k.
Basis ($k = 1$): by lemma 12, $\partial(\nabla)$ contains the first $|\partial(\nabla)|$ subsets in colex order and $|\partial(\nabla)| \leq |\partial(\mathcal{F})|$.
Inductive Step: Let $\partial^k(\nabla)$ contain the first $|\partial^k(\nabla)|$ subsets in colex order and $|\partial^k(\nabla)| \leq |\partial^k(\mathcal{F})|$ for some $k \geq 1$. From lemma 12, it follows that $\partial(\partial^k(\nabla)) = \partial^{(k+1)}(\nabla)$ contains the first $|\partial^{k+1}(\nabla)|$ subsets in colex order and $|\partial^{k+1}(\nabla)| \leq |\partial(\partial^k(\mathcal{F}))| = |\partial^{k+1}(\mathcal{F})|$. □

Theorem 14 gives a lower bound on the number of itemsets of a given size induced by a single collection. In practice, frequent itemsets of a given size may be induced by *any* frequent itemset of a higher cardinality. Hence, if we are looking at frequent j-itemsets, then these itemsets may be induced by any k-collection where $k > j$. Hence it is natural to generalize theorem 14 to two or more itemset collections over a given universe of items.

LEMMA 17. *Let $\mathcal{F}_1$ be an l_1-collection of size h_1 and $\mathcal{F}_2$ be a l_2-collection of size h_2 over $\mathcal{I}$, with $l_2 < l_1$. For $1 \leq k < l_2$, let $\mathcal{K}$ be the jointly induced k-collection given by $[\mathcal{F}_1]_k \cup [\mathcal{F}_2]_k$. Then $\min(|\mathcal{K}|) = \max(\mathsf{LB}_{l_1}^{l_1-k}(h_1), \mathsf{LB}_{l_2}^{l_2-k}(h_2))$; the minimum running over all such collections $\mathcal{F}_1$ and $\mathcal{F}_2$.*

PROOF. For any collection $\mathcal{F}_i$, $\mathsf{LB}_{l_i}^{l_i-k}(h_i)$ is the lower bound on the size of the k-collection induced by $\mathcal{F}_i$ and this lower bound is achieved by the first $h_i = |\mathcal{F}_i|$ subsets in colex order (by theorem 14 and lemma 16). Since both collections are over $\mathcal{I}$, $\min(|\mathcal{K}|) = \max(\mathsf{LB}_{l_1}^{l_1-k}(h_1), \mathsf{LB}_{l_2}^{l_2-k}(h_2))$. □

THEOREM 18. *Let $\mathcal{F}_i$ be a l_i-collection of size h_i over $\mathcal{I}$, with $l_i > k$ for $1 \leq i \leq w$. The lower bound on the number of jointly induced k-itemsets, is given by,*
$\min(|\bigcup_{i=1}^{w}[\mathcal{F}_i]_k|) = \max_{i=1}^{w}\{\mathsf{LB}_{l_i}^{l_i-k}(h_i)\}$,
the minimum running over all such collections $\mathcal{F}_i$.

PROOF. The proof is by induction on w.
Basis : For $w = 1$, the statement is the same as theorem 14. For $w = 2$, the basis is true by lemma 17.
Inductive step: Let it be true that
$\min(|\bigcup_{i=1}^{w-1}[\mathcal{F}_i]_k|) = \max_{i=1}^{w-1}\{\mathsf{LB}_{l_i}^{l_i-k}(h_i)\}$. This maximum is achieved by some p, where $2 \leq p \leq w-1$. Hence,
$\min(|\bigcup_{z=1}^{w}[\mathcal{F}_i]_k|) = \min(|[\mathcal{F}_p]_k \cup [\mathcal{F}_w]_k|) = \max\{\mathsf{LB}_{l_p}^{l_p-k}(h_p), \mathsf{LB}_{l_w}^{l_w-k}(h_w)\}$.
But, $\mathsf{LB}_{l_p}^{l_p-k}(h_p) = \max_{i=1}^{w-1}\{\mathsf{LB}_{l_i}^{l_i-k}(h_i)\}$
$\Rightarrow \min(|\bigcup_{z=1}^{w}[\mathcal{F}_i]_k|) = \max_{i=1}^{w}\{\mathsf{LB}_{l_i}^{l_i-k}(h_i)\}$. □

EXAMPLE 19. *Consider an example where A is a collection of 5-itemsets of size 3 and B is a collection of 3-itemsets of size 25, both collections over $\mathcal{I}$. As 2-itemsets are induced by both A and B, the problem is to determine the lower bound on the number of 2-itemsets jointly induced by both A and B (over all such collections A and B). If we consider the collections independently, the smallest 2-collection is induced by the first $|\mathsf{A}|$ 5-itemsets in* colex *order and the first $|\mathsf{B}|$ 3-itemsets in* colex *order, respectively. This is illustrated in figure 3. But it can be observed that the first $\mathsf{LB}_5^2(3)$ itemsets in B are already induced by A, as the collections are taken from the same universe of items, and these itemsets are in colex order by lemma 16. Hence, the lower bound on the size of the 2-collection jointly induced by A and B is the maximum of the lower bound independently induced by A and B. This construction is also illustrated in the figure.*

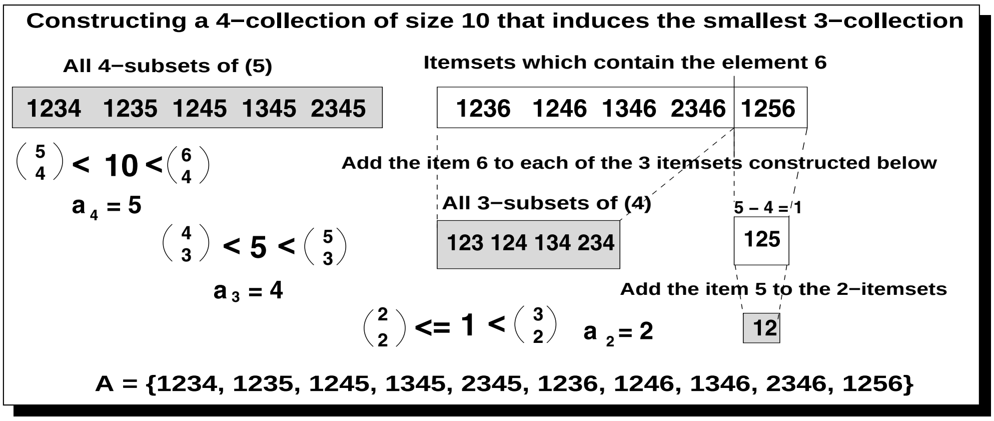

Constructing a 4−collection of size 10 that induces the smallest 3−collection
All 4−subsets of (5)
Itemsets which contain the element 6
1234 1235 1245 1345 2345
1236 1246 1346 2346 1256
$\binom{5}{4} < 10 < \binom{6}{4}$
Add the item 6 to each of the 3 itemsets constructed below
$a_4 = 5$
All 3−subsets of (4)
5 − 4 = 1
$\binom{4}{3} < 5 < \binom{5}{3}$
123 124 134 234
125
$a_3 = 4$
Add the item 5 to the 2−itemsets
$\binom{2}{2} <= 1 < \binom{3}{2}$
$a_2 = 2$
12
A = {1234, 1235, 1245, 1345, 2345, 1236, 1246, 1346, 2346, 1256}

Figure 2: Constructing a 4-collection of size 10

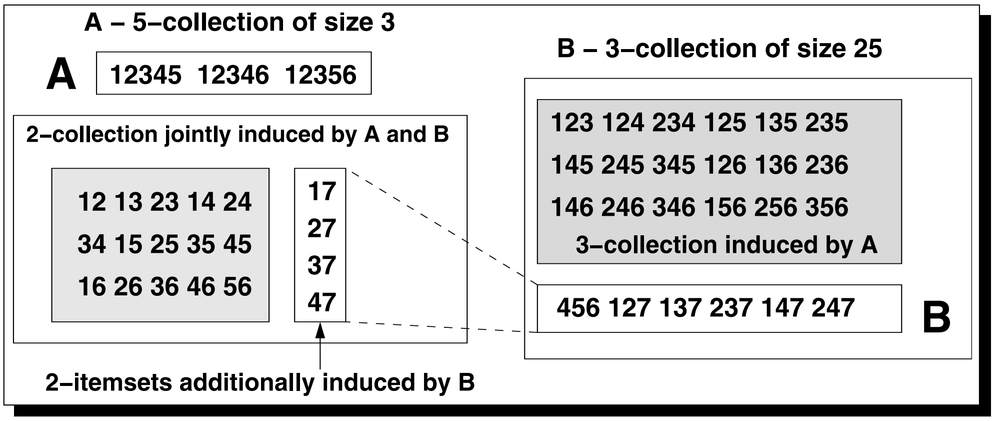


Figure 3: Two itemset collections A and B jointly inducing a Lower bound

5. SOLUTIONS TO FEASIBILITY PROBLEMS FOR ITEMSET COLLECTIONS

Let $\mathcal{F}$ be an itemset collection over $\mathcal{I}$ and let $\langle \mathcal{F} \rangle = \langle f_1, f_2, \cdots, f_l \rangle$ be its sequence representation (i.e., $|\mathcal{F} \cap \mathcal{I}^{(k)}| = f_k$ for $1 \leq k \leq l$), and let $\mathsf{S} = \langle s_1, s_2, \ldots, s_l \rangle$ be a sequence of nonnegative integers. We assume $|\mathcal{I}| \geq l$. We write $\langle \mathcal{F} \rangle = \mathsf{S}$ iff $f_k = s_k$, for $1 \leq k \leq l$. We address the following two existential questions for collections of frequent ($\mathcal{F} = \mathbf{F}$) and maximal frequent ($\mathcal{F} = \mathbf{MF}$) itemsets, as mentioned in problem 8: 1) Does there exist (unconstrained) $\mathcal{F}$ such that $\langle \mathcal{F} \rangle = \mathsf{S}$? 2) Given $\mathcal{I} = (\mathbf{n})$, does there exist ($\mathcal{I}$-constrained) $\mathcal{F}$ such that $\langle \mathcal{F} \rangle = \mathsf{S}$?

5.1 Frequent Itemset Collections

LEMMA 20. *If $f_l \geq 1$, then $f_k \geq \binom{l}{k}$, for $1 \leq k \leq l$.*

PROOF. If $f_l \geq 1$, then there is at least one frequent $l-$itemset, say X. By lemma 5, all subsets of X must be frequent, i.e., X induces $\binom{l}{k}$ k-itemsets. Hence, $f_k \geq \binom{l}{k}$, for $1 \leq k \leq l$. □

LEMMA 21. *If $f_l \geq 1$, then $f_k \leq \binom{f_1}{k}$, $1 \leq k \leq l$.*

PROOF. Without loss of generality we may assume that $\mathcal{I} = (\mathbf{f_1})$, since there are f_1 single items in $\mathcal{F}$. By definition, $[\mathcal{F}]_k \subseteq \mathcal{I}^{(k)} = (\mathbf{f_1})^{(k)}$. Hence, $f_k \leq \binom{f_1}{k}$, for $1 \leq k \leq l$. □

Lemma 20 and 21 give a simple lower and upper bound on f_k by using length of the longest itemset (l), and number of frequent items (f_1), respectively. Below we develop tighter bounds on f_k.

LEMMA 22. *If $f_l \geq 1$, then $f_k \geq \mathsf{LB}_l^{l-k}(f_l)$, for $1 \leq k < l$.*

PROOF. By theorem 14, the size of the smallest k-collection induced by any l-collection of size f_l, with $k < l$, is given as $\mathsf{LB}_l^{l-k}(f_l)$. Hence $f_k \geq \mathsf{LB}_l^{l-k}(f_l)$, for $1 \leq k < l$. □

Lemma 22 gives a lower bound on k-itemsets induced by only the l-itemsets. Hence this lower bound is tight only for the induced $(l-1)$-itemsets (i.e., for $k = 1$). For a tighter lower bound on f_k we have to consider the k-itemsets jointly induced by all j-collections of size f_j above level k, i.e., for $k < j \leq l$.

THEOREM 23. $f_k \geq \max_{j=k+1}^{l}\{\mathsf{LB}_j^{j-k}(f_j)\}$, $1 \leq k < l$.

PROOF. Frequent k-itemsets are induced by all frequent j-itemsets, for $k+1 \leq j \leq l$. Hence, by theorem 18, $f_k \geq \max_{j=k+1}^{l}\{\mathsf{LB}_j^{j-k}(f_j)\}$. □

THEOREM 24. [Unconstrained and $\mathcal{I}$-Constrained Solution] *Given a sequence of non-negative integers $\mathsf{S} = \langle s_1, s_2, \ldots, s_l \rangle$, there exists a frequent itemset collection $\mathbf{F}$ over $\mathcal{I}$, with $\langle \mathbf{F} \rangle = \mathsf{S}$ iff:*

1. $s_k \leq \binom{s_1}{k}$, $1 \leq k \leq l$.

2. $s_k \geq \max_{j=k+1}^{l}\{\mathsf{LB}_j^{j-k}(s_j)\}$ *for* $1 \leq k < l$.

3. ($\mathcal{I}$-constraint) If $\mathcal{I} = (\mathbf{n})$, $s_1 \leq n$.

PROOF. Suppose there exists a frequent itemset collection $\mathbf{F}$ over $\mathcal{I}$, such that $|\mathbf{F} \bigcap \mathcal{I}^{(k)}| = f_k = s_k$, $1 \leq k \leq l$. By lemma 21, we have $s_k \leq \binom{s_1}{k}$ and by theorem 23, $s_k \geq \max_{j=k+1}^{l}\{\mathsf{LB}_j^{j-k}(s_j)\}$. If $\mathcal{I} = (\mathbf{n})$ then at most n items can be used to construct the itemsets, thus $s_1 \leq n$.

Suppose S satisfies the three conditions, then let $[\mathbf{F}]_k = \mathcal{C}^{(k)}(s_k)$ be the collection of the first s_k k-itemsets over $\mathcal{I}$ (over $(\mathbf{n})$ for the constrained case) in colex order, for $1 \leq k \leq l$. By lemma 16, $\mathbf{F} = \cup_{k=1}^{l}[\mathbf{F}]_k$ satisfies the conditions of this theorem. □

5.2 Maximal Frequent Itemset Collections

FACT 25. *Let $X \subseteq \mathcal{I}^{(k)}$ for any $1 \leq k \leq |\mathcal{I}|$. Then X is a collection of maximal itemsets.*

THEOREM 26. [Unconstrained Solution] *Given sequence $\mathsf{S} = \langle s_1, s_2, \ldots, s_l \rangle$, there exists a collection of maximal frequent itemsets $\mathbf{MF}$, such that $|\mathbf{MF} \cap \mathcal{I}^{(k)}| = s_k$, $1 \leq k \leq l$.*

PROOF. Construct $\mathbf{MF}$ by adding exactly s_k itemsets of size k, $1 \leq k \leq l$, such that any two itemsets X, Y, across all levels k, are disjoint, i.e., $X \cap Y = \emptyset$. By construction each such itemset is maximal, giving $|\mathbf{MF} \cap \mathcal{I}^{(k)}| = s_k$. □

The above theorem states that it is feasible to generate a maximal collection for any sequence S, provided there is no constraint on $\mathcal{I}$. The construction above uses $\sum_{k=1}^{l} k \cdot s_k$ items for constructing $\mathbf{MF}$. Let us call this solution a *direct solution* to the feasibility problem for maximal itemset collections. A natural question arises: what is the minimum number of items, $\min|\mathcal{I}|$, such that $\langle \mathbf{MF} \rangle = \mathsf{S}$? For the following discussion let $\mathsf{S} = \langle s_1, s_2, \cdots, s_l \rangle$, let $r_l = 0$, and let $r_k = \max_{j=k+1}^{l}\{\mathsf{LB}_j^{j-k}(r_j + s_j)\}$ for $1 \leq k \leq l$. Note that $r_1 = \max_{j=2}^{l}\{LB_j^{j-1}(r_j + s_j)\}$.

THEOREM 27. *Given S, construct an itemset collection $\mathcal{F}$ as follows: For all $1 \leq k \leq l$, add to $\mathcal{F}$ the s_k itemsets of $\mathcal{I}^{(k)}$ in colex order with ranks $r_k+1, r_k+2, \cdots, r_k+s_k$. Then $\mathcal{F}$ is a maximal itemset collection, such that $\langle \mathcal{F} \rangle = \mathsf{S}$, and $\min|\mathcal{I}| = r_1 + s_1$ is the minimum number of items needed to construct $\mathcal{F}$.*

PROOF. We shall also prove by induction on k that $\cup_{j=k}^{l}\mathcal{F}_k$ contains maximal j-itemsets for $k \leq j \leq l$, where $\mathcal{F}_k = \mathcal{F} \cap \mathcal{I}^{(k)}$. Thus $\mathbf{MF} = \mathcal{F} = \cup_{j=1}^{l}\mathcal{F}_j$ will be a maximal collection. Let $\mathsf{S}_2 = \langle r_1 + s_1, r_2 + s_2, \cdots, r_l + s_l \rangle$. We shall also show that S_2 is the sequence representation of all frequent itemsets induced by $\mathcal{F}$. For the basis step we consider 2 cases:

- *Case I* ($k = l$): By construction, F_l contains first s_l l-itemsets of $\mathcal{I}^{(l)}$ in colex order. By lemma 16, F_l uses the minimum number of items, and by fact 25, F_l is a maximal collection. Note that at length l, there are only $r_l + s_l = 0 + s_l = s_l$ frequent (maximal) itemsets.
- *Case II* ($k = l - 1$): By lemma 16, F_l induces $r_{l-1} = \mathsf{LB}_l^1(s_l)$ itemsets of size $l-1$, which are the first r_{l-1} itemsets of $\mathcal{I}^{(l-1)}$ in colex order. None of these induced itemsets are maximal. The maximal $(l-1)$-itemsets which use the minimum number of items are the next s_{l-1} itemsets in colex order after rank r_{l-1}. That is, we add to F_{l-1} the s_{l-1} itemsets with colex ranks $r_{l-1}+1, r_{l-1}+2, \ldots r_{l-1}+s_{l-1}$, respectively. By fact 25 these new itemsets are maximal. Hence $\cup_{k=l-1}^{l}\mathcal{F}_k$ is a maximal collection. By adding the r_{l-1} induced itemsets and the s_{l-1} maximal itemsets we get $r_{l-1} + s_{l-1}$ frequent itemset at level $l-1$.

For the inductive hypothesis, assume that $\cup_{j=k+1}^{l}\mathcal{F}_j$ is a maximal collection using the minimum number of items, and that there are $r_j + s_j$ frequent induced j-itemsets for $k+1 \leq j \leq l$. Let $j = k$, then $r_k = \max_{i=k+1}^{l}\{\mathsf{LB}_i^{i-k}(r_i + s_i)\}$, gives the number of k-itemsets induced jointly by the i-collections, for $k+1 \leq i \leq l$. By lemma 16 these are the first r_k itemsets in colex order, and are non-maximal since they are induced. By fact 25, the s_k itemsets with colex rank $r_k+1, r_k+2, \ldots, r_k+s_k$ are maximal, and use the minimum number of items. Adding the r_k induced itemsets with s_k maximal itemset we get a total of $r_k + s_k$ frequent k-itemsets.

Therefore, the minimum number of items from $\mathcal{I}$ needed to construct $\mathbf{MF} = \mathcal{F}$ with $\langle \mathbf{MF} \rangle = \mathsf{S}$ is is given by $r_1 + s_1$. Furthermore $\langle [\mathbf{MF}] \rangle = \mathsf{S}_2$. □

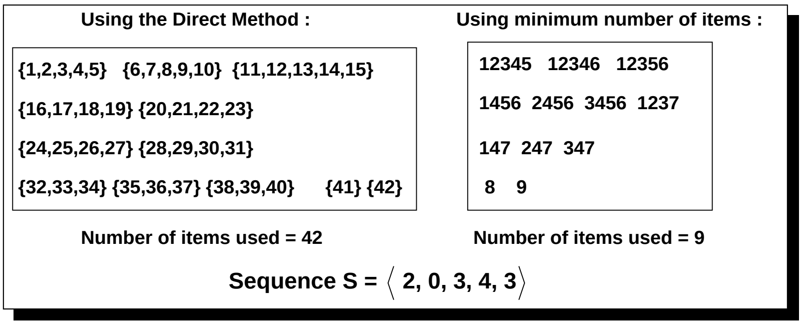
THEOREM 28. [$\mathcal{I}$-Constrained Solution] *Given S, let $\mathcal{I} = (\mathbf{n})$, where $n \geq r_1 + s_1$. Then there exists a maximal itemset collection MF, such that $\langle \mathbf{MF} \rangle = \mathsf{S}$.*

PROOF. By theorem 27, $r_1 + s_1$ is the minimum number of items, $\min|\mathcal{I}|$, required to construct a maximal frequent itemset collection satisfying $|\mathbf{MF} \cap \mathcal{I}^{(k)}| = s_k$, $1 \leq k \leq |\mathcal{I}|$. Thus if $n \geq r_1 + s_1$, then by the construction used in theorem 27 we can generate $\mathbf{MF} = \mathcal{F}$. □

EXAMPLE 29. *Figure 4 illustrates an example of constructing a maximal frequent itemset collection using the direct method and using the minimum number of items, whose length distribution is given by the sequence $\mathsf{S} = \langle 2, 0, 3, 4, 3 \rangle$. The direct method is to construct the itemsets such that they are pairwise disjoint. This method uses 42 items to construct the maximal frequent itemset collection. For constructing the collection using the minimum number of items, the 5-itemsets are constructed first. Since there are 3 itemsets of length 5 in the collection, these are precisely the first 3 itemsets in the* colex *ordering of 5-itemsets. These itemsets induce 4-itemsets (which are the first $\mathsf{LB}_5(3)$ 4-itemsets in colex order). The next 4 itemsets in the colex ordering of 4-itemsets are* 1456*,* 2456*,* 3456 *and* 1237 *respectively. Now, the 5-itemsets and the 4-itemsets jointly induce 3-itemsets (which are in colex order). This procedure is repeated until the itemsets are constructed for the entire sequence. The minimum number of items used to construct the maximal frequent collection for sequence S is 9.*

6. SOLUTION TO FEASIBILITY PROBLEM FOR DATABASE GENERATION

We now address the database generation problem, i.e., given a list of k sequences of non-negative integers, $\mathbf{S} = \mathsf{S}_1, \mathsf{S}_2, \cdots, \mathsf{S}_k$, generate a database DB and real numbers $\pi_i^{\min}$ $(1 \leq i \leq k)$, such that, mining maximal itemsets $\mathbf{MF}_i$ at minimum support $\pi_i^{\min}$ results in S_i as the sequence representation of $\mathbf{MF}_i$, for all $1 \leq i \leq k$. A formal problem statement appears in problem 9. We present solutions for constructing DB for a single sequence S_1, and for a pair of sequences $\mathsf{S}_1, \mathsf{S}_2$, before giving a general solution for any k.



Using the Direct Method :

{1,2,3,4,5} {6,7,8,9,10} {11,12,13,14,15}

{16,17,18,19} {20,21,22,23}

{24,25,26,27} {28,29,30,31}

{32,33,34} {35,36,37} {38,39,40} {41} {42}

Number of items used = 42

Using minimum number of items :

12345 12346 12356

1456 2456 3456 1237

147 247 347

8 9

Number of items used = 9

Sequence S = $\left\langle 2, 0, 3, 4, 3 \right\rangle$

Figure 4: Example - constructing maximal itemset collections

Let r be a natural number, and let $\mathcal{D}^r(\mathcal{F})$ be a *database generation* operator on itemset collection $\mathcal{F}$, defined as follows: For every $X \in \mathcal{F}$, $\mathcal{D}^r$ generates r *tids* and adds the transaction $T_i = X, \forall i = 1, \cdots, r$ to the database. Further, the *tids* are unique across all transactions. For convenience, we write $\mathcal{D}^1(\mathcal{F})$ as $\mathcal{D}(\mathcal{F})$. By construction $\mathcal{D}^r(\mathcal{F})$ has $r \times |\mathcal{F}|$ transactions, with r copies of each $X \in \mathcal{F}$.

Theorem 30. *Given sequence* S *and a maximal collection* $\mathbf{MF}$ *constructed by procedure of theorem 26. Then at* $\pi^{\min} = \frac{1}{|\mathsf{DB}|}$, $\mathsf{DB} = \mathcal{D}(\mathbf{MF})$ *is a feasible database, provided there is no constraint on the number of items used.*

Proof. By construction every $X \in \mathbf{MF}$ appears once in $\mathsf{DB} = \mathcal{D}(\mathbf{MF})$. Thus mining at $\pi^{\min} = \frac{1}{|\mathsf{DB}|}$ yields $\mathbf{MF}$, and by theorem 26, $\langle \mathbf{MF} \rangle = \mathsf{S}$. □

Theorem 31. *Given sequence* S *and a maximal collection* $\mathbf{MF}$ *constructed by procedure of theorem 27. Then,* $\mathsf{DB} = \mathcal{D}(\mathbf{MF})$ *and* $\pi^{\min} = \frac{1}{|\mathsf{DB}|}$ *is a solution for problem 9, when* $k = 1$ *(using minimum number of items).*

Proof. By construction every $X \in \mathbf{MF}$ appears once in $\mathsf{DB} = \mathcal{D}(\mathbf{MF})$. Thus mining at $\pi^{\min} = \frac{1}{|\mathsf{DB}|}$ yields $\mathbf{MF}$. By theorem 27, $\langle \mathbf{MF} \rangle = \mathsf{S}$, and $\mathbf{MF}$ uses the smallest number of items. □

Corollary 32. *Given sequence* S *and a maximal collection* $\mathbf{MF}$ *constructed by procedure of theorem 27. Then,* $\mathsf{DB} = \mathcal{D}([\mathbf{MF}])$ *and* $\pi^{\min} = \frac{1}{|\mathcal{D}([\mathbf{MF}])|}$ *is a solution for problem 9, when* $k = 1$.

Proof. Recall that $\mathcal{F} = [\mathbf{MF}]$ is the set of all induced frequent itemsets of $\mathbf{MF}$. Each $X \in \mathcal{F}$ appears once in $\mathsf{DB} = \mathcal{D}(\mathcal{F})$. Thus mining at $\pi^{\min} = \frac{1}{|\mathsf{DB}|}$ yields $\mathbf{MF}$. By theorem 27, $\langle \mathbf{MF} \rangle = \mathsf{S}$, and $\mathbf{MF}$ uses the smallest number of items. □

Lemma 33. [Multiplicity Lemma] *Given sequence* S *and a maximal collection* $\mathbf{MF}$ *constructed by procedure of theorem 27. Let* $N > 0$ *be a natural number denoting the desired number of transactions. If* $|\mathbf{MF}| \leq N$*, then there exists a database* DB *with* $|\mathsf{DB}| = N$ *and* $0 < \pi^{\min} \leq 1$*, such that* $\mathbf{MF}(\pi^{\min}, \mathsf{DB})$ *has the sequence representation* S.

Proof. Let $r = \lfloor \frac{N}{|\mathbf{MF}|} \rfloor$ and let $m = (N \ mod\ |\mathbf{MF}|)$. Let $\mathbf{MF}_m$ be any subset of $\mathbf{MF}$ of cardinality m. Let $\mathsf{DB} = \mathcal{D}(\mathcal{D}^r(\mathbf{MF}) \cup \mathcal{D}(\mathbf{MF}_m))$ [2] and $\pi^{\min} = \frac{r}{N}$. Since every $X \in \mathbf{MF}$, is replicated at least r times in DB, mining at $\pi^{\min} = r/N$ yields $\mathbf{MF}$. □

Lemma 34. *Let* DB *be a transaction database and let* $0 < \pi_1^{\min} \leq \pi_2^{\min} \leq 1$ *be two levels of minimum support. Then,* $\mathbf{F}(\pi_2^{\min}, \mathsf{DB}) \subseteq \mathbf{F}(\pi_1^{\min}, \mathsf{DB})$.

Proof. Let $X \in \mathbf{F}(\pi_2^{\min}, \mathsf{DB})$. By definition, $\pi(X, \mathsf{DB}) \geq \pi_2^{\min} \geq \pi_1^{\min}$. Hence $X \in F(\pi_1^{\min}, \mathsf{DB})$. □

Corollary 35. *Let* DB *be a transaction database and let* $0 < \pi_1^{\min} \leq \pi_2^{\min} \leq 1$ *be two levels of minimum support. Then,* $\mathbf{MF}(\pi_2^{\min}, \mathsf{DB}) \subseteq \mathbf{F}(\pi_1^{\min}, \mathsf{DB})$.

Proof. From lemma 34 and $\mathbf{MF}(\pi_2^{\min}, \mathsf{DB}) \subseteq \mathbf{F}(\pi_2^{\min}, \mathsf{DB})$, the corollary follows. □

Corollary 36. *Let* DB *be a transaction database and let* $0 < \pi_1^{\min} \leq \pi_2^{\min} \leq 1$ *be two levels of minimum support. Then,* $\forall X \in \mathbf{MF}(\pi_2^{\min}, \mathsf{DB}), \exists Y \in \mathbf{MF}(\pi_1^{\min}, \mathsf{DB})$*, such that* $X \subseteq Y$.

Proof. By corollary 35, $\mathbf{MF}(\pi_2^{\min}, \mathsf{DB}) \subseteq \mathbf{F}(\pi_1^{\min}, \mathsf{DB})$. By definition of $\mathbf{MF}$, $\forall X \in \mathbf{F}(\pi_1^{\min}, \mathsf{DB}), \exists Y \in \mathbf{MF}(\pi_1^{\min}, \mathsf{DB})$ such that $X \subseteq Y$. □

The following theorem gives a solution to problem 9 when the k sequences are all the same.

Theorem 37. [Equal Sequences] *Given sequences* $\mathsf{S} = \mathsf{S}_1 = \mathsf{S}_2 = \cdots = \mathsf{S}_k$ *and a maximal collection* $\mathbf{MF}$ *constructed by procedure of theorem 27. Then the database* $\mathsf{DB} = \mathcal{D}^k(\mathbf{MF})$ *and* $\pi_i^{\min} = \frac{i}{|\mathsf{DB}|}$*, for* $1 \leq i \leq k$ *is a solution to problem 9.*

Proof. Every $X \in \mathbf{MF}$ appears k times in DB. Mining at $\pi_i^{\min}$ yields $\mathbf{MF}$ for all $1 \leq i \leq k$. □

Before considering the general case, let us consider the case when $k = 2$. Suppose that we are given two distinct sequences $\mathsf{S}_1 = \langle s_1^1, \ldots, s_{n_1}^1 \rangle$ and $\mathsf{S}_1 = \langle s_1^2, \ldots, s_{n_2}^2 \rangle$. Under what conditions does a solution to problem 9 exist?

Theorem 38. *Let* DB *be a transaction database and let* $0 < \pi_1^{\min} \leq \pi_2^{\min} \leq 1$ *be two levels of minimum support. Let* $\mathbf{MF}(\pi_1^{\min}, \mathsf{DB})$ *and* $\mathbf{MF}(\pi_2^{\min}, \mathsf{DB})$ *have the sequence representations* S_1 *and* S_2*. For* $i = 1, 2$*, let* $r_{n_i}^i = 0$*, and* $r_j^i = \max_{k=j+1}^{l}\{\mathsf{LB}_k^{k-j}(r_k^i + s_k^i)\}$*, for* $1 \leq j < n_i$*. Then,* $\mathsf{DB}, \pi_1^{\min}, \pi_2^{\min}$ *is a solution for problem 9 when* $k = 2$ *iff*

1. $n_2 \leq n_1$.
2. $s_j^2 \leq (r_j^1 + s_j^1) - r_j^2$ *for* $1 \leq j < n_2$*. In the case when* $n_1 = n_2$*,* $s_{n_2}^2 \leq s_{n_2}^1$.

Proof. $\Rightarrow$: Let $\mathsf{DB}, \pi_1^{\min}, \pi_2^{\min}$ be a solution to problem 9.

[2] Let $Z = \mathcal{D}^r(\mathbf{MF}) \cup \mathcal{D}(\mathbf{MF}_m)$. By setting $\mathsf{DB} = \mathcal{D}(Z)$, every transaction has a unique tid.

```
Algorithm K-DbGen
Given: $k$ sequences $\mathsf{S}_1, \ldots, \mathsf{S}_k$. Sequence $\mathsf{S}_i$
        is of length $n_i$, $1 \leq i \leq k$
Output: A database $\mathsf{DB}$ and $k$ real numbers
        $\pi_1^{\min}, \ldots, \pi_k^{\min}$
1. $\mathsf{DB}_1 = \mathcal{D}(\mathbf{MF}_{\mathsf{S}_1}); a_1 = \mathsf{msup}(\mathsf{DB}_1); c_1 = 1;$
2. $j = 2; \mathsf{DB} = \mathsf{DB}_1;$
3. while($j < k$){
       (a) $\mathsf{DB}_j = \mathcal{D}^{a_{j-1}+1}(\mathbf{MF}_{\mathsf{S}_j});$
       (b) $c_j = a_{j-1} + 1;$
       (c) $\mathsf{DB} = \mathcal{D}(\mathsf{DB} \cup \mathsf{DB}_j);$
       (d) $a_j = \mathsf{msup}(\mathsf{DB});$
   } //end while
4. for($i = 0; i <= k; i++$) $\pi_i^{\min} = \frac{c_i}{|\mathsf{DB}|};$
```

Figure 5: Construct Database given k sequences $\{\mathsf{S}_1, \ldots, \mathsf{S}_k\}$

1. To prove $(n_2 \leq n_1)$: Suppose that $n_2 > n_1$. There exists an $X \in \mathbf{MF}(\pi_2^{\min}, \mathsf{DB})$ such that $|X| = n_2$. Since $n_2 > n_1$, $\nexists Y \in \mathbf{MF}(\pi_1^{\min}, \mathsf{DB})$ such that $X \subseteq Y$, contradicting lemma 34.

2. $\mathbf{MF}(\pi_1^{\min}, \mathsf{DB})$ has the sequence representation given by S_1 and uses the minimum number of items. Thus $\mathbf{MF}$ corresponds to the maximal collection constructed using theorem 27. Since $\pi_1^{\min} \leq \pi_2^{\min}$, $\mathbf{MF}(\pi_2^{\min}, \mathsf{DB}) \subseteq [\mathbf{MF}(\pi_1^{\min}, \mathsf{DB})]$. The number of j-itemsets in the collection $[\mathbf{MF}(\pi_1^{\min}, \mathsf{DB})]$ is given by $r_j^1 + s_j^1$, where r_j^1 gives the j-itemsets induced by the higher cardinality itemsets and s_j^1 gives the maximal itemsets in the collection $\mathbf{MF}(\pi_1^{\min}, \mathsf{DB})$. Hence this is the upper bound on the number of maximal j-itemsets at any level of minimum support greater than $\pi_1^{\min}$. At support level $\pi_2^{\min}$, the number of j-itemsets that are induced (and hence non-maximal) by higher cardinality itemsets is r_j^2. These itemsets cannot belong to $\mathbf{MF}(\pi_2^{\min}, \mathsf{DB})$. Hence the number of possible j-itemsets is given by $r_j^1 + s_j^1 - r_j^2$.

$\Leftarrow$: Let S_1 and S_2 be two sequences satisfying the two conditions in the theorem. Denote by $\mathbf{MF}_{\mathsf{S}_i}$ the maximal collection with sequence representation S_i. Let $\mathbf{MF}_{\mathsf{S}_1}$ be the maximal itemset collection constructed using theorem 27. This construction uses the minimum number of items to construct the maximal itemsets. Construct the set $\mathbf{MF}_{\mathsf{S}_2}$ (from the same universe of items) as follows. Add the first $s_{n_2}^2$ n_2-itemsets in colex order to $\mathbf{MF}_{\mathsf{S}_2}$. For each $1 \leq k < n_2$, add to $\mathbf{MF}_{\mathsf{S}_2}$ the s_k^2 k-itemsets with ranks $r_k^2 + 1, r_k^2 + 2, \cdots, r_k^2 + s_k^2$ in the colex order. Then, the database $\mathsf{DB} = \mathcal{D}(\mathcal{D}(\mathbf{MF}_{\mathsf{S}_1}) \cup \mathcal{D}(\mathbf{MF}_{\mathsf{S}_2}))$, $\pi_1^{\min} = \frac{1}{|\mathsf{DB}|}$, $\pi_2^{\min} = \frac{2}{|\mathsf{DB}|}$ is a solution to problem 9. This is because by construction $\mathbf{MF}_{\mathsf{S}_2} \subseteq [\mathbf{MF}_{\mathsf{S}_1}]$. The maximal itemset collection $\mathbf{MF}(\pi_1^{\min}, \mathsf{DB})$ is precisely the set $\mathbf{MF}_{\mathsf{S}_1}$ and $\mathbf{MF}(\pi_2^{\min}, \mathsf{DB})$ is precisely the set $\mathbf{MF}_{\mathsf{S}_2}$. □

Theorem 39. *Let $1 \leq i \leq k$, and $\mathsf{S}_i = \langle s_1^i, s_2^i, \cdots, s_{n_i}^i \rangle$, with $k > 2$. Let DB be a database and let $0 < \pi_1^{\min} \leq \pi_2^{\min} \leq \cdots \leq \pi_k^{\min} \leq 1$ be k minimum supports, such that $\mathbf{MF}(\pi_i^{\min}, \mathsf{DB})$ has sequence representation S_i. Let r_j^i be as defined in theorem 38, for $1 \leq i \leq k$ and $1 \leq j < n_i$. Then, DB, and $\pi_1^{\min}, \ldots, \pi_k^{\min}$ give a solution to problem 9 iff*

1. $n_i \leq n_{i-1}$ for $1 < i \leq k$.

2. $s_j^i \leq (r_j^{i-1} + s_j^{i-1}) - r_j^i$ for $1 \leq j < n_i$, $1 < i \leq k$.

Proof. $\Rightarrow$: Let $\mathsf{DB}, \pi_1^{\min}, \cdots, \pi_k^{\min}$ as defined above be a solution to problem 9. Since $0 \leq \pi_1^{\min} \leq \pi_2^{\min} \leq \cdots \leq \pi_k^{\min}$, by lemma 34 $\mathbf{MF}(\pi_k^{\min}, \mathsf{DB}) \subseteq [\mathbf{MF}(\pi_{k-1}^{\min}, \mathsf{DB})] \cdots \subseteq [\mathbf{MF}(\pi_1^{\min}, \mathsf{DB})]$. Being a solution to problem 9, $\mathbf{MF}(\pi_i^{\min}, \mathsf{DB})$ has sequence representation S_i, for $1 \leq i \leq k$. By applying theorem 38 on each pair of sequences and composing the results, we have $n_1 \geq n_2 \cdots \geq n_k$ and $s_j^i \leq (r_j^{i-1} + s_j^{i-1}) - r_j^i$ for $1 \leq j < n_i$, $1 < i \leq k$.
$\Leftarrow$: For $1 \leq i \leq k$, let $\mathsf{S}_i = \langle s_1^i, \cdots, s_{n_1}^1 \rangle$ be k sequences $(k > 2)$ and let the elements in the sequence satisfy the conditions given in the theorem. Let $\mathbf{MF}_{\mathsf{S}_i}$, $1 \leq i \leq k$, be the maximal itemset collections with sequence representation S_i, constructed by using theorem 27. Unlike theorem 38, finding $\pi_i^{\min}$ is not straightforward. Let $\mathsf{msup}(\mathsf{DB})$ be the maximum over all the support values of individual items in DB at that stage. The database DB is constructed in k stages as shown in figure 5. From the algorithm, each itemset in $\mathbf{MF}_{\mathsf{S}_j}$ is replicated enough times and added to the database, to make it the maximal frequent collection obtained at minimum support $\pi_j^{\min}$ and hence, by construction $\mathbf{MF}_{\mathsf{S}_j}$ has the sequence representation S_j. □

Example 40. *Consider an example where three sequences $\mathsf{S}_1 = \langle 2, 3, 3, 4 \rangle$, $\mathsf{S}_2 = \langle 2, 3, 3 \rangle$ and $\mathsf{S}_3 = \langle 1, 3, 2 \rangle$ are given. The problem is to generate a database DB and three values of minimum support $\pi_1^{\min} < \pi_2^{\min} < \pi_3^{\min}$ such that the maximal frequent itemsets obtained by mining DB at minimum support level $\pi_i^{\min}$ has the sequence representation S_i (and hence the distribution given by S_i) for $i = 1, 2, 3$. To do so, the maximal frequent itemset collections corresponding to the three sequences are constructed using theorem 27. Let the collections be denoted as MF_1, MF_2 and MF_3 respectively. By the construction procedure, collection MF_ihas a sequence representation S_i, for $i = 1, 2, 3$. In the first step, every itemset in MF_1 is added as a transaction to the database. The resulting database is denoted $DB1$. Then the support of all the items in $DB1$ is computed and the maximum of all these values is denoted $\mathsf{msup}(DB1)$. At this stage, the absolute support of any item in $DB1$ is at most $\mathsf{msup}(DB1)$. Hence, by replicating itemsets in MF_2 one more than this value, we ensure that at absolute support $1 + \mathsf{msup}(DB1)$ the maximal itemsets obtained by mining the database is MF_2 which has the sequence representation S_2 (and hence the distribution). The process is repeated once more for the third sequence and the resulting database is the final database. For computing the three levels of minimum support, the msup values computed at each stage are used. The values of minimum support at which each distribution is obtained is calculated as $\frac{1+\mathsf{msup}(DB_i)}{|DB|}$, where DB_i is the database after processing sequence i. This is illustrated in figure 6. After processing sequence S_1, the database contains 12 transactions, one for each itemset in MF_1. The msup value at this stage is 7 and hence, each itemset in MF_2 is replicated $1 + 7 = 8$ times. At this stage, the database contains 76 transactions and the msup value is 39. In the final stage, the itemsets in MF_3 are replicated $1 + 39 = 40$ times to*

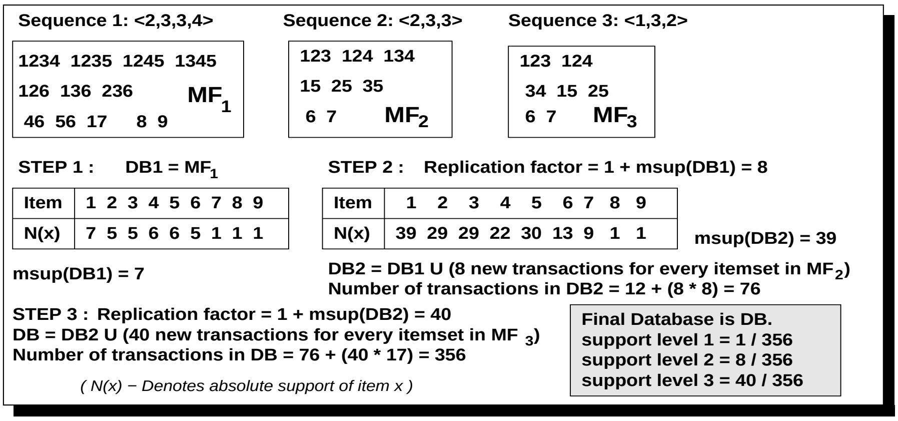

Sequence 1: <2,3,3,4>
Sequence 2: <2,3,3>
Sequence 3: <1,3,2>
1234 1235 1245 1345
126 136 236
46 56 17 8 9
MF1
123 124 134
15 25 35
6 7
MF2
123 124
34 15 25
6 7
MF3
STEP 1 : DB1 = MF1
Item 1 2 3 4 5 6 7 8 9
N(x) 7 5 5 6 6 5 1 1 1
msup(DB1) = 7
STEP 2 : Replication factor = 1 + msup(DB1) = 8
Item 1 2 3 4 5 6 7 8 9
N(x) 39 29 29 22 30 13 9 1 1
msup(DB2) = 39
DB2 = DB1 U (8 new transactions for every itemset in MF2)
Number of transactions in DB2 = 12 + (8 * 8) = 76
STEP 3 : Replication factor = 1 + msup(DB2) = 40
DB = DB2 U (40 new transactions for every itemset in MF 3)
Number of transactions in DB = 76 + (40 * 17) = 356
(N(x) − Denotes absolute support of item x)
Final Database is DB.
support level 1 = 1 / 356
support level 2 = 8 / 356
support level 3 = 40 / 356

Figure 6: Generating a database for three given sequences

give the final database with 356 *transactions. The minimum support levels at which the three maximal itemset collections are obtained as output are* $\frac{1}{256}$, $\frac{8}{256}$ *and* $\frac{40}{256}$ *respectively.*

6.1 Analysis of the Database Generation Procedure

The K-DbGen algorithm was implemented in C++ following the approach given in figure 5. Given k user supplied sequences satisfying the constraints of theorem 39, this implementation uses procedure ColexListing in figure 7 and minor variants to compute $\mathbf{MF}_{\mathsf{S}_i}$, for $1 \leq i \leq k$. The algorithm generates the database in k stages (iterations of step 3 in K-DbGen). The initial stage generates the database with a replication factor of 1. Support counts are maintained for each individual item in the database generated upto any given stage. At the end of stage 1, the support counts are used to compute msup and replication factor values for stage 2. At any given stage j, $(2 \leq j \leq k)$, the replication factor value is used to generate transactions for every maximal itemset in $\mathbf{MF}_{\mathsf{S}_j}$ and the transactions are written to the database. Support counts are updated for each individual items during this process. After the generation process, a linear scan of the support values of individual items is performed to compute msup and replication factor values for stage $j + 1$. An illustrative example of database construction using K-DbGen is given in figure 6. Procedure ColexListing generates all the k-subsets of $(\mathbf{n})$ in colex order. A small modification to the input specification and the termination condition is used to adapt this procedure to generate $\mathbf{MF}_{\mathsf{S}_j}$ in K-DbGen.

The complexity of procedure ColexListing can be analyzed as follows. Step 1 takes $O(k)$ time. Step 2 and 3 take constant time (assuming that itemsets are stored in an array). Step 4 takes $O(k)$ time for each iteration. Hence the overall complexity of procedure ColexListing is $O(zk)$ where z is the number of k-itemsets that are output (hence linear in output size). The complexity of algorithm K-DbGen can be analyzed as follows. Step 1 takes $O(|\mathsf{DB}_1|)$ time since DB_1 is identical to $\mathbf{MF}_{\mathsf{S}_1}$ which is generated using procedure ColexListing. Step 2 takes constant time. Each iteration of step 3 takes $O(a_j.|\mathbf{MF}_{\mathsf{S}_j}|) = O(|\mathsf{DB}_j|)$ time. Hence the complexity of K-DbGen is $O(|\mathsf{DB}|)$ where DB is the database output by the algorithm (linear in the output database size).

Even though algorithm K-DbGen is simple to implement, it has some drawbacks. It can be observed from algorithm K-DbGen that the msup values are monotonically increasing during every stage of database generation. This could lead to an inflation in the number of transactions in the database (depending on the number of sequences and the number of itemsets in each sequence), as discussed in the following example.

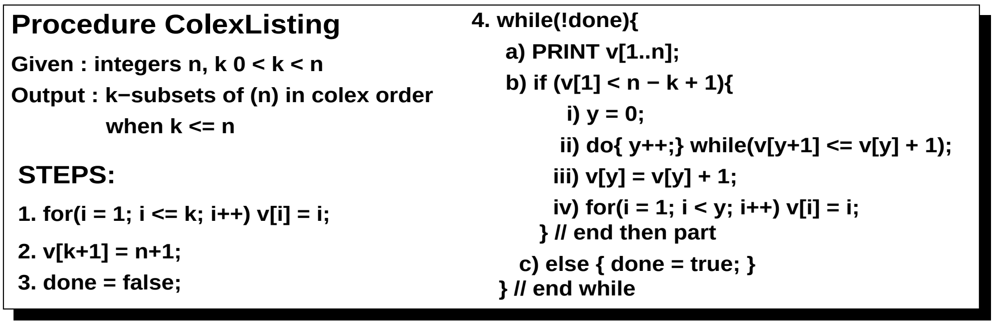
As an example, consider the three sets of sequences shown in figure 8. These sequences represent maximal frequent itemset distributions found in real datasets at increasing levels of minimum support (in this case CHESS, CONNECT and MUSHROOM [3]).

Figure 9 lists the generation statistics for each stage of database generation, for each of the examples in figure 8. It can be observed that the number of transactions in the generated databases are extremely large due to an inflation in the replication factor at each stage. In the case of the example from CHESS and CONNECT sequences, the number is in the order of tens to hundreds of billions of transactions while in the case of sequences from the MUSHROOM dataset, the number is in the order of 10^{12} transactions.

Thus, naively generating databases by physical replication can be prohibitively expensive due to large replication factors that occur in practice. One way to avoid the overhead of physical replication is to logically replicate transactions using **transaction maps**, which are defined as follows.

Definition 41. [Transaction Map] *A* transaction map *is a triple* $\langle C, i, T \rangle$*, where* $C > 0$ *is a positive integer,* i *is a nonnegative integer and* $T \subset \mathcal{I}$ *is an itemset. The nonnegative integer* i *is a unique identifier called the transaction map identifier. The positive integer* C *denotes a count value which gives the number of times the itemset* T *is replicated. A transaction map database is a sequence of transaction maps where each map in the sequence has a unique transaction map identifier.*

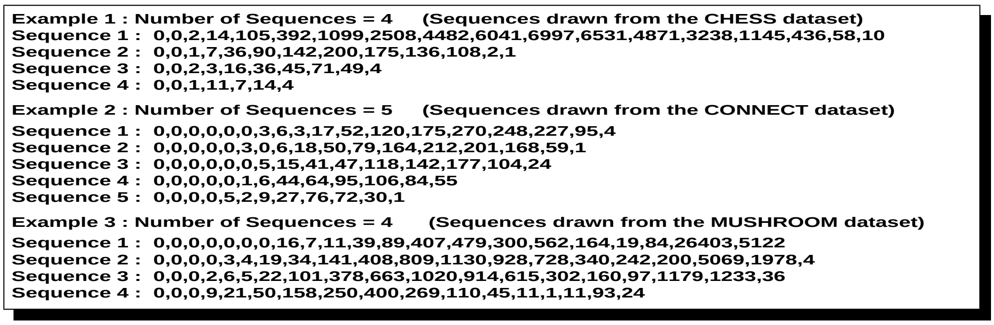
Figure 10 shows an example of a transaction map database for the three sequences in example 40. Generating transaction map databases reduces the overhead in storage and time by eliminating the costly physical replication factor overhead. Figure 9 shows the number of transaction maps generated for the examples in figure 8. It can be observed that the storage savings is significant (upto a factor of 10^8). The number of transaction maps that are output by K-DbGen is $\sum_{j=1}^{k} |\mathbf{MF}_{\mathsf{S}_j}| = \sum_{j=1}^{k} \sum_{i=1}^{n_j} s_{j,i}$, where $\mathsf{S}_1, \mathsf{S}_2, \ldots, \mathsf{S}_k$ are k given sequences of nonnegative integers, such that $\mathsf{S}_j =$



```
Procedure ColexListing
Given : integers n, k 0 < k < n
Output : k−subsets of (n) in colex order
            when k <= n

STEPS:
1. for(i = 1; i <= k; i++) v[i] = i;
2. v[k+1] = n+1;
3. done = false;
4. while(!done){
     a) PRINT v[1..n];
     b) if (v[1] < n − k + 1){
          i) y = 0;
          ii) do{ y++;} while(v[y+1] <= v[y] + 1);
          iii) v[y] = v[y] + 1;
          iv) for(i = 1; i < y; i++) v[i] = i;
        } // end then part
     c) else { done = true; }
   } // end while
```

Figure 7: Prints out the k-subsets of the set $\{1, 2, \ldots, n\}$ in Colex Order



```
Example 1 : Number of Sequences = 4     (Sequences drawn from the CHESS dataset)
Sequence 1 :  0,0,2,14,105,392,1099,2508,4482,6041,6997,6531,4871,3238,1145,436,58,10
Sequence 2 :  0,0,1,7,36,90,142,200,175,136,108,2,1
Sequence 3 :  0,0,2,3,16,36,45,71,49,4
Sequence 4 :  0,0,1,11,7,14,4

Example 2 : Number of Sequences = 5     (Sequences drawn from the CONNECT dataset)
Sequence 1 :  0,0,0,0,0,0,3,6,3,17,52,120,175,270,248,227,95,4
Sequence 2 :  0,0,0,0,0,3,0,6,18,50,79,164,212,201,168,59,1
Sequence 3 :  0,0,0,0,0,0,5,15,41,47,118,142,177,104,24
Sequence 4 :  0,0,0,0,0,1,6,44,64,95,106,84,55
Sequence 5 :  0,0,0,0,5,2,9,27,76,72,30,1

Example 3 : Number of Sequences = 4      (Sequences drawn from the MUSHROOM dataset)
Sequence 1 :  0,0,0,0,0,0,0,16,7,11,39,89,407,479,300,562,164,19,84,26403,5122
Sequence 2 :  0,0,0,0,3,4,19,34,141,408,809,1130,928,728,340,242,200,5069,1978,4
Sequence 3 :  0,0,0,2,6,5,22,101,378,663,1020,914,615,302,160,97,1179,1233,36
Sequence 4 :  0,0,0,9,21,50,158,250,400,269,110,45,11,1,11,93,24
```

Figure 8: Three Examples drawn from Real Datasets for Database Generation

Generation Statistics for Sequences from the CHESS dataset

STAGE	msup(DB)	Replication Factor	Number of Transactions	Number of Transaction Maps
1	-	1	37929	37929
2	23652	23653	21278323	38827
3	13857152	13857153	3152994901	39053
4	1938298517	1938298518	74870040067	39090

Generation Statistics for Sequences from the CONNECT dataset

STAGE	msup(DB)	Replication Factor	Number of Transactions	Number of Transaction Maps
1	-	1	1220	1220
2	1111	1112	1069852	2181
3	906089	906090	610868422	2854
4	595718053	595718054	271662582992	3309
5	2073330731	2073330732	731942005496	3531

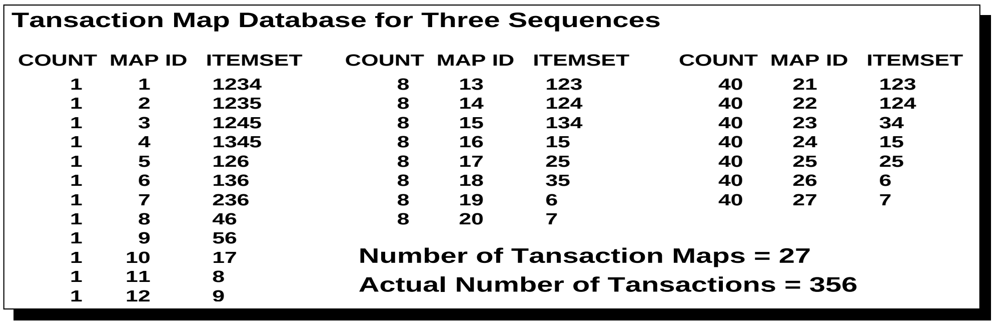
Generation Statistics for Sequences from the MUSHROOM dataset

STAGE	msup(DB)	Replication Factor	Number of Transactions	Number of Transaction Maps
1	-	1	33702	33702
2	29643	29644	356858530	45739
3	316684050	316684051	2132590573913	52472
4	2047308738	2047308739	5105282862941	53924

Figure 9: Table showing values at each stage of database generation for the examples in figure 8

$\langle s_{j,1}, s_{j,2}, \ldots, s_{j,n_j} \rangle$ (i.e, there are n_j nonnegative integers in sequence S_j).

Current mining algorithms [2, 16, 15, 12] can be altered with little overhead to mine transaction map databases. In the case of level wise mining algorithms like Apriori (and variants), a data structure is used to store candidate patterns and each time a transaction is processed, the counts of the candidate patterns contained in the transaction is incremented by 1. In the case of a transaction map, the increment is done using the transaction map count instead of incrementing by 1. This is a modification to the count step. The same approach also applies to FP-tree methods where the transactions are used to increment support counts for frequent itemsets in the structure. In the case mining algorithms that use tidlist intersection to count support (e.g. Eclat), the transaction map ids are intersected and instead



Tansaction Map Database for Three Sequences

COUNT	MAP ID	ITEMSET	COUNT	MAP ID	ITEMSET	COUNT	MAP ID	ITEMSET
1	1	1234	8	13	123	40	21	123
1	2	1235	8	14	124	40	22	124
1	3	1245	8	15	134	40	23	34
1	4	1345	8	16	15	40	24	15
1	5	126	8	17	25	40	25	25
1	6	136	8	18	35	40	26	6
1	7	236	8	19	6	40	27	7
1	8	46	8	20	7			
1	9	56						
1	10	17						
1	11	8						
1	12	9						

Number of Tansaction Maps = 27

Actual Number of Tansactions = 356

Figure 10: Transaction Map Database for the sequences in Example 40

of the length of the intersection, the sum of the counts associated with each transaction map id in the intersection gives the support. This causes little overhead, as only a vector of counts need to be maintained with each transaction map id in order to obtain their count values. This makes it practical for algorithms to mine transaction map databases.

Figure 11 summarizes the results of the database generation procedure on the examples in figure 8. The experiments were carried out on a Dell Inspiron 8100 laptop with a $1GHz$ Intel Pentium III processor with 256 MB of RAM running SuSE Linux 8.0. It can be observed that the generation procedure takes under 1 second in generating the transaction map databases for these examples.

7. CONCLUSIONS AND FUTURE WORK

The distribution of frequent and maximal frequent itemsets in a transaction database determines the resource requirements of extant pattern mining algorithms. Hence, empirical performance comparisons are needed for informed algorithm selection among numerous alternatives. This requirement motivated our study that serves as a long needed step in characterizing maximal/frequent itemset length distributions in databases.

In this paper, we addressed two crucial issues in frequent pattern mining problems. We characterized the length distribution of frequent and maximal frequent itemset collections. Tight bounds were developed to answer questions on whether a given sequence representation is a feasible length distribution of a frequent or a maximal frequent itemset collection. We also characterized the conditions under which one can embed such distributions in a database. This has direct application in generating benchmark databases to compare current association mining algorithms.

The paper also discusses issues related to database generation given k distributions of maximal itemsets and provides a technique to prevent an expensive blowup in the number of transactions through transaction maps. However, the generation procedure has some limitations. Currently, all transactions or transaction maps that are added to the database are maximal frequent itemsets. This constraint causes database sizes to be huge. In practice, entire transactions are themselves not maximal frequent itemsets but contain them. The other observation is that all itemsets in a given sequence have the same replication factor and this factor is chosen conservatively. We plan to explore how the number of transactions can be reduced by relaxing these constraints.

Our ongoing study is also exploring the various ramifications of the problem. The algorithm presented in this paper generates a database using the minimum number of items but not necessarily the minimum number of transactions. Work is in progress to provide generation techniques (i) that use the minimum number of transactions and (ii) that allows for use of relaxed constraints on the number of items. Such approaches have potential applications in generation of synthetic databases which preserve the distributions of patterns upto a certain number of support levels and yet offer privacy preservation with respect to disclosure of information. Answers to these questions will generate databases that are more practical in terms of the pattern distributions that can be found in real datasets.

Other applications of feasible distributions include how one can exploit them in a mining algorithm. Goethals, et. al. [8] already demonstrated one such method where a tight bound on the expected space of candidate patterns was derived and used in a level-wise mining algorithm. Can other methods that perform heuristic search instead of level-wise search benefit from these results? Finally, there is the issue of extension of these bounds to other pattern spaces apart from frequent and maximal frequent itemset collections.

8. ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for their valuable suggestions and feedback.

9. REFERENCES

[1] R. Agrawal, C. Aggarwal, and V. Prasad. Depth First Generation of Long Patterns. In *7th Int'l Conference on Knowledge Discovery and Data Mining*, Aug. 2000.

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

[3] S. Bay. *The UCI KDD Archive (kdd.ics.uci.edu).* University of California, Irvine. Department of Information and Computer Science.

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

[5] B. Bollobás. *Combinatorics.* Cambridge University Press, 1986.

[6] S. Brin, R. Motwani, J. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for

Source	# maps	# trans. in dataset	Time	Map Database Size (in Bytes)
chess	39090	3196	0.313s	539442
connect	3531	67557	0.029s	174438
mushroom	53924	8124	0.619s	3551048

Figure 11: Table showing statistics of transaction map databases generated using Algorithm K-DbGen

market basket data. In *ACM SIGMOD Conf. Management of Data*, May 1997.

[7] D. Burdick, M. Calimlim, and J. Gehrke. MAFIA: a maximal frequent itemset algorithm for transactional databases. In *Intl. Conf. on Data Engineering*, Apr. 2001.

[8] B. Goethals, F. Geerts, and J. V. den Bussche. A tight upper bound on the number of candidate patterns. In *1st IEEE International Conference on Data Mining*, Nov. 2001.

[9] K. Gouda and M. J. Zaki. Efficiently mining maximal frequent itemsets. In *1st IEEE Int'l Conf. on Data Mining*, Nov. 2001.

[10] D. Gunopulos, R. Khardon, H. Mannila, and H. Toivonen. Data mining, hypergraph transversals, and machine learning. In *16th ACM Symp. Principles of Database Systems*, May 1997.

[11] J. Han and M. Kamber. *Data Mining: Concepts and Techniuqes*. Morgan Kaufmann Publishers, San Francisco, CA, 2001.

[12] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *ACM SIGMOD Conf. Management of Data*, May 2000.

[13] G. Katona. A theorem of finite sets. In P. Erdos and G. Katona, editors, *Theory of Graphs*, pages 187–207. Akademiai Kiado, Budapest, 1968.

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

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

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

[17] Z. Zheng, R. Kohavi, and L. Mason. Real world performance of association rule algorithms. In *7th Intl. Conf. on Knowledge Discovery and Data Mining*, Aug. 2001.