Average Case Performance of the Apriori Algorithm

Paul W. Purdom Dirk Van Gucht
Computer Science Department, Indiana University

Dennis P. Groth
School of Informatics, Indiana University

Abstract:

The failure rate of the Apriori Algorithm is studied both analytically and experimentally. The time needed by the Apriori Algorithm is determined by the number of item sets that are output (successes: item sets that occur in at least k baskets) and the number of item sets that are counted but not output (failures: item sets where all subsets of the item set occur in at least k baskets but the full set occurs in less than k baskets). The number of successes is a property of the data, no algorithm that is required to output each success can avoid doing work associated with the successes. The number of failures is a property of both the algorithm and the data.

This paper studies the failure rate using both analytical and experimental approaches. The analytical work is done on a simple model where each shopper buys at random. The experimental work is done with three data sets that are more like the usual applications of the algorithm, the buying patterns of the various shoppers are highly correlated.

We find that under a wide range of conditions the performance of the Apriori Algorithm is almost as bad as is permitted under sophisticated worst-case analyses. In particular, there is usually a bad level with two properties: (1) it is the level where nearly all of the work is done and (2) nearly all candidate item sets are failures. Let l be the level with the most successes, and let the number of successes on level l be approximately $\binom{m}{l}$ for some m. Then, typically, the Apriori Algorithm has total output proportional to approximately $\binom{m}{l}$ and total work proportional to approximately $\binom{m}{l+1}$. In addition m is usually much larger than l so the ratio of work to output is proportional to approximately $m/(l+1)$.

1. Introduction

The Apriori Algorithm [2, 3, 7, 17] solves the frequent item sets problem. The algorithm analyzes a data set to determine which combination of items occur together frequently. Consider a store with $|I|$ items where b shoppers each have a single basket. Each shopper selects a set of items for his basket. The input to the Apriori Algorithm is a list giving the set of items in each basket. For a fixed threshold k, the algorithm determines which sets of items are contained in at least k of the b baskets.

The Apriori Algorithm is at the core of various algorithms for data mining problems. The best known such problem is the problem of finding the *association rules* that hold in a basket-items relation [2, 3, 17, 21]. Other data mining problems based on the Apriori Algorithm are discussed in [7, 16, 17, 19, 22, 23].

Let J_l be a subset of size $l \geq 1$ that is selected from the $|I|$ items. For J_l, define J_l^{-h} to be the set obtained from J_l by omitting element h (a set of size $l-1$ when h is in J_l). The key idea of the Apriori Algorithm is that the set J_l can not possibly have k occurrences unless each of the sets J_l^{-h} (h in J_l) has k occurrences. Since the algorithm considers possible sets in order of their size, it has already gathered the information about all the sets of size $l-1$ before it considers sets of size l.

For each set J_l the algorithm verifies from its internal tables that each of the sets J_l^{-h} with h in J_l occurs at least k times (l cases to verify). We call this the *candidacy test*. A set J_l that passes this test is called a *candidate*. For those sets J_l that are candidates, the algorithm examines the data (basket contents) to count the number of baskets that contain each J_l thereby determining which set of items occurs in at least k baskets. This counting and comparing with the threshold is called the *frequency test*. A set that passes the frequency test is called a *success* (or a *frequent item set*). The algorithm remembers successes at level l to generate candidates at level $l+1$.

In the best case, the candidacy test always correctly predicts the result of the frequency test, and the amount of time spent by the Apriori Algorithm is essentially the amount of time spent verifying that all the output sets should be output. A naive upper bound comes from the fact that each candidate is based on a previous success. Each success can lead to at most $|I|$ candidates. Many previous papers have focused on the total amount of work done by the Apriori Algorithm without concern as to the amount of output generated, but such studies can be misleading in that the amount of output is the main factor that determines how much work an Apriori-like algorithm will do, and the amount of output is a feature of the problem instance rather than of the algorithm. The simple lower and upper bound analyses say that the ratio between the number of candidates and the number of successes (i.e, the output) is between 1 and $|I|$.

The worst-case time needed by the Apriori Algorithm is polynomial in the sum of the size of the input plus output. Since the worst-case output is exponentially larger than the input, the worst-case time needed can be exponental in the size of the input. If every shopper buys every item, the algorithm must output each subset of the I items. A closely related problem that is NP-complete is to determine whether or not there are any sets of size l that occur k times, is NP-complete because the Balanced Complete Bipartite Subgraph Problem [10] reduces to it. The appendix has the details of proofs for this and many other statements. The appendix also has the proofs for the more complex equations in this paper.

The analytical part of this paper has an average-time analysis of the Apriori Algorithm under a parameterized probability model where the baskets are filled at random. Each basket has probability p of containing each item, independent of the other items and independent of the other baskets. This is the same probability model that has previously been used to consider the probability that is a set is a success [3,18,22]. In real life, the Apriori Algorithm is used to analyze data that is more complex. Presumably, no one is interested in running the algorithm on truly random data. Rather, one is interested in the way in which the data differs from random. None-the-less, we believe that analysis with this simple probability model brings out the main features of the performance of the algorithm. In principle, the techniques in this paper can be applied to more complex probability models of shopping. The challenge is to carry out the resulting calculations so that one can understand the implications of the formulas that result when the analysis is done on more general probability models.

With our probability model we calculate two quantities:

1. *Success rate*: the probability that a set is a success (i.e., passes the frequency test) and test and
2. *Failure rate*: the probability that a set is a candidate (i.e., passes the candidacy test) but not a success (i.e., fails the frequency test).

Notice that the success rate is a property of the probability model, not the algorithm. All correct algorithms will have the same success rate. The Apriori Algorithm never believes that an item set occurs k times without verifying the fact by counting occurrences in the data base. For algorithms that use this approach, the success rate represents unavoidable work. The Apriori Algorithm is clever in trying to reduce the failure rate. The failure rate represents work that one might hope to avoid.

It is not logically necessary that an algorithm verify occurrences by explicit counting. One alternative algorithm uses ideas that are the complement of those used by the Apriori Algorithm. The key idea for this complementary algorithm is that if some superset of a set J occurs at least k times, then so does set J [1, 5, 4, 13]. Also, when there is a total of b baskets, b_A baskets with item A, and b_B items with item B, the Apriori Algorithm is not aware that there must be at least $2b - b_A - b_B$ baskets that contain both items A and B [12]. Similar ideas are explored in [8].

The advantage that comes from using a parameterized probability model is that one can study the performance of the algorithm under a wide range of conditions. While the later sections contain many mathematical details, the main conclusions are fairly simple. For most values of the parameters, random data results in a high success rate (essentially 1) for small values of l, with a sudden switch to a low success rate (essentially 0) for large values of l. When l is below the level of the switch, the number of candidates is just below $\binom{|I|}{l}$. Almost all of the candidates are successes. When l is above the level of the switch, almost all candidates are failures, and the number of candidates decreases rapidly. The level for the switch depends on the parameters.

It is interesting to compare our average-case analysis with the worst-case bound of Geerts, Goethals, and Van de Bussche [11, Theorem 1]. They represent the actual number of successes on level l as

$$\sum_{0 \le i \le r} \binom{m_{l,l-i}}{l-i} \tag{1}$$

for appropriate $m_{l,i}$, where $m_{l,i} > m_{l,i-1}$ for $i - 1 \ge r$. They give the following upper bound on the number of candidates

$$\sum_{0 \le i \le r} \binom{m_{l,l-i}}{l-i+1}. \tag{2}$$

This bound is exact for some distributions, including one that results in no failutes. Experiments show that in many cases this upper bound is close to the exact value. Eq. 2 makes use of the number of successes on the current level to bound the number of candidates on the next level.

At first sight, the approach of Theorem 1 of [11] looks quite different from the approach in this paper. However, in those cases where the number of candidates on level l is exactly $\binom{m_{l,l}}{l}$ for some $m_{l,l}$ there are just two differences. First, [11] only considers items that might still be active during the current level of the Apriori Algorithm ($m_{l,l}$). This is their main insight. Second, the upper limit corresponds to setting our p to 1. The work in [11] shows that the number of candidates on level l can never be more than $\binom{m_{l,l}}{l+1}$. If $m_{l,l}$ is much bigger than l, the ratio of candidates on level $l+1$ to candidates on level l is approximately $m(l,l)/(l+1)$. When the number of candidates can not be represented as an appropriate binomial, there are additional technical differences between the approach in [11] and in this paper, but they are not significant to a qualitative understanding of the situation.

In the experimental part of the paper, we compared the predictions of the random basket theory with three data sets: (1) synthetic data generated by the generator from the IBM Quest Research Group [14], (2) U. S. Census data, using Public Use Microdata Samples (PUMS) (the same sample that was used by [6, 7] and processed it in the same way), and (3) a web data set [25].

The analytical results are for baskets that are filled in an uncorrelated way. For such data, the Apriori algorithm will have essentially no failures on the first few levels (depending on the value of p) followed by an extremely high failure rate on the remaining levels. For interesting values of the parameters, if l_* is the level with the most frequent item sets, then the total number of item sets will be only slightly larger than the number on the level with the most item sets. For this level and all later levels, almost every candidate will be a failure. Thus, if we have approximately $\binom{|I|}{l_*}$ frequent item sets to output, the algorithm will do approximately $\binom{|I|}{l_*+1}$ work. If p is small, the l_* will be much less than $|I|$, and the ratio of work to output will be approximately a constant times $|I|/(l_*+1)$.

For the experimental data, correlations are extremely important. When thinking about the experimental data, it is useful to imagine that the data is described by effective values for $|I|$, b, and p depend on the level. To illustrate this idea, consider a situtation where 1000 shoppers buy at random buying an average of one of 10,000 different items. Suppose an additional 100 buyers all buy the same 10 items (a highly corelated subset of buyers). Based on simple statistics that an Apriori Algorithm might collect, level 1 looks very much like 1100 buyers buying from 10,000 items with a probability of about 2/10,000. By level 5, however, the data looks similar to 100 buyers buying from 10 items with a probability of 1. In the presence of correlations, you might expect that as the level increases, the effective value for $|I|$ will decrease, that for b will decrease, and that for p will increase.

The analytical result for uncorrelated data says that the critical level occurs when l is such that k is about the same size of bp^l. For correlated data, even with the effective b and p varying, unless the variations are strong, there is still likely to be just one critical value l_* such that the success rate is high on level l_* but low on level l_*+1. When we look at the experimental results, we indeed find (in almost all cases), that the total amount of work done is in agreement with the analytical results, provided one change is made. The analytical theory says that the number of candidates on that level should be $\binom{|I|}{l_*+1}$. Replace this with

$\binom{m_{l_*,l_*}}{l_*+1}$, where m_{l_*,l_*} is choosen so that $\binom{m_{l_*,l_*}}{l_*}$ is the number of candidates on level l_* (approximate when the number of candidates is not exactly a binomial coefficent of this form). This comes from the upper bound result of [11]. Thus, the conclusion from the experimental data is that you can understand the total work done by the Apriori Algorithm if you take the worst-case number of candidates from [11] along with the result from the analytical theory for uncorrelated data that the failure rate is nearly always high on the level where most of the work is done. The ratio of work done by the Apriori Algorithm is proportional to the ratio of total number of candidates to the total number of successes. Thus, when the level with the most output is l_* and when l_* is much less than m_{l_*,l_*}, the ratio of work to output will be approximately proportional to $m_{l_*,l_*}/(l_*+1)$.

2. The Apriori Algorithm

The Apriori Algorithm does the following computation:

Apriori Algorithm:

Step **1.** For l from 1 to $|I|$ do

Step **2.** For each set J_l such that for each $h \in J_l$ the set J_l^{-h} occurs in at least k baskets do

Step **3.** Examine the data to determine whether the set J_l occurs in at least k baskets. Remember those cases where the answer is 'yes'.

For typical data sets, a careful implementation of the Apriori Algorithm will spend most of its time accessing the data base (the list of basket contents). The implementation should exit the l loop early if there are no 'yes' answers for some value of l. It should consider on level l only those sets that are formed from sets that passed the frequency test on level $l-1$. In addition, no set of size l should be generated more than once. The sets can be generated by assigning an order to the items and extending each set S on level $l-1$ only with items that are greater than the largest item in S. *Assuming unit time for hash table look-ups* (for looking up various subsets of the extended S) the algorithm can do the work for a single candidate set on level l in time bounded by a constant times $l+1$. See [2] for more discussion of the techniques used in good implementations.

3. Best and worst cases

We use the number of candidates as a proxy for the amount of computing that the Apriori Algorithm does. Let N_S be the number of item sets that are successes. Let N_F be the number of item sets that are candidates but not found to be successes. Then, the total work is proportional to $N_S + N_F$. Of this, N_S represents work that must be done by any algorithm that outputs every frequent item set and it is a property of the data, not of the algorithm. N_F represent avoidable work. For this class of algorithms, the ratio $N_S/(N_S + N_F)$ is the natural measure of algorithm efficiency.

The lower limit on the amount of work done by the Apriori Algorithm is N_S. This occurs for problem instances where every candidate is actually a frequent item set.

Now consider an upper limit on the amount of work. Let $N_S(l+1) + N_F(l+1)$ be the number of candidates which are counted on level $l+1$, and let $N_S(l)$ be the number of frequent item sets on level l. Consider the graph where each candidate on level $l+1$ (each J_{l+1}) is connected to each of its associated frequent items sets (J_{l+1}^{-h}). Thus, there are $[N_S(l+1) + N_F(l+1)](l+1)$ arcs. Each of the $N_S(l)$ frequent item sets on level l is connected to at most $|I| - l$ candidates on level $l+1$. Indeed, if we use $|I_l|$ to be the number of items that occur among the frequent item sets of level l, there are at most $|I_l| - l$ connections from a single level l frequent item set. Thus, we have

$$[N_S(l+1) + N_F(l+1)](l+1) \leq N_S(l)(|I_l| - l), \tag{3}$$

$$[N_S(l+1) + N_F(l+1)] \leq \left(\frac{|I_l| - l}{l+1}\right) N_S(l). \tag{4}$$

This gives

$$\sum_{0 \leq l \leq |I_l|-1} \left(\frac{|I_l| - l}{l+1}\right) N_S(l) \tag{5}$$

as an upper bound on the amount of work. In many cases, $N_S(l)$ increases rapidly for small l and then suddenly drops rapidly to zero. In such cases, the largest term in this sum is a good approximation to its total value.

We see that the Apriori Algorithm is rather efficient in the class of algorithms that output every frequent item set. The total work is never more than a factor $|I|$ larger than N_S, the least amount of work that could possibly be done by any algorithm in the class. When most of the work is concentrated on level $l+1$, the amount of work is better than this product by a factor of $l+1$.

4. Average case

We now start an exact computation of the average case performance of the Apriori Algorithm for the case when the baskets are filled at random. We eventually show that for most values of the parameters, the average performance is not significantly better than that suggested by the worst-case analysis (eq. 5).

Let S_l be the probability that the set consisting of items 1 to l is a frequent item set, and F_l be the probability that the same set is a candidate but fails to be a frequent item set. Since each basket is filled randomly, any other set of l items has the same probability of success and failure. The expected number of successes is

$$\sum_{1 \leq l \leq |I|} \binom{|I|}{l} S_l, \tag{6}$$

and the expected number of failures is

$$\sum_{1 \leq l \leq |I|} \binom{|I|}{l} F_l. \tag{7}$$

The number of item sets for which the basket data is examined is

$$\sum_{1\le l\le |I|} \binom{|I|}{l}(S_l + F_l). \tag{8}$$

Under the above assumptions, the running time is bounded by a constant times

$$\sum_{1\le l\le |I|} (l+1)\binom{|I|}{l}(S_l + F_l). \tag{9}$$

Define the following conditions with respect to a single basket:

- M_0: the basket has all the items 1 to l and
- M_h $(1 \le h \le l)$: the basket has all items from 1 to l except that it does *not* have item h.

These conditions are disjoint; each basket obeys at most one of the conditions M_h, $0 \le h \le l$.

The probability that a randomly filled basket obeys condition M_0 is

$$P(l) = p^l. \tag{10}$$

The probability that a randomly filled basket obeys condition M_h (for any h in the range 1 to l) is

$$Q(l) = p^{l-1}(1-p). \tag{11}$$

Note that

$$P(l-1) = P(l) + Q(l). \tag{12}$$

It is worth noticing in passing, that if one wants a model of shoppers that are independent of each other, but which have more complex shopping behavior than assumed in this paper, the key step is to change the formulas for computing $P(l)$ and $Q(l)$. Our results that are expressed in terms of P and Q (but not those expressed in terms of p) would still hold for these more complex shoppers.

The probability that at least k baskets obey condition M_0 is

$$S_l = \sum_{j\ge k}\binom{b}{j}[P(l)]^j[1-P(l)]^{b-j} = 1 - \sum_{j<k}\binom{b}{j}[P(l)]^j[1-P(l)]^{b-j}. \tag{13}$$

The probability that j_0 baskets obey condition M_0, j_1 baskets obey condition M_1, ..., j_l baskets obey condition M_l, and the remaining $b - j_0 - \cdots - j_l$ baskets do not obey any of the conditions is

$$\binom{b}{j_0,\ldots,j_l, b-j_0-\cdots-j_l}[P(l)]^{j_0}[Q(l)]^{j_1+\cdots+j_l}[1-P(l)-lQ(l)]^{b-j_0-\cdots-j_l}, \tag{14}$$

where the multinomial coefficient is the number of ways to arrange b distinct baskets into $l+1$ sets, where set 0 has j_0 baskets, ..., set l has j_l baskets, and $b - j_0 - \cdots - j_l$ baskets are not in any of the $l+1$ sets.

The item set $\{1, \ldots, l\}$ is a candidate l if and only if each condition M_h $(1 \le h \le l)$ is satisfied by at least k baskets. Condition M_1 is passed when $j_0 + j_1 \ge k$. In other words, to pass condition M_1, we must obtain at least k when we add the number of baskets that have all the items from 1 to l to the number of baskets that do not have item 1 but do have all the items from 2 to l. Condition M_2 is passed when $j_0 + j_2 \geq k$, etc. Thus, item set $\{1, \ldots, l\}$ is a candidate in just those cases where the conditions

$$j_0 + j_1 \geq k,\ j_0 + j_2 \geq k,\ \ldots,\ j_0 + j_l \geq k \tag{15}$$

are all true. Thus, the probability that the set $\{1, \ldots, l\}$ is a candidate is the above probability (eq. 14) summed over those cases that satisfy the conditions (eq. 15),

$$C_l = \sum_{\substack{j_0 \\ j_1 \geq k - j_0 \\ j_2 \geq k - j_0 \\ \cdots \\ j_l \geq k - j_0}} \binom{b}{j_0, \ldots, j_l, b - j_0 - \cdots - j_l} [P(l)]^{j_0} [Q(l)]^{j_1 + \cdots + j_l} [1 - P(l) - lQ(l)]^{b - j_0 - \cdots - j_l}. \tag{16}$$

Since the set $\{1, \ldots, l\}$ either does or does not occurs in at least k baskets, the probability that the item set $\{1, \ldots, l\}$ is a candidate but not a frequent item set is

$$\begin{aligned} F_l &= C_l - S_l \\ &= \sum_{\substack{j_0 < k \\ j_1 \geq k - j_0 \\ j_2 \geq k - j_0 \\ \cdots \\ j_l \geq k - j_0}} \binom{b}{j_0, \ldots, j_l, b - j_0 - \cdots - j_l} [P(l)]^{j_0} [Q(l)]^{j_1 + \cdots + j_l} [1 - P(l) - lQ(l)]^{b - j_0 - \cdots - j_l}. \end{aligned} \tag{17}$$

4.1. Efficient Computation of F_l

The number of arithmetic operations needed to compute S for fixed b, l, and p (using the right part of eq. 13) is $O(k)$. Furthermore, the number of operations for fixed l and p and for all k is only $O(b)$.

The number of operations needed to compute F by direct application of eq. 17 is $O(kb^l)$. However, using the recurrence equations below, F can be computed in time that is independent of k and polynomial in b and l.

Write eq. 17 as

$$F_l = \sum_{j_0 < k} \binom{b}{j_0} [P(l)]^{j_0} R_{k - j_0}(b - j_0, l, l, l) \tag{18}$$

where

$$R_k(b, l, m, n) = \sum_{\substack{j_1 \geq k \\ j_2 \geq k \\ \cdots \\ j_l \geq k}} \binom{b}{j_1, \ldots, j_l, b - j_1 - \cdots - j_l} [Q(m)]^{j_1 + \cdots + j_l} [1 - P(m) - nQ(m)]^{b - j_1 - \cdots - j_l}. \tag{19}$$

By considering the sum over j_l (represented by j in the sum below) separately, we have

$$R_k(b, l, m, n) = \sum_{j \geq k} \binom{b}{j} [Q(m)]^j R_k(b - j, l - 1, m, n) \tag{20}$$

with boundary condition

$$R_k(b, 0, m, n) = [1 - P(m) - nQ(m)]^b. \tag{21}$$

With these equations, a particular $R_k(b, l, m, n)$ can be computed from the various $R_k(c, l-1, m, n)$ $(k \le c \le b)$ in $O(b)$ operations. To compute R by repeated application of eq. 21, we need l levels with $O(b)$ R's per level. This leads to time $O(lb^2)$ to compute a set of R. The time to compute F_l is dominated by the time needed to compute the R's, leading to time $O(lb^2)$ to compute a particular F. For fixed p and b and for all k, F_l can also be computed in time $O(lb^2)$.

5. Approximations

5.1. Chernoff bounds

The sums for S_l and F_l are incomplete binomial sums. They do not have closed forms (implied by [15]), but, as we show below, Chernoff techniques [9] lead to useful approximations. For

$$L(i) = \begin{cases} 1, & i \le k, \\ 0, & i > k; \end{cases} \quad \text{and} \quad U(i) = \begin{cases} 0, & i < k, \\ 1, & i \ge k; \end{cases} \tag{22}$$

and for some fixed k in the range $0 \le k \le n$, we have

$$\sum_{0 \le i \le k} a_i = \sum_{0 \le i \le n} a_i L(i) \quad \text{and} \quad \sum_{k \le i \le n} a_i = \sum_{0 \le i \le n} a_i U(i). \tag{23}$$

In addition, when each $a_i \ge 0$, replacing $L(i)$ (or $U(i)$) with a pointwise upper bound gives an upper bound on the sum. Chernoff [9] noticed that useful bounds for partial binomial sums result when one uses

$$L(i) = x^{-k+i} \quad \text{with} \quad x \le 1 \quad \text{and} \quad U(i) = x^{-k+i} \quad \text{with} \quad x \ge 1, \tag{24}$$

and then chooses the x that gives the smallest upper bound.

The Chernoff bound for S is

$$S_l \le x^{-k} \sum_j \binom{b}{j} [xP(l)]^j [1-P(l)]^{b-j} = x^{-k}[1+(x-1)P(l)]^b \tag{25}$$

for any $x \ge 1$.

A Chernoff bound for R is

$$R_k(b, l, m, n) \le x^{-kl} \sum_{j_1, \ldots, j_l} \binom{b}{j_1, \ldots, j_l, b-j_1-\cdots-j_l} [xQ(m)]^{j_1+\cdots+j_l} [1-P(m)-nQ(m)]^{b-j_1-\cdots-j_l} \tag{26}$$

$$\le x^{-kl}[1-P(m)-(n-lx)Q(m)]^b \tag{27}$$

for any $x \ge 1$.

Using this Chernoff bound for R leads to the following Chernoff bound for F,

$$F_l \le y^{-k+1} \sum_j \binom{b}{j} [yP(l)]^j x^{-(k-j)l} [1-P(l)+l(x-1)Q(l)]^{b-j} \tag{28}$$

$$\le x^{-kl} y^{-k+1} [1+(x^l y-1)P(l)+l(x-1)Q(l)]^b, \tag{29}$$

for any $x \geq 1$ and any $y \leq 1$.

5.2. Regions and boundaries for S_l

The optimum x for eq. 25 is either on the boundary ($x = 1$) or when the derivative with respect to x is equal to zero. Letting x_* be the x value that gives a derivative of zero, we have

$$x_* = \frac{k[1 - P(l)]}{(b-k)P(l)}. \tag{30}$$

When $x_* \geq 1$, it is the optimum x for eq. 25. Otherwise ($x_* < 1$) the optimum x is 1. In addition, we check whether x_* is strictly within range ($x_* > 1$). This is the case when

$$k > bP(l). \tag{31}$$

This completes the first stage of finding the Chernoff approximation to S_l. The second stage, which is done in Section 5.4.1, is to determine just how small the Chernoff bound is as a function of the parameters (b, k, l, and p). We will show that the bound on S_l is an exponential function of the negative of the square of the distance α_1 (with $\alpha_1 = k/b - P(l)$) inside the boundary eq. 31. Thus, S_l is extremely small inside the region defined by eq. 31 except near the boundary. We show in Section 5.4.2 that S_l is close to one once we go on the other side of the boundary; the difference between S_l and one is an exponential function of the negative of the square of the distance α_2 ($\alpha_2 = (P(l) - k/b) - 1/b = -\alpha_1 - 1/b$) from the boundary. Thus, knowing whether the optimizing x is strictly within range or not gives us the most basic information about S_l (whether it is small ($x_* > 1$) or large ($x_* < 1$)). Sections 5.4.1 and 5.4.2 are needed to determine the details (just how small or large).

5.3. Regions and boundaries for F_l

To find the optimum value for x and y in eq. 29 we start by taking derivatives of the bound with respect to x and y, setting each result to zero, and solving for x and y. We want the x_* that satisfies

$$(b-k)P(l)x_*^l y + (b - kl)Q(l)x_* - k[1 - P(l) - lQ(l)] = 0. \tag{32}$$

We want the y_* that satisfies

$$(b-k+1)P(l)x^l y_* - (k-1)[1 - P(l) + lQ(l)(x-1)] = 0. \tag{33}$$

When considering whether the optimum x and y are strictly within range ($x_* > 1$, $y_* < 1$) or on the boundary, there are four cases to investigate:

Region 1. Eq. 32 with $y = 1$, $x_* > 1$;

Region 2. Eq. 33 with $x = 1$, $y_* < 1$;

Region 3. eqs. 32 and 33, $x_* > 1$, $y_* < 1$; and

Region 4. $x = 1$, $y = 1$.

In eqs. 32 and 33, x_* is associated with the effectiveness of the candidacy test (eq. 27), and y_* is associated with the probability of a set failing the frequency test (eq. 29).

The main regions of interest are Region 1, where we will show that F_l is small because the candidacy test fails with high probability, Region 2, where F_l is small because S_l is near one (F_l can never be larger than $1 - S_l$ since failure requires not only passing the candidacy test but also failing the frequency test), and Region 4 where F_l has the trivial bound of 1. In Section 5.3.3 we show that the set of parameter values that satisfy the conditions for Region 3 includes the intersection of Region 1 and Region 2. Also Region 3 has not values outside of the union of Regions 1 and 2.

When the optimum value for at least one of x and y is strictly within range (not equal to 1) then the bound for F_l is smaller. It will be shown in Section 5.5 that the bound on F_l is an exponential function of the square of the distance (basically the difference between k/b and $P(l)$ or $P(l-1)$, see Section 5.5 for details) of x_* or y_* from the boundary, so F_l rapidly becomes extremely small as x_* or y_* moves away from the boundary.

5.3.1. Region 1.

When $x_* > 1$, we will show in Section 5.5.1 that the candidacy test fails with high probability. To find when this occurs, notice that eq. 32 is satisfied by $x = 1$, $y = 1$ when

$$k = b[P(l) + Q(l)] = bP(l-1). \tag{34}$$

As b decreases, x_* increases. This implies that, for $y = 1$, $x_* > 1$ when

$$k > bP(l-1). \tag{35}$$

5.3.2. Region 2.

When $y_* < 1$, we will show in Section 5.5.2 that the frequency test succeeds with high probability. Consequently, since F_l can be no larger than $1 - S_l$, F_l is near zero in this case.

When $x = 1$, the solution to eq. 33 is

$$y_* = \frac{(k-1)[1-P(l)]}{(b-k+1)P(l)}. \tag{36}$$

This results in $y_* < 1$ when

$$k < bP(l) + 1. \tag{37}$$

For most parameters values, the regions of eq. 35 and eq. 37 do not overlap. However, subtracting the right side of eq. 37 from the right side of 35, we find that they do overlap when

$$bQ(l) < 1. \tag{38}$$

This happens both when p^{l-1} is small ($p^{l-1} \le 1/b$ is small enough) and also when $1-p$ is small ($1-p \le 1/b$ is small enough). When eq. 38 is true for all l, the Apriori Algorithm has no bad level. In this case, S_l is small for every l. Conditions where the Apriori Algorithm does have a bad level (cases where S_l is near 1) are discussed in Section 5.5.4.

5.3.3. Region 3.

To find values for the parameters such that $x_* > 1$ and $y_* < 1$ we need to satisfy eqs. 32 and 33 simultaneously. This results in the values

$$x_* = \frac{1 - P(l) - lQ(l)}{(b-k-l+1)Q(l)}, \tag{39}$$

$$y_* = (k-1)\left(\frac{(b-k-l+1)Q(l)}{1-P(l)-lQ(l)}\right)^{l-1} \frac{Q(l)}{P(l)}. \tag{40}$$

We have $x_* > 1$ when

$$k + l - 1 < b < k + l - 1 + \frac{1 - P(l) - lQ(l)}{Q(l)}. \tag{41}$$

The upper and lower limits are the same when $l = 1$, so the range is empty in that case.

All solutions to eq. 41 are in the union of Regions 1 (eq. 35) and 2 (eq. 37). The smallest k that satisfies eq. 35 is k just above $bP(l-1)$. This value for k satisfies eq. 41 when

$$b < \frac{1}{Q(l)}. \tag{42}$$

Eq. 42 is true under the same conditions that eq. 38 is true. Thus, eq. 41 is satisfied by k values outside of Region 1 only when Regions 1 and 2 overlap. Since Region 1 gives a lower limit on k and Region 2 gives an upper limit, when Regions 1 and 2 overlap, their union includes all k values.

For $k = 1$, eq. 40 implies that $y = 0$, which is less than 1. For $l = 1$ eq. 40 has no solutions. For $k \ge 2$ and $l \ge 2$, eq. 40 implies $y_* < 1$ when

$$b < k + l - 1 + \frac{1 - P(l) - lQ(l)}{Q(l)} \left(\frac{P(l)}{(k-1)Q(l)}\right)^{1/(l-1)}. \tag{43}$$

For parameter values to be in Region 3, both eqs. 41 and 43 must be satisfied.

The upper bound on b from eq. 43 is greater than the lower bound from eq. 41. The upper bound on b from eq. 43 is less than the upper bound from eq. 41 when

$$k > \frac{1}{1-p}. \tag{44}$$

For $p < 1/2$, this condition is the same as $k > 1$.

Since

$$\frac{1 - P(l) - lQ(l)}{Q(l)} \left(\frac{P(l)}{(k-1)Q(l)}\right)^{1/(l-1)} > 0, \tag{45}$$

any $k \geq b - l + 1$ always satisfies eq. 43. For $l = 2$, this rightmost term from eq. 43 reduces to

$$\frac{1}{k-1}, \tag{46}$$

which is less than 1 for $k \geq 2$. Thus, for $l = 2$, the only solution to eq. 43 is $k \geq b - l + 1$.

The left most term of the right side of eq. 43 (k) increases linearly with k, the rightmost term decreases with k. The rate of decrease slows down as k increases. As a result, the bound on b decreases at first and then increases. In some cases the bound (for fixed l) holds for small k, does not hold for moderate k, and then holds again for large k. As shown above (below eq. 45) the bound on b is always obeyed when k is large. Numerical investigations show that sometimes the bound also holds for small k, sometimes it does not; sometimes the small k region extends all the way to the large k region, sometimes it does not.

5.3.4. Region 4

In this case $x_* < 1$ and $y_* > 1$. Thus by Section 5.3.1, $k < bP(l-1)$ and by Section 5.3.2, $k > bP(l) + 1$. Thus, for such k, we have $bP(l) < k < bP(l-1)$. In such cases, the candidacy test succeeds with probability near 1, but the frequency test fails with high probability (F_l is high). Thus, The Apriori Algorithm experiences a bad level in this Region. In Section 5.5.4 we give bounds on F_l.

5.4. Bounds on S_l

Section 5.2 found the boundary between the region where S_l is small and where it is large. We now compute just how small (with an upper bound) or large (with a lower bound).

5.4.1. Upper bound on S_l

We now give an upper bound on S_l when $k > bP(l)$ to show that it is near 0. In the next section we give a lower bound when $k > bP(l)$ to show that in that case it is near 1.

By plugging the x_* value from eq. 30 into the bound from eq. 25 we obtain

$$S_l \leq \left(\frac{P(l)}{k}\right)^k \left(\frac{1-P(l)}{b-k}\right)^{b-k} b^b \tag{47}$$

so long as $x_* \geq 1$. By eq. 31 the condition $x_* > 1$ is equivalent to $k > bP(l)$, so we will define α_1 by

$$k = b[P(l) + \alpha_1]. \tag{48}$$

When k is greater than $bP(l)$, S_l goes to zero rapidly. In particular

$$S_l \leq e^{-b\alpha_1^2/\{2P(l)[1-P(l)]\}+O(b\alpha_1^3[1-P(l)]^{-2})} \tag{49}$$

when $\alpha_1 > 0$.

5.4.2. Lower bound on S_l

To obtain a lower bound on S_l when it is near 1, start with the right part of eq. 13. Shift the relation between k and α_1 by one so that α_2 is defined by

$$k = b[P(l) - \alpha_2] - 1. \tag{50}$$

We can now modify the derivation of eq. 49 (with $x_* < 1$) to obtain

$$S_l \geq 1 - e^{-b\alpha_2^2/\{2P(l)[1-P(l)]\}+O(b\alpha_2^3[1-P(l)]^{-2})} \tag{51}$$

when $\alpha_2 > 0$.

5.5. Bounds for F_l

Section 5.3 found the parameters regions relevant to F_l. In this section we will establish bounds on F_l associated with these regions.

5.5.1. Bounds on F_l in Region 1

When $k > bP(l-1)$ we are in Region 1 of Section 5.3. We now give an upper bound on F_l to show that it is near 0 in this case. Thus in this region the candidacy test fails with high probability.

By eq. 29 with $y = 1$

$$F_l \leq x_*^{-kl}[1 + (x_*^l - 1)P(l) + l(x_* - 1)Q(l)]^b. \tag{52}$$

(Note that bounds on F_l obtained with $y = 1$ are also bounds on $C_l = F_l + S_l$. The definition for C_l (eq. 16) has a sum over all values of j_0, but setting $y = 1$ also sums at unit weight over all values of j_0.) The optimum x_* is given by eq. 32. Solve eq. 32 (with $y = 1$) for x_* with $x_* = 1 + \delta$ and small δ. Let θ stand for a function that approaches 1 in the limit as δ approaches 0. (Just as various big O are associated with different implied constants, different θ's are associated with different functions that approach 1 in the limit.)

$$\delta = \frac{k - bP(l) - bQ(l)}{b[lP(l-1)] - kl[P(l-1)]}\left(1 + \frac{[k - bP(l) - bQ(l)](b-k)l(l-1)P(l)\theta/2}{\{b[lP(l) + Q(l)] - kl[P(l-1)]\}^2}\right)^{-1}. \tag{53}$$

Define α_3 by

$$k = b[P(l-1) + \alpha_3]. \tag{54}$$

In eq. 52 replace k by its value in terms of α_3 and in plug the value of x implied by eq. 53 to obtain

$$F_l \leq e^{-bl\theta\alpha_3^2/(2\{P(l-1)+(l-1)P(l)-l[P(l-1)]^2\})} \tag{55}$$

when α_3 is small enough, i.e.,

$$\alpha_3 = \{lP(l) + Q(l) - l[P(l-1)]^2\}o(1). \tag{56}$$

5.5.2. Region 2

When $k < bP(l) + 1$ we are in Region 2 of Section 5.3 and by eq. 51 nearly all item sets pass the frequency test. Since an item set must first pass the candidacy test and then fail the frequency test, F_l can be no larger than $1 - S_l$, which (by eq. 51) gives the bound

$$F_l \le e^{-b\alpha_2^2/\{2P(l)[1-P(l)]\}+O(\alpha_2^3 b[1-P(l)]^{-2})}, \tag{57}$$

where α_2 is defined by $k = b[P(l) - \alpha_2] - 1$ (eq. 50).

5.5.3. Region 3

Since Region 3 is entirely inside of Regions 1 and 2, we can use results from the previous two sections to obtain upper bounds on F_l. With additional algebra even better upper bounds could be obtained, but the previous bounds are good enough for most purposes.

5.5.4. Region 4

When $bP(l) < k < bP(l-1)$ we are in Region 4 of Section 5.3. The candidacy test succeeds with high probability but the frequency test succeeds with low probability. We now give a lower bound on F_l to show that there are cases where it is near 1.

In eq. 19, the quantity R_k is defined by sums where each $j_i \ge k$ (for $1 \le i \le l$). Using inclusion-exclusion arguments, an alternate way to compute R_k is

$$R_k(b,l,m,n) = \sum_h (-1)^h \binom{l}{h} r_k(b,l,m,n,h), \tag{58}$$

where

$$r_k(b,l,m,n,h) = \sum_{\substack{j_1<k\\ j_2<k\\ \cdots\\ j_h<k\\ j_{h+1},\ldots,j_l}} \binom{b-j_1}{j_2,\ldots,j_l,b-j_1-\cdots-j_l} [Q(m)]^{j_1+\cdots+j_l}[1-P(m)-nQ(m)]^{b-j_1-\cdots-j_l} \tag{59}$$

$$= \sum_{\substack{j_1<k\\ j_2<k\\ \cdots\\ j_h<k}} \binom{b-j_1}{j_2,\ldots,j_l,b-j_1-\cdots-j_l} [Q(m)]^{j_1+\cdots+j_h}[1-P(m)-(n-l+h)Q(m)]^{b-j_1-\cdots-j_h}. \tag{60}$$

The $h = 0$ term of eq. 58 is the sum over the full range for the j's. The $h = 1$ term subtracts (for each j) the part of the range that is not included in the definition of R. The $h = 2$ corrects for the overcorrection of the $h = 1$ term (regions where two j's where out of range were subtracted off twice). Each successive h corrects for the previous h. Therefore, if the sum over h is terminated at some value before l, the result is a lower or upper limit on R depending on whether the first omitted term is negative or positive. We use the following case of this result.

$$R_k(b,l,m,n) \geq r_k(b,l,m,n,0) - l r_k(b,l,m,n,1) \tag{61}$$

$$\geq [1-P(m)-(n-l)Q(m)]^b - l\sum_{j<k}\binom{b}{j}[Q(m)]^j[1-P(m)-(n-l+1)Q(m)]^{b-j}. \tag{62}$$

By eq. 18 we have

$$F_l \geq \sum_{j_0<k}\binom{b}{j_0}[P(l)]^{j_0}\left([1-P(l)]^{b-j_0} - l\sum_{j<k}\binom{b-j_0}{j}[Q(l)]^j[1-P(l)-Q(l)]^{b-j_0-j}\right). \tag{63}$$

A lower bound on the sum that comes from the first term in the large parentheses is given by eq. 51. The reasoning that leads to eq. 29 gives the following bound for the sum coming from the second term

$$\sum_{j_0<k}\binom{b}{j_0}[P(l)]^{j_0}\sum_{j<k}\binom{b-j_0}{j}[Q(l)]^j[1-P(l)-Q(l)]^{b-j_0-j} \leq x^{-k}y^{-k+1}[1+P(l)(xy-1)+(x-1)Q(l)]^b. \tag{64}$$

By setting $y=1$ and using eq. 12 the bound becomes $x^{-k}[1+(x-1)P(l-1)]^b$, which is the upper bound on S_{l-1} (eq. 25). Thus, combining this with eqs. 51 and 49, we obtain

$$F_l \geq 1 - e^{-b\alpha_1^2/\{2P(l)[1-P(l)]\}+O(b\alpha_1^3[1-P(l)]^{-2})} - le^{-b\alpha_4^2/\{2P(l-1)[1-P(l-1)]\}+O(b\alpha_4^3[1-P(l-1)]^{-2})} \tag{65}$$

with α_1 and α_4 related to k by $k = b[P(l)+\alpha_1]$ and $k = b[P(l-1)-\alpha_4]-1$ when both α_1 and α_4 are positive.

This bound is good enough to show that for some k the Apriori Algorithm has one bad level. Consider k equal to the integer nearest $[bP(l)+bP(l-1)-1]/2$, i.e.,

$$k = \frac{bP(l)+bP(l-1)-1}{2} + \eta \tag{66}$$

with $|\eta| \leq 1/2$. This results in

$$\alpha_1 = \frac{bP(l-1)-bP(l)-1}{2} + \eta, \qquad \alpha_4 = \frac{bP(l-1)-bP(l)-1}{2} - \eta. \tag{67}$$

This results in α_1 and α_4 both being $\Theta(k(1-p))$ when $b(1-p)P(l-1)$ is above 3. (If $b(1-p)P(l-1)$ is below 3 then there may not be room to have an integer that is both between $bP(l-1)-1$ and $bP(l)$ and also far away from both of them.) When $b(1-p)P(l-1)$ is above 3 (which implies that p is not near 1) the second exponent in eq. 65 (the one with α_4) is $-\Theta(bk^2/\{P(l-1)[1-P(l-1)]\})$ and (for small p) the first exponent is more negative yet, i.e., $-\Theta(bk^2/\{P(l)[1-P(l)]\})$. Thus, when $bP(l-1)$ is large, there is a k value where F_l is extremely close to 1. When k is near $bP(l)$ for some l the bound from eq. 65 is not good enough to show that F_l is close to 1. The sample calculations (in a following section), however, show that for such k values there are usually two l values that are each moderately bad (F_l above a constant), at least when $b(1-p)P(l-1)$ is not small. Thus, the conclusion is that the Apriori Algorithm, when it is run on random data, usually has one bad level or two half-bad levels. (When the threshold, k, is small it may have no bad levels.)

6. Total work

Eq. 65 shows that for random data there are many cases where the Apriori Algorithm has one bad level, i.e., a level where many item sets pass the candidacy test but few of them pass the frequency test. Eq. 38 shows that there are rare cases where the Apriori Algorithm has no bad levels. The Apriori Algorithm has a reputation for being effective in practice [2, 7, 21]. In this section we show that for many parameter values even when there is a bad level, the bad level comes before the algorithm has done much work and the algorithm is extremely good for the levels after the bad one. This leads to good overall performance. Under the assumption that accesses to the original data dominate the running time, large running time result from those terms in eq. 8 where the binomial coefficient is large and $S_l + F_l$ is not small. No algorithm that explicitly examines the data to verify the number of occurrences for a set can be fast if a large fraction of the possible sets for large l must be processed. The merit of the Apriori Algorithm is that $S_l + F_l$ usually becomes extremely small once l increases beyond the value that results in $k > bP(l)$. This is shown by the following rough calculation. Consider the ratio of the l and $l+1$ terms from eq. 8:

$$\binom{|I|}{l+1} \Big/ \binom{|I|}{l} = \frac{|I|-l}{l+1} \approx \frac{|I|}{l+1}, \tag{68}$$

so long as l is much less than $|I|$. Choose l so that k is near to $bP(l-1)$. Using this value of l in eq. 55 results in α_3 near 0 (from eq. 56) and F_l near 1. Now consider eq. 55 with l one larger. For this l, we have $\alpha = P(l-1) - P(l)$ and eq. 55 gives the bound

$$F_l \leq e^{-b(l+1)[P(l-1)-P(l)]^2\theta/(2\{P(l)+lP(l+1)-(l+1)[P(l)]^2\})}. \tag{69}$$

Since k is approximately bp^l, for small p this bound is approximately

$$e^{-b(l+1)p^{l-2}/2} \approx e^{-(l+1)k/(2p)}. \tag{70}$$

The ratio of the amount of work that the Apriori Algorithm does on level $l+1$ to the amount of work on level l is approximately

$$\frac{|I|}{l}e^{-(l+1)k/(2p)}. \tag{71}$$

In most interesting cases this ratio will be much less than 1. There is further improvement as l increases. For most parameter values and for random data the amount of work that the Apriori Algorithm does drops rapidly after the bad level.

7. Sample computations

This section contains sample calculations for $b = 1024$ baskets, $1 \leq l \leq 5$, with thresholds in the range $1 \leq k \leq 1024$. Table 1 gives S_l for $p = 1/2$, $1 \leq l \leq 5$. Table 2 gives S_l for $p = 1/16$. Table 3 gives F_l

for $p = 1/2$. Table 4 gives F_l for $p = 1/16$. Each table has results for only a few selected values of k. The selected values for k includes those were F_l is maximum, where it is just above 1/2, and where it is just below 1/2. Figure 1 is a graph of S_l for $p = 1/2$. Upper and lower bounds from eqs. 49 and 51 are also included in Figure 1. Figure 2 is a graph of F_l for $p = 1/2$. Figure 3 is a graph of F_l along with the bounds from eqs. 55, 57, and 65. For all bounds plotted in the figures, big O terms were ignored and θ was set to 1. The $p = 1/16$ cases does not lead to clear graphs, so none are given. For this case, one can best see what is happening by examining the tables.

When deciding which results to report, we had to balance the interest in large values for b (up to 100,000 in [2]) with the need to keep the computing time reasonable. Also, when we had to balance the interest in small values for p with the need for results to show the various characteristics of the algorithm. Also, it is difficult to compute $(1-p)^j$ accurately when p is near zero and j is large. We used code where the number of multiplications increased only as fast as $\ln j$. In addition, the values of S were computed exactly using Maple and then converted to floating point. The Maple program was too slow to compute F in this way. The values for S were computed with both exact and floating point arithmetic, but F was computed only with floating point arithmetic. For S the results from the two ways were not significantly different, but the floating point calculations sometimes gave zero for values below 10^{-70}. Also, it was difficult to tell just how close to 1 a floating point value was once it went above $1 - 10^{-12}$.

From Table 1 and also from Figure 1, we see that, for fixed, moderate-sized values of k, S_l is extremely close to 1 for small values of l and that S_l is extremely small for large values of l. The transition from near 1 to small is quite sharp with increasing l. The transition value of l increases as k decreases. For large k, even S_1 is small. For small k one must go to large l values (not shown) before S_l becomes small. In Figure 1, the three rightmost curves refer to $l = 1$. The very rightmost is the upper bound from eq. 49. The next rightmost is the actual value from eq. 13. The least rightmost in the group of three is the lower bound from eq. 51. Proceeding to the left, we have corresponding groups for $l = 2$, 3, 4, and 5. For $l = 3$, 4, and 5, one can notice that the plotted "upper bound" goes below the actual value. This is because the big O term was omitted, and it is significant in these cases. None-the-less even without the big O term the upper bound gives the general idea for how the actual function behaves. Table 2 shows that the $p = 1/16$ is similar to the $p = 1/2$ case. Notice that S with $p = 1/2$ and $l = 4$ has approximately the same value as S does for $p = 1/16$ and $l = 1$, particularly when k is small.

Table 3 and also Figure 2 show the values of F_l for $p = 1/2$ from eqs. 18, 19, and 21. The rightmost (at the top) curve is for $l = 1$. The rightmost curve with a hump is for $l = 2$. In Figure 2, the leftmost curve is for $l = 5$. For any fixed k, there is one or sometimes two values of l for which F_l is not small. For most large values of k, there is just one l value where F_l is large, and for that one l value the resulting F_l is extremely close to 1, but for some large k values, there are two l values for which F_l is moderately large. As k becomes smaller, the l value that results in F_l being near one decreases. Also, F_l no longer becomes quite so close to one. Figure 3 shows the same F_l values as Figure 2, and it also shows the upper and lower bounds computed

k	S_1	S_2	S_3	S_4	S_5
1	$1.0 - 5.6 \times 10^{-309}$	$1.0 - 1.2 \times 10^{-128}$	$1.0 - 4.1 \times 10^{-60}$	$1.0 - 2.0 \times 10^{-29}$	$1.0 - 7.6 \times 10^{-15}$
2	$1.0 - 5.7 \times 10^{-306}$	$1.0 - 4.0 \times 10^{-126}$	$1.0 - 6.1 \times 10^{-58}$	$1.0 - 1.4 \times 10^{-27}$	$1.0 - 2.6 \times 10^{-13}$
3	$1.0 - 2.9 \times 10^{-303}$	$1.0 - 6.8 \times 10^{-124}$	$1.0 - 4.5 \times 10^{-56}$	$1.0 - 4.8 \times 10^{-26}$	$1.0 - 4.4 \times 10^{-12}$
4	$1.0 - 1.0 \times 10^{-300}$	$1.0 - 7.7 \times 10^{-122}$	$1.0 - 2.2 \times 10^{-54}$	$1.0 - 1.1 \times 10^{-24}$	$1.0 - 5.0 \times 10^{-11}$
5	$1.0 - 2.5 \times 10^{-298}$	$1.0 - 6.6 \times 10^{-120}$	$1.0 - 8.1 \times 10^{-53}$	$1.0 - 1.9 \times 10^{-23}$	$1.0 - 4.3 \times 10^{-10}$
22	$1.0 - 1.5 \times 10^{-265}$	$1.0 - 3.1 \times 10^{-95}$	$1.0 - 2.3 \times 10^{-34}$	$1.0 - 1.5 \times 10^{-10}$	$1.0 - 2.4 \times 10^{-2}$
32	$1.0 - 9.2 \times 10^{-250}$	$1.0 - 3.3 \times 10^{-84}$	$1.0 - 5.4 \times 10^{-27}$	$1.0 - 2.0 \times 10^{-6}$	5.2×10^{-1}
33	$1.0 - 2.9 \times 10^{-248}$	$1.0 - 3.4 \times 10^{-83}$	$1.0 - 2.4 \times 10^{-25}$	$1.0 - 4.3 \times 10^{-6}$	4.5×10^{-1}
45	$1.0 - 2.4 \times 10^{-231}$	$1.0 - 5.7 \times 10^{-72}$	$1.0 - 1.6 \times 10^{-19}$	$1.0 - 4.2 \times 10^{-3}$	1.6×10^{-2}
57	$1.0 - 6.8 \times 10^{-216}$	$1.0 - 3.1 \times 10^{-62}$	$1.0 - 3.7 \times 10^{-14}$	8.3×10^{-1}	3.1×10^{-5}
58	$1.0 - 1.2 \times 10^{-214}$	$1.0 - 1.7 \times 10^{-61}$	$1.0 - 9.2 \times 10^{-14}$	8.0×10^{-1}	1.6×10^{-5}
64	$1.0 - 1.9 \times 10^{-207}$	$1.0 - 4.0 \times 10^{-57}$	$1.0 - 1.4 \times 10^{-11}$	5.2×10^{-1}	2.5×10^{-7}
65	$1.0 - 2.9 \times 10^{-206}$	$1.0 - 2.0 \times 10^{-56}$	$1.0 - 3.0 \times 10^{-11}$	4.7×10^{-1}	1.2×10^{-7}
91	$1.0 - 6.2 \times 10^{-178}$	$1.0 - 1.9 \times 10^{-40}$	$1.0 - 1.1 \times 10^{-4}$	5.8×10^{-4}	2.3×10^{-18}
120	$1.0 - 1.5 \times 10^{-150}$	$1.0 - 7.6 \times 10^{-27}$	7.9×10^{-1}	5.3×10^{-11}	2.0×10^{-34}
121	$1.0 - 1.2 \times 10^{-149}$	$1.0 - 1.9 \times 10^{-26}$	7.6×10^{-1}	2.6×10^{-11}	4.7×10^{-35}
128	$1.0 - 1.3 \times 10^{-143}$	$1.0 - 9.8 \times 10^{-24}$	5.1×10^{-1}	1.4×10^{-13}	1.7×10^{-39}
129	$1.0 - 8.8 \times 10^{-143}$	$1.0 - 2.3 \times 10^{-23}$	4.8×10^{-1}	6.6×10^{-14}	3.8×10^{-40}
186	$1.0 - 2.8 \times 10^{-100}$	$1.0 - 7.1 \times 10^{-8}$	1.3×10^{-7}	8.9×10^{-39}	6.4×10^{-83}
247	$1.0 - 3.7 \times 10^{-65}$	7.5×10^{-1}	2.0×10^{-24}	1.2×10^{-75}	5.2×10^{-139}
248	$1.0 - 1.2 \times 10^{-64}$	7.3×10^{-1}	8.7×10^{-25}	2.4×10^{-76}	5.3×10^{-140}
256	$1.0 - 9.6 \times 10^{-61}$	5.1×10^{-1}	1.1×10^{-27}	7.2×10^{-82}	4.7×10^{-148}
257	$1.0 - 2.9 \times 10^{-60}$	4.8×10^{-1}	4.8×10^{-28}	1.4×10^{-82}	4.6×10^{-149}
377	$1.0 - 8.1 \times 10^{-18}$	3.8×10^{-17}	1.4×10^{-87}	9.8×10^{-182}	4.9×10^{-286}
503	7.2×10^{-1}	6.9×10^{-62}	1.5×10^{-178}	2.2×10^{-314}	2.1×10^{-458}
504	7.0×10^{-1}	2.4×10^{-62}	2.3×10^{-179}	1.5×10^{-315}	7.0×10^{-460}
512	5.1×10^{-1}	4.0×10^{-66}	4.4×10^{-186}	6.6×10^{-325}	9.3×10^{-472}
513	4.9×10^{-1}	1.3×10^{-66}	6.3×10^{-187}	4.4×10^{-326}	3.0×10^{-473}
533	1.0×10^{-1}	1.6×10^{-76}	3.3×10^{-204}	5.6×10^{-350}	1.9×10^{-503}

Table 1. S_l for $b = 1024$, $p = 1/2$, and selected values of k.

from eqs. 55, 57, and 65. Table 4 shows F_l for $p = 1/16$. Notice that, for small k, F with $p = 1/2$ and $l = 4$ has approximately the same value as F does for $p = 1/16$ and $l = 1$. Table 5 shows the extend of the various regions when $b = 1024$, $p = 1/2$, and $1 \leq l \leq 5$. Table 6 shows information for the $p = 1/16$ case.

k	S_1	S_2	S_3	S_4	S_5
1	$1.0 - 2.0 \times 10^{-29}$	$1.0 - 1.8 \times 10^{-2}$	2.2×10^{-1}	1.6×10^{-2}	9.8×10^{-4}
2	$1.0 - 1.4 \times 10^{-27}$	$1.0 - 9.1 \times 10^{-2}$	2.6×10^{-2}	1.2×10^{-4}	4.8×10^{-7}
3	$1.0 - 4.8 \times 10^{-26}$	7.6×10^{-1}	2.2×10^{-3}	6.3×10^{-7}	1.5×10^{-10}
4	$1.0 - 1.1 \times 10^{-24}$	5.7×10^{-1}	1.3×10^{-4}	2.4×10^{-9}	3.8×10^{-14}
5	$1.0 - 1.9 \times 10^{-23}$	3.7×10^{-1}	6.6×10^{-6}	7.6×10^{-12}	7.3×10^{-18}
22	$1.0 - 1.5 \times 10^{-10}$	3.0×10^{-10}	3.2×10^{-35}	1.3×10^{-61}	4.2×10^{-88}
32	$1.0 - 2.0 \times 10^{-6}$	1.0×10^{-18}	1.0×10^{-55}	3.7×10^{-94}	1.1×10^{-132}
33	$1.0 - 4.3 \times 10^{-6}$	1.2×10^{-19}	7.3×10^{-58}	1.7×10^{-97}	3.1×10^{-137}
45	$1.0 - 4.2 \times 10^{-3}$	9.2×10^{-32}	2.0×10^{-84}	1.6×10^{-142}	1.1×10^{-192}
57	8.3×10^{-1}	2.5×10^{-45}	1.9×10^{-112}	5.5×10^{-181}	1.3×10^{-249}
58	8.0×10^{-1}	1.7×10^{-46}	7.8×10^{-115}	1.4×10^{-184}	2.1×10^{-254}
64	5.2×10^{-1}	8.9×10^{-54}	2.5×10^{-129}	2.6×10^{-206}	2.3×10^{-283}
65	4.7×10^{-1}	5.1×10^{-55}	8.9×10^{-132}	5.9×10^{-210}	3.3×10^{-288}
91	5.8×10^{-4}	2.0×10^{-89}	1.6×10^{-197}	5.1×10^{-307}	1.4×10^{-416}
120	5.3×10^{-11}	5.6×10^{-132}	4.8×10^{-275}	1.9×10^{-419}	6.1×10^{-564}
121	2.6×10^{-11}	1.6×10^{-133}	8.7×10^{-278}	2.1×10^{-423}	4.3×10^{-569}
128	1.4×10^{-13}	2.3×10^{-144}	4.5×10^{-297}	4.1×10^{-451}	3.1×10^{-605}
129	6.6×10^{-14}	6.3×10^{-146}	7.7×10^{-300}	4.4×10^{-455}	2.1×10^{-610}
186	8.9×10^{-39}	7.9×10^{-241}	1.8×10^{-463}	2.4×10^{-687}	2.6×10^{-911}
247	1.2×10^{-75}	1.0×10^{-352}	6.7×10^{-649}	3.0×10^{-945}	1.2×10^{-1243}
248	2.4×10^{-76}	1.3×10^{-354}	5.1×10^{-652}	1.5×10^{-950}	3.5×10^{-1249}
256	7.1×10^{-82}	5.4×10^{-370}	4.9×10^{-677}	3.3×10^{-985}	1.8×10^{-1293}
257	1.4×10^{-82}	6.3×10^{-372}	3.6×10^{-680}	1.5×10^{-989}	5.2×10^{-1299}

Table 2. S_l for $b = 1024$, $p = 1/16$, and selected values of k.

k	F_1	F_2	F_3	F_4	F_5
1	5.6×10^{-309}	1.1×10^{-128}	4.1×10^{-60}	2.0×10^{-29}	7.6×10^{-15}
2	5.7×10^{-306}	4.0×10^{-126}	6.1×10^{-58}	1.4×10^{-27}	2.6×10^{-13}
3	2.9×10^{-303}	6.8×10^{-124}	4.5×10^{-56}	4.8×10^{-24}	4.4×10^{-12}
4	1.0×10^{-300}	7.7×10^{-122}	2.2×10^{-54}	1.1×10^{-24}	5.0×10^{-11}
5	2.5×10^{-298}	6.6×10^{-120}	8.0×10^{-53}	1.9×10^{-23}	4.2×10^{-10}
22	1.5×10^{-265}	3.1×10^{-95}	2.3×10^{-34}	1.5×10^{-10}	2.4×10^{-2}
32	9.2×10^{-250}	3.3×10^{-83}	5.4×10^{-27}	2.0×10^{-6}	4.8×10^{-1}
33	2.9×10^{-248}	3.4×10^{-82}	2.4×10^{-26}	4.3×10^{-6}	5.5×10^{-1}
45	2.4×10^{-231}	5.7×10^{-72}	1.6×10^{-19}	4.2×10^{-3}	$1.0-3.4\times10^{-2}$
57	6.8×10^{-216}	3.1×10^{-62}	3.7×10^{-14}	1.7×10^{-1}	5.6×10^{-1}
58	1.2×10^{-214}	1.7×10^{-61}	9.2×10^{-14}	2.0×10^{-1}	5.0×10^{-1}
64	1.9×10^{-207}	4.0×10^{-57}	1.4×10^{-11}	4.8×10^{-1}	1.8×10^{-1}
65	2.9×10^{-206}	2.0×10^{-56}	3.0×10^{-11}	5.3×10^{-1}	1.4×10^{-1}
91	6.2×10^{-178}	1.9×10^{-40}	1.1×10^{-4}	$1.0-1.0\times10^{-3}$	5.6×10^{-7}
120	1.5×10^{-150}	7.6×10^{-27}	2.1×10^{-1}	5.2×10^{-1}	2.0×10^{-18}
121	1.2×10^{-149}	1.9×10^{-26}	2.4×10^{-1}	4.7×10^{-1}	6.8×10^{-19}
128	1.3×10^{-143}	9.8×10^{-24}	4.9×10^{-1}	1.9×10^{-1}	2.0×10^{-22}
129	8.8×10^{-143}	2.3×10^{-23}	5.2×10^{-1}	1.6×10^{-1}	6.1×10^{-23}
186	2.8×10^{-100}	7.1×10^{-8}	$1.0-3.4\times10^{-7}$	3.8×10^{-13}	7.3×10^{-60}
247	3.7×10^{-65}	2.5×10^{-1}	5.0×10^{-1}	4.7×10^{-40}	4.7×10^{-112}
248	1.2×10^{-64}	2.7×10^{-1}	4.7×10^{-1}	1.3×10^{-40}	4.7×10^{-113}
256	1.0×10^{-60}	4.9×10^{-1}	2.2×10^{-1}	4.5×10^{-45}	1.1×10^{-122}
257	2.9×10^{-60}	5.1×10^{-1}	1.9×10^{-1}	1.2×10^{-45}	3.9×10^{-124}
377	8.0×10^{-18}	1.0	8.7×10^{-30}	3.8×10^{-165}	
503	2.8×10^{-1}	5.2×10^{-1}	2.7×10^{-109}		
504	3.0×10^{-1}	4.9×10^{-1}	1.5×10^{-110}		
512	4.9×10^{-1}	2.6×10^{-1}	2.2×10^{-121}		
513	5.1×10^{-1}	2.4×10^{-1}	7.4×10^{-123}		
533	9.0×10^{-1}	1.0×10^{-2}	1.8×10^{-158}		

Table 3. F_l for $b = 1024$, $p = 1/2$, and selected values of k.

k	F_1	F_2	F_3	F_4	F_5
1	$2.0{\times}10^{-29}$	$1.8{\times}10^{-2}$	$7.3{\times}10^{-1}$	$1.9{\times}10^{-3}$	$6.4{\times}10^{-10}$
2	$1.4{\times}10^{-27}$	$9.1{\times}10^{-2}$	$7.3{\times}10^{-1}$	$2.9{\times}10^{-5}$	$6.3{\times}10^{-13}$
3	$4.8{\times}10^{-26}$	$2.4{\times}10^{-1}$	$4.5{\times}10^{-1}$	$2.3{\times}10^{-7}$	$3.0{\times}10^{-16}$
4	$1.1{\times}10^{-24}$	$4.3{\times}10^{-1}$	$2.0{\times}10^{-1}$	$1.2{\times}10^{-9}$	$9.8{\times}10^{-20}$
5	$1.9{\times}10^{-23}$	$6.3{\times}10^{-1}$	$6.3{\times}10^{-2}$	$4.7{\times}10^{-12}$	$2.4{\times}10^{-23}$
22	$1.5{\times}10^{-10}$	$1.0-6.0\times10^{-10}$	$7.1{\times}10^{-23}$	$3.9{\times}10^{-61}$	$5.6{\times}10^{-93}$
32	$2.0{\times}10^{-6}$	$1.0-4.1\times10^{-6}$	$1.6{\times}10^{-40}$	$1.7{\times}10^{-93}$	$2.0{\times}10^{-137}$
33	$4.3{\times}10^{-6}$	$1.0-8.7\times10^{-6}$	$2.1{\times}10^{-42}$	$8.1{\times}10^{-97}$	$6.0{\times}10^{-142}$
45	$4.2{\times}10^{-3}$	$1.0-8.4\times10^{-3}$	$2.1{\times}10^{-68}$	$1.1{\times}10^{-137}$	$2.7{\times}10^{-197}$
57	$1.7{\times}10^{-1}$	$6.9{\times}10^{-1}$	$2.4{\times}10^{-92}$	$5.1{\times}10^{-180}$	$1.2{\times}10^{-274}$
58	$2.0{\times}10^{-1}$	$6.4{\times}10^{-1}$	$1.4{\times}10^{-94}$	$1.3{\times}10^{-183}$	$1.7{\times}10^{-287}$
64	$4.8{\times}10^{-1}$	$2.7{\times}10^{-1}$	$3.4{\times}10^{-108}$	$2.8{\times}10^{-205}$	
65	$5.3{\times}10^{-1}$	$2.2{\times}10^{-1}$	$1.7{\times}10^{-110}$	$6.4{\times}10^{-209}$	
91	$1.0-5.8\times10^{-4}$	$3.3{\times}10^{-7}$	$3.9{\times}10^{-173}$		
120	$1.0-5.3\times10^{-11}$	$2.8{\times}10^{-21}$	$4.0{\times}10^{-250}$		
121	$1.0-2.6\times10^{-11}$	$6.7{\times}10^{-22}$	$3.5{\times}10^{-253}$		
128	$1.0-1.4\times10^{-13}$	$2.1{\times}10^{-26}$	$9.7{\times}10^{-276}$		
129	$1.0-6.3\times10^{-14}$	$4.4{\times}10^{-27}$	$4.1{\times}10^{-279}$		
186	1.0	$7.0{\times}10^{-77}$			
247	1.0	$1.3{\times}10^{-150}$			
248	1.0	$5.8{\times}10^{-152}$			
256	1.0	$5.1{\times}10^{-163}$			
257	1.0	$2.0{\times}10^{-164}$			

Table 4. F_l for $b = 1024$, $p = 1/16$, and selected values of k.

l	1	2	3	4	5
Region 1		$k \geq 513$	$k \geq 257$	$k \geq 129$	$k \geq 65$
Region 2	$k \leq 512$	$k \leq 256$	$k \leq 128$	$k \leq 64$	$k \leq 32$
Region 3		$1023 \leq k \leq 1023$	$1022 \leq k \leq 1022$	$1020 \leq k \leq 1021$	$1016 \leq k \leq 1020$
Region 4	$513 \leq k \leq 1024$	$257 \leq k \leq 512$	$129 \leq k \leq 256$	$65 \leq k \leq 128$	$33 \leq k \leq 64$

Table 5. Region boundaries for $b = 1024$, $p = 1/2$.

l	1	2	3	4	5
Region 1		$k \geq 65$	$k \geq 5$	$k \geq 1$	$k \geq 1$
Region 2	$k \leq 64$	$k \leq 4$	$k \leq 0$	$k \leq 0$	$k \leq 0$
Region 3		$1023 \leq k \leq 1023$	$1020 \leq k \leq 1022$	$833 \leq k \leq 1021$	$2 \leq k \leq 1020$
Region 4	$65 \leq k \leq 1024$	$5 \leq k \leq 64$	$1 \leq k \leq 4$		

Table 6. Region boundaries for $b = 1024$, $p = 1/16$.

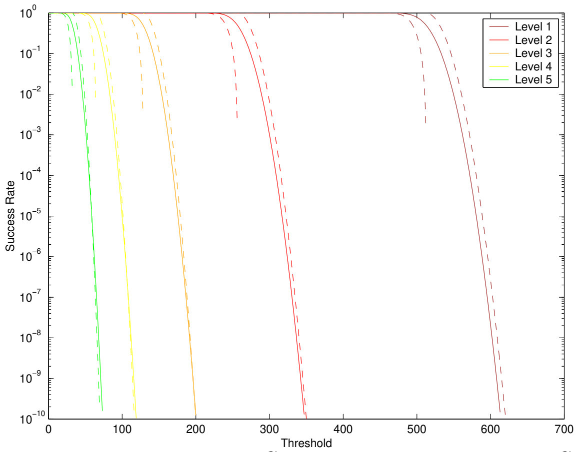

Figure 1. The value of S_l and approximations to S_l for $p = 1/2$ and $1 \leq l \leq 5$. The rightmost curve is the upper bound on S_1 (eq. 49 with the big O term omitted). The next rightmost curve is the actual value of S_1. The next rightmost curve is the lower bound on S_1 (eq. 51 with the big O term omitted). The bounds are plotted only for the range where they are valid. Proceeding to the left, each group of three curves show similar information on S_l for $l = 2$, 3, 4 and 5.

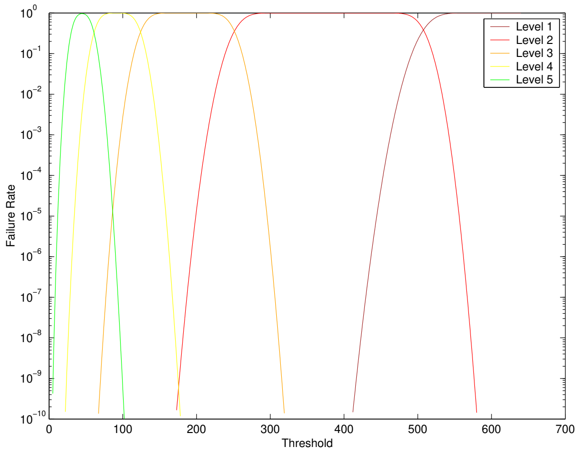

Figure 2. The value of F_l for $p = 1/2$ and $1 \leq l \leq 5$. The left most hump is the curve for $l = 5$, the next leftmost hump is for $l = 4$, etc.

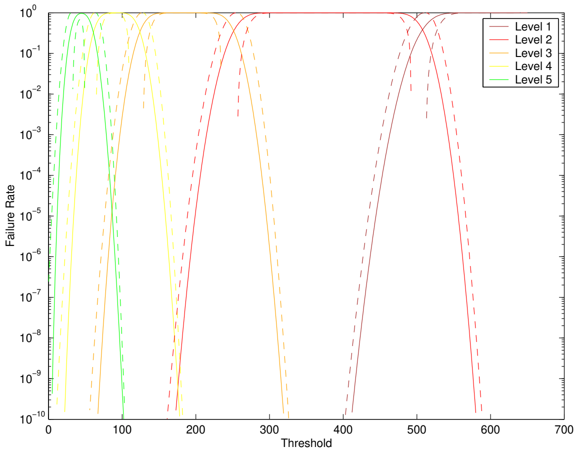

Figure 3. The value of F_l for $p = 1/2$ and $1 \leq l \leq 5$ along with upper and lower bound.

8. Experimental results.

In this section we report the results of running the Apriori Algorithm on data that is more like that used in practice. We used three data sets: (1) synthetic data produced by the generator from the IBM Quest Research Group [14], (2) real data based on the U. S. Census [24], and (3) real web data [25]. These data sets have two major differences from the sets analyzed in the previous sections: (1) the probability of an item being in a basket is not the same for every item, but varies greatly from item to item (this effect is particularly strong in the census and web data sets) and (2) the items are correlated [6].

Given the major differences between the data used in the analytical study and the data used for the experimental study, it is not surprising that many of the results from the experiments differ from those of the analysis. However, the experiments did verify one important conclusions of the analysis: work done by the Apriori Algorithm is dominated by the failure rate and the total failure rate is almost as high as it possibly can be. To be more precise, for most thresholds, there is a single level where most of the work is done, and on this level almost every candidate fails to be a success.

In the random model, there is a fixed number of items, $|I|$, each occurring in a basket with probability p. For level l, when the threshold, k, is such that $k < bp^l$, almost every combination of items ($\binom{|I|}{l}$ combinations) is frequent. Once l is large enough that $k > bp^l$, then almost no combination is frequent. When p is small, the value of bp^l changes rapidly with l.

One can obtain an informal understanding of the experimental data presented below by permitting $|I|$, b and p to vary (both with the threshold, k, and the level, l). In the random model, level l will have approximately $\binom{|I|}{l}$ successful item sets so long as k is much less than bp^l and it will have a lot less if k is much larger than bp^l. Also, when p is small, the value of bp^l changes rapidly with l. The ratio of the number of item sets on level l to level $l-1$ is $\binom{|I|}{l}/\binom{|I|}{l-1}$ which is approximately $|I|/l$ if $|I|$ is much larger than l. Thus, we would expect the number of item sets to increase rapidly with l until the level where bp^l became larger than k. Even when the effective $|I|$ decreases with l and the effective p increases, one is likely to get behavior that is qualitatively the same. The number of item sets will increase rapidly at first, and then it will suddenly decrease. The cause of the decrease will be the high failure rate on the level just before the decrease. This will be the level where most of the work is done, because it is the level with the most candidates.

Now, let's look at some actual data sets. These results were computed using a version of the Apriori Algorithm that computes frequent item sets for all values of the threshold with a lower limit being impossed on the threshold as necessary to avoid running out of computer memory. Figures 4–9 shows the results for three data sets. On each figure, the bottom axis shows the threshold. Each color curve refers to one level of the Apriori Algorithm — 1: brown, 2: red, 3: orange, 4: yellow, 5: green. The even numbered figures show the number of candidates as a solid curve, the upper bound from [11] as a dotted curve, and the number of failures as a dashed curve. For a given level, the upper bound is upper most, the number of candidates is in the middle, and the number of failures is lower most. Often the dotted and/or dashed curves are so close

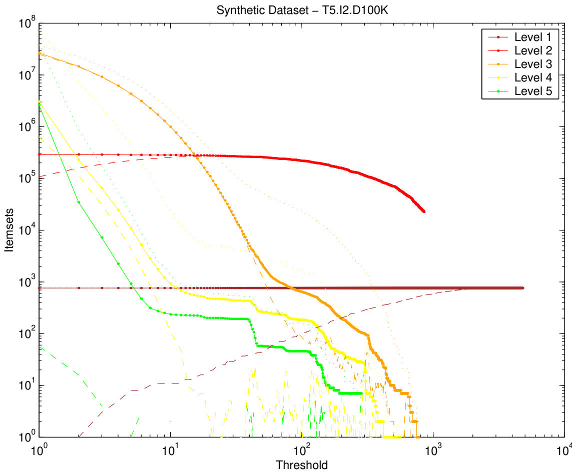

Figure 4. Behavior of the algorithm on synthetic data. The dotted line shows the upper bound, the solid line shows the actual number of candidates considered, and the dashed line shows the number of candidates that fail for each level.

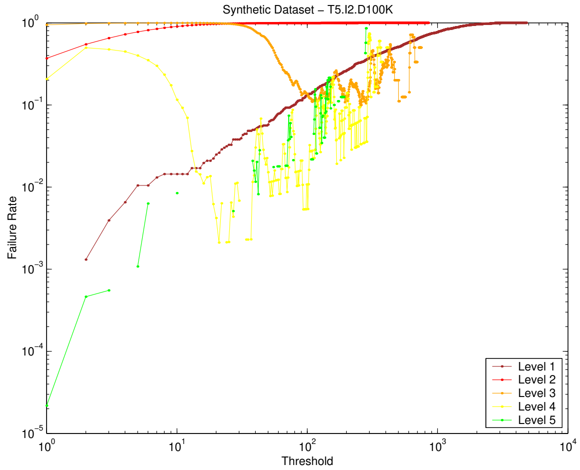

Figure 5. The failure rate for the synthetic dataset. value

to the solid curve that they don't show up. Each even numbered figure is followed by an odd number figure that shows the ratio of failures to candidates (for each level).

8.1 Synthetic Data

The first data set was generated using the synthetic dataset generator from the IBM Quest Research Group [14] to create datasets in the same fashion as [3]. We tested several synthetic datasets, each of which had 1000 items. We report the results for the T5.I2.D100K (average transaction size 5, size of the average maximal frequest item set 2, number of transactions 100,000), as they are representative of the experimental results for the other synthetic datasets.

The solid curves in Figure 4 show the number of candidates on each level (which is proportional to the work done on the level). Notice that for high thresholds (above about 1000) the most work is done on level 1, for intermediate thresholds (between about 20 and 999) the most work is done on level 2, and for low thresholds (between 1 and about 19) the most work is done on level 3. Notice that for the highest solid curve (which one is highest depends on the threshold) the number of failures is almost equal to the number of candidates. In Figure 4 this shows up by the dashed curve falling on top of the solid curve (or almost on top). Figure 5 shows the ratio of failures to candidates. This ratio is close to 1 for the curve that is upper most in Figure 4.

For each threshold, the upper most curve represents the bad level, the level where most of the work is done and where almost every candidate is a failure. For levels passed the bad level one may want to use a version of the Apriori Algorithm that counts item sets for all remaining levels in one pass. (This was one of the main motivations for [11].)

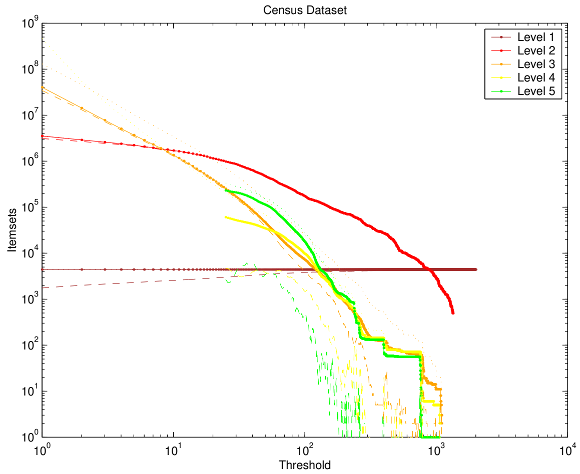

Figure 6. Behavior of the algorithm on census data. The dotted line shows the upper bound, the solid line shows the actual number of candidates considered, and the dashed line shows the number of candidates that fail for each level.

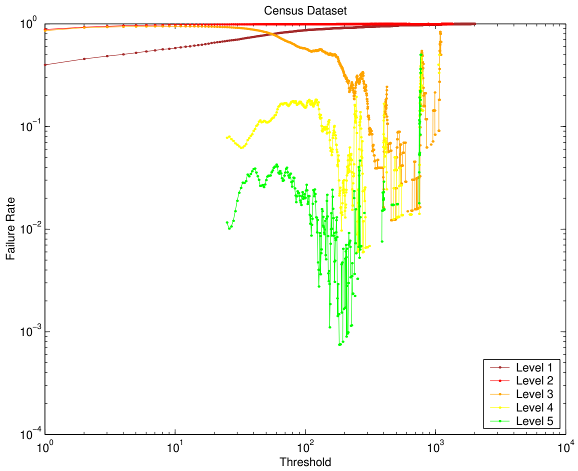

Figure 7. The failure rate for the census dataset. value

8.2 Census Data

The second data set was U. S. Census data, using Public Use Microdata Samples (PUMS) [24]. We chose the same sample that was used by [6, 7]. The data represents a five percent sample of the 1990 U.S. Census data for Washington, D. C. It contains 30,370 entries with 122 attributes.

Following [6, 7, 12], we modified the data in the following ways. For monetary values, we took the ceiling of the logarithm of the value. Then, we assigned a unique integer to each possible value in the data. In total, the converted PUMS data was 16.6 MB, with 7523 unique items. In addition, we pruned the highly frequent items, which yield an intractable number of frequent itemsets.

Due to the high number of correlated items in this type of data our computing resources did not permit us to compute levels 4 and 5 for thresholds less than 25.

The solid curves in Figure 6 show the number of candidates on each level. Notice that for high thresholds (above about 1000) the most work is done on level 1 and for intermediate thresholds (between 25 and 999) the most work is done on level 2. We are missing data for low thresholds. Figures 6 and 7 show that for the level where most of the work is done, the ratio of failures to candidates is almost 1.

8.3 Web Data

The third dataset was the BMS-WebView-1 dataset described in [25]. This data contains clickstream data from a web-based merchandising company. There are 59,602 transaction and 497 distinct items in the web data.

The solid curves in Figure 8 show the number of candidate tests Figure 8 shows the results for the web data. Resources did not permit thresholds below 35 for levels 4 and above.

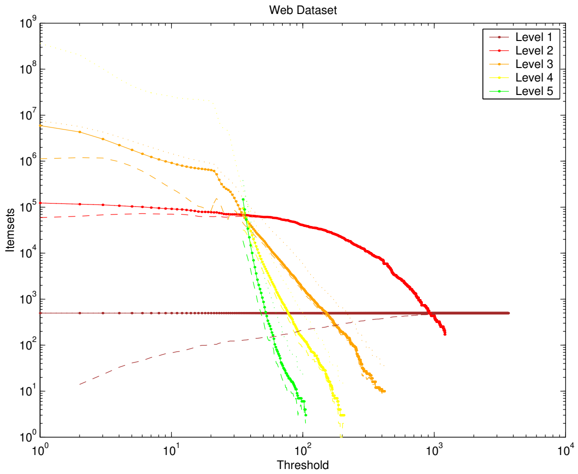

Figure 8. Behavior of the algorithm on web data. The dotted line shows the upper bound, the solid line shows the actual number of candidates considered, and the dashed line shows the number of candidates that fail for each level.

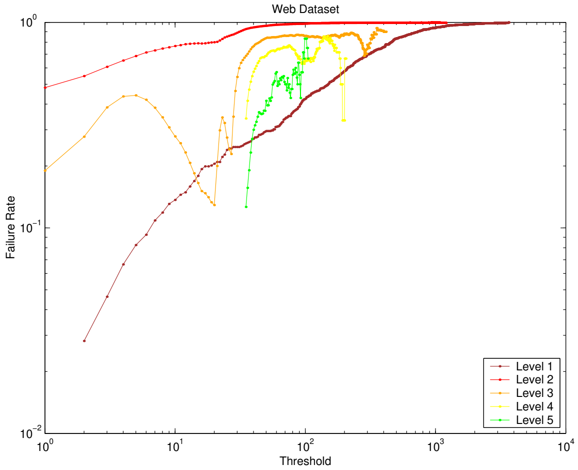

Figure 9. The failure rate for the web dataset. Note that each of the first three levels is "bad" for certain ranges of the threshold value

Level 3, however, has a large enough number of successes to suggest that levels 4 and 5 are the truly "bad" levels for this data. The slope of these two levels is extremely steep, with level 5 being nearly vertical. Figure 9 shows the failure rate for the web data.

9. Discussion

Most people using the Apriori Algorithm are probably interested in applying it to data generated by nonrandom shoppers. The formulas for random data predict that the algorithm uses a lot of time until it is processing sets with so many items that a random set is unlikely to have more occurrences than the threshold. Once this point is reached the algorithm does very little additional work. The synthetic dataset was like this except that the various items had various probabilities. This resulted in the transition to small work to be smeared out. This lead to a moderate improvement in the efficiency of the algorithm, but otherwise things were qualitatively similar. The census and web datasets had more important deviations from the properties of a random dataset, but the result was that the Apriori Algorithm did much better than the random theory predicted.

References

1. R. Agrawal, C. Aggarwal and V. Prasad, *Depth First Generation of Long Patterns*, in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (2000) pp 108–118.
2. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen and A. Verkamo, *Fast Discovery of Association Rules*, in Advances in Knowledge Discovery and Data Mining, U. M. Fayyad, G. Piatesky-Shapiro, P. Smyth and R. Uthurusamy (eds.), MIT Press, (1996) pp 307–328.
3. R. Agrawal, R. Srikant, *Fast algorithms for mining association rules*, in Proceedings of the 1994 Very Large Data Bases Conference, (1994) pp 487–499.
4. R. Bayardo, *Efficiently Mining Long Patterns From Databases*, in Proceedings of the ACM SIGMOD International Conference of Management of Data, (1998) pp 85–93.

6. S. Brin, R. Motwani and C. Silverstein, *Beyond Market Baskets: Generalizing Association Rules to Correlations*,in Proceedings of ACM the SIGMOD International Conference of Management of Data, (1997) pp 265–276.
7. S. Brin, R. Motwani, J. D. Ullman and S. Tsur, Dynamic Itemset Counting and Implication Rules for Market Basket Data, in Proceedings of the ACM SIGMOD Conference on Management of Data, (1997) pp 255–264.
9. Herman Chernoff, *A Measure of Asymptotic Efficiency for Test of a Hypothesis Based on the Sum of Observations*, Annals of Mathematical Statistics, **23** (1942) pp 493–507.
10. Michael R. Garey and David S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, San Francisco, 1979.
11. Floris Geerts, Bart Goethals, and Jan Van den Buusche, *A Tight Upper Bound on the Number of Candidate Patterns*.
12. Dennis P. Groth and Edward L. Robertson, *Discovering Frequent Itemsets in the Presence of Highly Frequent Items*, Rule Based Data Mining Workshop, in Conjunction with the 14th International Conference On Applications of Prolog, pp 237–245, 2001.
13. J. Han, J. Pei and Y. Yin, *Mining Frequent Patterns Without Candidate Generation*, in Proceedings of ACM SIGMOD International Conference of Management of Data, (2000) pp 1–12.
14. http://www.almaden.ibm.com/cs/quest//syndata.html
16. Laks V.S. Lakshmanan, Raymand Ng, Jiawei Han, and Alex Pang, *Optimization of Constraint Frequent Set Queries with 2-variable Constraints*, in Proceedings of ACM SIGMOD International Conference of Management of Data, pp 157–168, 1999.
15. Michael Karr, *Summations in Finite Terms*, J. ACM **28** (1981) pp 305–350.
17. H. Mannila, *Methods and problems in data mining*, in Proceedings of the International Conference on Database Theory, pp 41–55, 1997.
18. H. Mannila, H. Toivonen, A. I. Verkamo, *Efficient Algorithms for discovering Association Rules*, Knowledge Discovery in Databases (KDD'94) (1994), AAAI Press, Seattle, pp 181–192.
19. R. T. Ng, L. V. S. Lakshmanan, J. Han and A. Pang, *Exploratory Mining and Pruning Optimizations of Constrained Association Rules*, in Proceedings of ACM SIGMOD International Conference of Management of Data, pp 13–24, 1998.
20. Paul Walton Purdom, Jr. and Cynthia A. Brown, The Analysis of Algorithms, Holt, Rinehart and Winston, New York (1985).
21. S. Sarawagi, S. Thomas and R. Agrawal, *Integrating Association Rule Mining with Relational Database Systems: Alternatives and Implications*, in Proceedings of ACM SIGMOD International Conference of Management of Data, pp 343–354, 1998.
22. Hannu Toivonnen, *Discovery of Frequent Patterns in Large Data Collections*, PhD Thesis, Report A-1996-5, University of Helsinki, Department of Computer Science, November 1996.
23. D. Tsur, J. D. Ullman, S. Abiteboul, C. Clifton, R. Motwani, S. Nestorov and A. Rosenthal, *Query Flocks: A Generalization of Association-Rule Mining*, in Proceedings of ACM SIGMOD International Conference of Management of Data, pp 1–12, 1998.
24. http://www.census.gov/DES/www/welcome.html
25. Zijian Zheng, Ron Kohavi and Llew Mason, *Real World Performance of Association Rule Algorithms*, in Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 401–406, 2001.

Appendix

This section gives proofs and the derivations of equations.

A subpart of the Apriori Algorithm is NP-complete. Determining whether some set of size l occurs k times is in NP because one can guess the set and then verify the number of occurrence by counting the occurrences. The proof that the problem is NP-hard uses reduction from the Balanced Complete Bipartite Subgraph Problem: given a positive integer K and a bipartite graph with vertices V and edges E determine whether there are two disjoint sets of edges (V_1 and V_2) such that $|V_1| = K$, $|V_2| = K$, and such that there is an edge in E between each vertex in V_1 and each vertex in V_2. Since any such subgraph must be in a single connected component of the original graph, we can process each connected component of the graph separately. In a single connected component, the vertices of a bipartite graph naturally fall into two groups where all the edges in a group are connected by paths of even length. To map the given single component bipartite graph to baskets and items, associate (in a one to one manner) each vertex of one part with an item, and associate (in a one to one manner) each vertex of the other part with a basket. Have item i in basket b if and only if the vertex associated with i has an edge connecting to the vertex associated with b. If there is solution to the given instance of the Balanced Complete Bipartite Subgraph Problem then that solution directly gives an item set of size K that occurs in K baskets. Also if there is an item set of size K that occurs in K baskets, then the corresponding subgraph is a solution to the given instance. QED.

Eq. 13. We have j ($j \geq k$) baskets that contain the set, $b - j$ baskets that do not contain the set, so

$$S_l = \sum_{j \geq k} \binom{b}{j} [P(l)]^j [1 - P(l)]^{b-j}. \tag{13a}$$

From the binomial theorem we have

$$\sum_{j} \binom{b}{j} [P(l)]^j [1 - P(l)]^{b-j} = 1, \tag{A1}$$

so

$$\sum_{j \geq k} \binom{b}{j} [P(l)]^j [1 - P(l)]^{b-j} = 1 - \sum_{j < k} \binom{b}{j} [P(l)]^j [1 - P(l)]^{b-j}. \tag{13b}$$

Eq. 14. The factor $[P(l)]^{j_0}$ is the probability that the first j_0 baskets obey condition M_0, $[Q(l)]^{j_1}$ is the probability that the next j_1 baskets obey condition M_1, $\ldots$, $[Q(l)]^{j_l}$ is the probability that the next j_l baskets obey condition M_l, and $[1 - P(l) - lQ(l)]^{b-j_0-\cdots-j_l}$ is the probability that the remaining baskets obey none of the conditions M_0, $\ldots$, M_l. (Notice that each basket obeys at most one of the conditions M_h ($0 \leq h \leq l$).) However, the various baskets can come in any order, and the multinomial coefficient allows for this. Thus the probability that j_h baskets obey condition M_h ($0 \leq h \leq l$) and that the remaining $b - j_0 - \cdots - j_l$ baskets do not obey any of the conditions is

$$\binom{b}{j_0, \ldots, j_l, b - j_0 - \cdots - j_l} [P(l)]^{j_0} [Q(l)]^{j_1+\cdots+j_l} [1 - P(l) - lQ(l)]^{b-j_0-\cdots-j_l}. \tag{14}$$

Eq. 20.

$$R_k(b,l,m,n) = \sum_{\substack{j_1 \geq k \\ j_2 \geq k \\ \cdots \\ j_l \geq k}} \binom{b}{j_1, \ldots, j_l, b - j_1 - \cdots - j_l} [Q(m)]^{j_1 + \cdots + j_l} [1 - P(m) - nQ(m)]^{b - j_1 - \cdots - j_l} \tag{19}$$

$$= \sum_{j_1} \binom{b}{j_1} [Q(m)]^{j_1} \times \sum_{\substack{j_2 \geq k \\ j_3 \geq k \\ \cdots \\ j_l \geq k}} \binom{b - j_1}{j_2, \ldots, j_l, b - j_1 - \cdots - j_l} [Q(m)]^{j_2 + \cdots + j_l} [1 - P(m) - nQ(m)]^{(b - j_1) - j_2 - \cdots - j_l} \tag{A2}$$

$$= \sum_{j \geq k} \binom{b}{j} [Q(m)]^j R_k(b - j, l - 1, m, n). \tag{20}$$

Eq. 21. The boundary condition is just eq. 19 with l replaced with 0.

Eq. 29. Using the binomial theorem on the j_l sum in eq. 27, we have

$$x^{-kl} \sum_{j_1, \ldots, j_l} \binom{b}{j_1, \ldots, j_l, b - j_1 - \cdots - j_l} [xQ(m)]^{j_1 + \cdots + j_l} [1 - P(m) - nQ(m)]^{b - j_1 - \cdots - j_l} =$$
$$x^{-kl} \sum_{j_1, \ldots, j_{l-1}} \binom{b}{j_1, \ldots, j_{l-1}, b - j_1 - \cdots - j_{l-1}} [xQ(m)]^{j_1 + \cdots + j_{l-1}} [1 - P(m) - (n - x)Q(m)]^{b - j_1 - \cdots - j_{l-1}}. \tag{A3}$$

The remaining $l - 1$ sums can be done the same way to obtain

$$R_k(b,l,m,n) \leq x^{-kl} \sum_{j_1, \ldots, j_l} \binom{b}{j_1, \ldots, j_l, b - j_1 - \cdots - j_l} [xQ(m)]^{j_1 + \cdots + j_l} [1 - P(m) - nQ(m)]^{b - j_1 - \cdots - j_l}$$
$$\leq x^{-kl} [1 - P(m) - (n - lx)Q(m)]^b. \tag{27}$$

Eq. 30. Less algebra is needed to minimize the logarithm of the bound, and it leads to the same result. Start with the derivative of the logarithm of eq. 25.

$$\frac{d\left\{-k \ln x + b \ln[1 + (x - 1)P(l)]\right\}}{dx} = \frac{-k}{x} + \frac{bP(l)}{1 + P(l)(x - 1)}. \tag{A4}$$

Now set the logarithm to zero and replace x (the free variable) with x_* (the value that results in the derivative being zero).

$$\frac{-k}{x_*} + \frac{bP(l)}{1 + P(l)(x_* - 1)} = 0, \tag{A5}$$

$$-kP(l)(x_* - 1) - k + bP(l)x_* = 0, \tag{A6}$$

$$x_* = \frac{k[1 - P(l)]}{(b - k)P(l)}. \tag{30}$$

Eq. 31.

$$\frac{k[1 - P(l)]}{(b - k)P(l)} > 1, \tag{A7}$$

$$k[1 - P(l)] > (b - k)P(l), \tag{A8}$$

(since $b > k$)

$$k > bP(l). \tag{31}$$

Eq. 32. The derivative of the logarithm of the bound on F (eq. 29) with respect x is

$$\frac{d\,[-kl\ln x - (k-1)\ln y + b\ln[1 + (x^l y - 1)P(l) + l(x-1)Q(l)]}{dx}$$
$$= \frac{-kl}{x} + \frac{b[lx^{l-1}yP(l) + lQ(l)]}{1 + (x^l y - 1)P(l) + l(x-1)Q(l)}. \tag{A9}$$

Setting this to zero gives

$$-kl[1 + (x_*^l y - 1)P(l) + l(x_* - 1)Q(l)] + bx_*[lx_*^{l-1}yP(l) + lQ(l)] = 0, \tag{A10}$$

$$-kl - klP(l)x_*^l y + klP(l) + kl^2Q(l) - kl^2Q(l)x_* + blP(l)x_*^l y + blQ(l)x_* = 0, \tag{A11}$$

$$(b-k)P(l)x_*^l y + (b - kl)Q(l)x_* - k[1 - P(l) - lQ(l)] = 0. \tag{32}$$

Eq. 33. The derivative of the logarithm bound (eq. 29) with respect to y gives

$$\frac{d\,[-kl\ln x - (k-1)\ln y + b\ln[1 + (x^l y - 1)P(l) + l(x-1)Q(l)]}{dy} \tag{A12}$$
$$= -\frac{k-1}{y} + \frac{bx^lP(l)}{1 + (x^l y - 1)P(l) + l(x-1)Q(l)}. \tag{A13}$$

Setting this to zero gives

$$-(k-1)[1 + (x^l y_* - 1)P(l) + l(x-1)Q(l)] + bx^l y_* P(l) = 0, \tag{A14}$$

$$(b - k + 1)P(l)x^l y_* - (k-1)[1 - P(l) + lQ(l)(x-1)] = 0. \tag{33}$$

Eq. 34. Eq. 32 with $x_* = 1$, $y = 1$ is

$$(b-k)P(l) + (b - kl)Q(l) - k[1 - P(l) - lQ(l)] = 0, \tag{A15}$$

$$k = b[P(l) + Q(l)] = bP(l-1). \tag{34}$$

Eq. 35. By implicit differentiation of eq. 32 (with $y = 1$) we have

$$\frac{d\,\{(b-k)P(l)x_*^l + (b-kl)Q(l)x_* - k[1 - P(l) - lQ(l)]\}}{db} = 0, \tag{A16}$$

$$P(l)x_*^l + Q(l)x_* + [(b-k)lP(l)x_*^{l-1} + (b-kl)Q(l)]\frac{d\,x_*}{db} = 0, \tag{A17}$$

$$\frac{d\,x_*}{db} = -\frac{P(l)x_*^l + Q(l)x_*}{(b-k)lP(l)x_*^{l-1} + (b-kl)Q(l)}. \tag{A18}$$

Since x_* solves eq. 32 (with $y = 1$) we have

$$(b-k)P(l)x_*^l + (b-kl)Q(l)x_* = k[1-P(l)-lQ(l)], \tag{A19}$$

which is positive. Thus

$$(b-k)P(l)lx_*^{l-1} + (b-kl)Q(l) = \frac{(b-k)P(l)x_*^l + (b-kl)Q(l)x_*}{x_*} + (l-1)(b-k)P(l)x_*^{l-1} \tag{A20}$$

is also positive (because $x_* > 0$, $b-k > 0$, and $l \geq 1$). Thus, $d\,x_*/db$ is negative. If we start at the b value that results in $x_* = 1$, and decrease b, then x_* increases. Thus, it becomes larger than 1 and stays larger than 1.

Eq. 36. Eq. 33 with $x = 1$ is

$$(b-k+1)P(l)y_* - (k-1)[1-P(l)] = 0, \tag{A21}$$

$$y_* = \frac{(k-1)[1-P(l)]}{(b-k+1)P(l)}. \tag{36}$$

Eq. 37.

$$\frac{(k-1)[1-P(l)]}{(b-k+1)P(l)} < 1, \tag{A22}$$

$$(k-1)[1-P(l)] < (b-k+1)P(l), \tag{A23}$$

$$k-1 < bP(l), \tag{A24}$$

$$k < bP(l) + 1. \tag{37}$$

Eq. 39. From eq. 32

$$P(l)x_*^l y = \frac{k[1-P(l)-lQ(l)] - (b-kl)Q(l)x}{b-k}. \tag{A25}$$

From eq. 33

$$P(l)x^l y_* = \frac{(k-1)[1-P(l)+l(x-1)Q(l)]}{b-k+1}. \tag{A26}$$

Setting x to x_*, y to $y_*$the two right sides equal and clearing fractions gives

$$k(b-k+1)[1-P(l)-lQ(l)] - (b-kl)(b-k+1)Q(l)x = (k-1)(b-k)[1-P(l)+lQ(l)x-lQ(l)], \tag{A27}$$

$$(b^2-bk+b-bkl+k^2l-kl)Q(l)x = (bk-k^2+k)[1-P(l)-lQ(l)]-(bk-k^2-b+k)[1-P(l)+lQ(l)x-lQ(l)], \tag{A28}$$

$$(b^2-bk+b-bkl+k^2l-kl+bkl-k^2l-bl+kl)Q(l)x = (bk-k^2+k-bk+k^2+b-k)[1-P(l)-lQ(l)], \tag{A29}$$

$$(b^2-bk+b-bl)Q(l)x = b[1-P(l)-lQ(l)], \tag{A30}$$

$$x = \frac{1-P(l)-lQ(l)}{(b-k-l+1)Q(l)}. \tag{39}$$

Eq. 40. Plugging the value of x from eq. 39 into eq. 32 gives

$$(b-k)\left(\frac{1-P(l)-lQ(l)}{(b-k-l+1)Q(l)}\right)^l P(l)y + \frac{(b-kl)[1-P(l)-lQ(l)]}{b-k-l+1} - k[1-P(l)-lQ(l)] = 0, \tag{A31}$$

$$y = \frac{k[1 - P(l) - lQ(l)] - \frac{(b-kl)[1-P(l)-lQ(l)]}{b-k-l+1}}{(b-k)\left(\frac{1-P(l)-lQ(l)}{(b-k-l+1)Q(l)}\right)^l P(l)}, \tag{A32}$$

$$y = \frac{(b-k-l+1)^{l-1}Q(l)^l\{k(b-k-l+1)[1-P(l)-lQ(l)] - (b-kl)[1-P(l)-lQ(l)]\}}{(b-k)[1-P(l)-lQ(l)]^l P(l)}. \tag{A33}$$

$$y = \frac{(b-k-l+1)^{l-1}Q(l)^l(bk-k^2-kl+k-b+kl)[1-P(l)-lQ(l)]}{(b-k)[1-P(l)-lQ(l)]^l P(l)}, \tag{A34}$$

$$y = \frac{(b-k-l+1)^{l-1}Q(l)^l(k-1)[1-P(l)-lQ(l)]}{[1-P(l)-lQ(l)]^l P(l)}, \tag{A35}$$

$$y = (k-1)\left(\frac{(b-k-l+1)Q(l)}{1-P(l)-lQ(l)}\right)^{l-1}\frac{Q(l)}{P(l)}. \tag{40}$$

Eq. 41. To have $x > 1$ we need

$$\frac{1-P(l)-lQ(l)}{(b-k-l+1)Q(l)} > 1. \tag{A36}$$

For $b > k+l-1$ we have

$$1 - P(l) - lQ(l) > (b-k-l+1)Q(l), \tag{A37}$$

$$1 - P(l) > (b-k+1)Q(l), \tag{A38}$$

$$b < k - 1 + \frac{1-P(l)}{Q(l)}. \tag{A39}$$

So that the upper and lower bounds look more similar, we rewrite the upper bound on b by adding and subtracting l.

$$k + l - 1 < b < k + l - 1 + \frac{1-P(l)-lQ(l)}{Q(l)}. \tag{41}$$

Suppose $b < k+l-1$. Then from eq. A36 we have

$$1 - P(l) - lQ(l) < (b-k-l+1)Q(l), \tag{A40}$$

$$1 - P(l) < (b-k+1)Q(l), \tag{A41}$$

$$b > k - 1 + \frac{1-P(l)}{Q(l)}, \tag{A42}$$

$$k + l - 1 > b > k - 1 + \frac{1-P(l)}{Q(l)}. \tag{A43}$$

For this range to be non-empty, we need

$$k + l - 1 > k - 1 + \frac{1-P(l)}{Q(l)}, \tag{A44}$$

$$0 > \frac{1-P(l)}{Q(l)} - l, \tag{A45}$$

$$0 > \frac{1-P(l)-lQ(l)}{Q(l)}, \tag{A46}$$

but this can not be. We have $P(l) + lQ(l) = p^l + l(1-p)p^{l-1}$, which are some of the terms in the binomial expansion of $[p + (1-p)]^l = 1$. Since all of the terms are nonnegative (for $0 \geq p \geq 1$) the sum of some of the terms is no more than 1, so the right side of eq. A46 is nonnegative. Thus, the range is always empty.

Eq. 42.

$$b < bP(l-1) + l - 1 + \frac{1 - P(l) - lQ(l)}{Q(l)}, \tag{A47}$$

$$b[1 - P(l-1)] < \frac{1 - P(l) - Q(l)}{Q(l)}, \tag{A48}$$

$$b[1 - P(l-1)] < \frac{1 - P(l-1)}{Q(l)}, \tag{A49}$$

$$b < \frac{1}{Q(l)}. \tag{A50}$$

Eq. 43. To have $y < 1$ we need

$$(k-1)\left(\frac{(b-k-l+1)Q(l)}{1 - P(l) - lQ(l)}\right)^{l-1} \frac{Q(l)}{P(l)} < 1, \tag{A51}$$

For $l \geq 2$

$$\frac{(b-k-l+1)Q(l)}{1 - P(l) - lQ(l)} \left(\frac{(k-1)Q(l)}{P(l)}\right)^{1/(l-1)} < 1, \tag{A52}$$

$$b - k - l + 1 \leq \frac{1 - P(l) - lQ(l)}{Q(l)\left(\frac{(k-1)Q(l)}{P(l)}\right)^{1/(l-1)}}, \tag{A53}$$

$$b < k + l - 1 + \frac{1 - P(l) - lQ(l)}{Q(l)} \left(\frac{P(l)}{(k-1)Q(l)}\right)^{1/(l-1)}. \tag{43}$$

The upper bound on b from eq. 43 is greater than the lower bound from eq. 41.

$$k + l - 1 + \frac{1 - P(l) - lQ(l)}{Q(l)} \left(\frac{P(l)}{(k-1)Q(l)}\right)^{1/(l-1)} > k + l - 1, \tag{A54}$$

$$\frac{1 - P(l) - lQ(l)}{Q(l)} \left(\frac{P(l)}{(k-1)Q(l)}\right)^{1/(l-1)} > 0. \tag{A55}$$

All the terms on the left are positive for $0 < p < 1$.

Eq. 44. We now consider when the upper bound on b from eq. 43 is less than the upper bound from eq. 41.

$$k + l - 1 + \frac{1 - P(l) - lQ(l)}{Q(l)} \left(\frac{P(l)}{(k-1)Q(l)}\right)^{1/(l-1)} < k + l - 1 + \frac{1 - P(l) - lQ(l)}{Q(l)}, \tag{A56}$$

$$\frac{1 - P(l) - lQ(l)}{Q(l)} \left(\frac{P(l)}{(k-1)Q(l)}\right)^{1/(l-1)} < \frac{1 - P(l) - lQ(l)}{Q(l)}, \tag{A57}$$

$$0 < \frac{1 - P(l) - lQ(l)}{Q(l)} \left[1 - \left(\frac{P(l)}{(k-1)Q(l)}\right)^{1/(l-1)}\right]. \tag{A58}$$

As shown above (below eq. A46), the first factor is always positive. For the second factor to be positive we need

$$1 > \left(\frac{P(l)}{(k-1)Q(l)}\right)^{1/(l-1)}, \tag{A59}$$

$$1 > \frac{P(l)}{(k-1)Q(l)}, \tag{A60}$$

$$(k-1)(1-p)p^{l-1} > p^l, \tag{A61}$$

$$(k-1)(1-p) > p, \tag{A62}$$

$$k - kp - 1 + p > p, \tag{A63}$$

$$k(1-p) > 1, \tag{A64}$$

$$k > \frac{1}{1-p}. \tag{44}$$

Eq. 47. Plugging eq. 30 into eq. 25 gives

$$S_l \leq \left(\frac{k[1-P(l)]}{(b-k)P(l)}\right)^{-k} \left[1 + \left(\frac{k[1-P(l)]}{(b-k)P(l)} - 1\right) P(l)\right]^b, \tag{A65}$$

$$S_l \leq \{k[1-P(l)]\}^{-k}[P(l)]^k(b-k)^{-b+k}(b-k+\{k[1-P(l)]-(b-k)P(l)\})^b,$$

$$S_l \leq \left(\frac{P(l)}{k}\right)^k \left(\frac{1-P(l)}{b-k}\right)^{b-k} b^b. \tag{47}$$

Eq. 49. Replace k in eq. 47 with its value in terms of α_1 (eq. 48)

$$S_l \leq \left(\frac{P(l)}{b[P(l)+\alpha_1]}\right)^{b[P(l)+\alpha_1]} \left(\frac{1-P(l)}{b-b[P(l)+\alpha_1]}\right)^{b-b[P(l)+\alpha_1]} b^b, \tag{A66}$$

$$S_l \leq \left(\frac{P(l)}{b[P(l)+\alpha_1]}\right)^{b[P(l)+\alpha_1]} \left(\frac{1-P(l)}{b[1-P(l)-\alpha_1]}\right)^{b[1-P(l)-\alpha_1]} b^b, \tag{A67}$$

$$S_l \leq \left(\frac{1}{b[1+\alpha_1/P(l)]}\right)^{b[P(l)+\alpha_1]} \left(\frac{1}{b\{1-\alpha_1/[1-P(l)]\}}\right)^{b[1-P(l)-\alpha_1]} b^b, \tag{A68}$$

$$S_l \leq \left(\frac{1}{1+\alpha_1/P(l)}\right)^{b[P(l)+\alpha_1]} \left(\frac{1}{1-\alpha_1/[1-P(l)]}\right)^{b[1-P(l)-\alpha_1]}. \tag{A69}$$

To further simplify this, we will write it as $S_l \leq e^X$ with

$$X = \ln\left[\left(\frac{1}{1+\alpha_1/P(l)}\right)^{b[P(l)+\alpha_1]} \left(\frac{1}{1-\alpha_1/[1-P(l)]}\right)^{b[1-P(l)-\alpha_1]}\right] \tag{A70}$$

$$= -b[P(l)+\alpha_1]\ln\left(1+\frac{\alpha_1}{P(l)}\right) - b[1-P(l)-\alpha_1]\ln\left(1-\frac{\alpha_1}{[1-P(l)]}\right). \tag{A71}$$

Dividing by b, we have

$$\frac{X}{b} = -[P(l)+\alpha_1]\ln\left(1+\frac{\alpha_1}{P(l)}\right) - [1-P(l)-\alpha_1]\ln\left(1-\frac{\alpha_1}{[1-P(l)]}\right) \tag{A72}$$

$$= -[P(l)+\alpha_1]\left[\left(\frac{\alpha_1}{P(l)}\right) - \frac{1}{2}\left(\frac{\alpha_1}{P(l)}\right)^2 + O\left(\left(\frac{\alpha_1}{P(l)}\right)^3\right)\right] \tag{A73}$$

$$+ [1-P(l)-\alpha_1]\left[\left(\frac{\alpha_1}{1-P(l)}\right) + \frac{1}{2}\left(\frac{\alpha_1}{1-P(l)}\right)^2 + O\left(\left(\frac{\alpha_1}{1-P(l)}\right)^3\right)\right] \tag{A74}$$

$$= -\alpha_1 + \frac{\alpha_1^2}{2P(l)} - O\left(\frac{\alpha_1^3}{P(l)^2}\right) - \frac{\alpha_1^2}{P(l)} + \frac{\alpha_1^3}{2P(l)^2} - O\left(\frac{\alpha_1^4}{P(l)^3}\right)$$

$$+ \alpha_1 + \frac{\alpha_1^2}{2[1-P(l)]} + O\left(\frac{\alpha_1^3}{[1-P(l)]^2}\right) - \frac{\alpha_1^2}{1-P(l)} - \frac{\alpha_1^3}{2[1-P(l)]^2} - O\left(\frac{\alpha_1^4}{[1-P(l)]^3}\right) \tag{A75}$$

$$= -\frac{\alpha_1^2}{2P(l)} - \frac{\alpha_1^2}{2[1-P(l)]} + O\left(\frac{\alpha_1^3}{[1-P(l)]^2}\right) - O\left(\frac{\alpha_1^3}{P(l)^2}\right). \tag{A76}$$

The big O is with respect to α_1. We assume that $0 < p < 1$. Since negative big O terms can be dropped in an upper limit,

$$S_l \le e^{-b\alpha_1^2/\{2P(l)[1-P(l)]\}+O(b\alpha_1^3[1-P(l)]^{-2})}. \tag{49}$$

Eq. 53. Eq. 32 with $y=1$ is

$$(b-k)P(l)x^l + (b-kl)Q(l)x - k[1-P(l)-lQ(l)] = 0. \tag{A77}$$

Let $x = 1+\delta$ with small δ and expand to second order. Let θ stand for quantities that approaches 1 in the limit as δ approaches 0. In other words, θ is short hand for $[1+o(1)]$, where δ is the variable that is approaching zero.

$$(b-k)P(l)\left(1+l\delta+\frac{l(l-1)\delta^2\theta}{2}\right) + (b-kl)Q(l)(1+\delta) = k[1-P(l)-lQ(l)]. \tag{A78}$$

$$(b-k)P(l)\left(l\delta+\frac{l(l-1)\delta^2\theta}{2}\right) + (b-kl)Q(l)\delta = k[1-P(l)-lQ(l)] - (b-k)P(l) - (b-kl)Q(l), \tag{A79}$$

$$\delta = \frac{k[1-P(l)-lQ(l)] - (b-k)P(l) - (b-kl)Q(l)}{l(b-k)P(l)[1+(l-1)\theta\delta/2] + (b-kl)Q(l)}, \tag{A80}$$

$$\delta = \frac{k - bP(l) - bQ(l)}{b[lP(l)+Q(l)] - kl[P(l)+Q(l)] + (b-k)l(l-1)P(l)\theta\delta/2}, \tag{A81}$$

$$\delta = \frac{k - bP(l-1)}{b[lP(l)+Q(l)] - klP(l-1) + (b-k)l(l-1)P(l)\theta\delta/2}, \tag{A82}$$

$$\delta = \frac{k-bP(l-1)}{b[lP(l)+Q(l)] - klP(l-1)}\left(1+\frac{(b-k)l(l-1)P(l)\theta\delta/2}{b[lP(l)+Q(l)] - klP(l-1)}\right)^{-1}, \tag{A83}$$

$$\delta = \frac{k-bP(l-1)}{b[lP(l)+Q(l)] - klP(l-1)}\left(1+\frac{[k-bP(l-1)](b-k)l(l-1)P(l)\theta/2}{\{b[lP(l)+Q(l)] - klP(l-1)\}^2}\right)^{-1}. \tag{53}$$

Eq. 55. Write eq. 52 as

$$F_l \le e^X \tag{A84}$$

with

$$X = \ln\{x^{-kl}[1 + (x^l - 1)P(l) + l(x-1)Q(l)]^b\} \tag{A85}$$

$$= -kl \ln x + b \ln[1 + (x^l - 1)P(l) + l(x-1)Q(l)]. \tag{A86}$$

Replace x with $1 + \delta$.

$$X = -kl \ln(1+\delta) + b \ln\{1 + [(1+\delta)^l - 1]P(l) + lQ(l)\delta\}. \tag{A87}$$

Expanding X in a power series to second order gives

$$\begin{aligned} X &= -kl \ln(1+\delta) + b \ln\left(1 + l\delta P(l) + \frac{l(l-1)P(l)\theta\delta^2}{2} + lQ(l)\delta\right) \\ &= -kl\delta + \frac{kl\theta\delta^2}{2} + b\left(lP(l)\delta + \frac{l(l-1)\delta^2 P(l)\theta}{2} + lQ(l)\delta\right) \\ &\quad - \frac{b}{2}\left(lP(l)\delta + \frac{l(l-1)P(l)\theta\delta^2}{2} + lQ(l)\delta\right)^2 \theta \qquad \text{(A88)} \\ &= -l[k - bP(l) - bQ(l)]\delta + \frac{kl + bl(l-1)P(l) - bl^2[P(l)+Q(l)]^2}{2}\theta\delta^2 \qquad \text{(A89)} \\ &= -l[k - bP(l-1)]\delta + \frac{kl + bl(l-1)P(l) - bl^2[P(l-1)]^2}{2}\theta\delta^2. \qquad \text{(A90)} \end{aligned}$$

Replace k by its definition in terms of α_3 (eq. 54) to obtain

$$\begin{aligned} X &= -l[k - bP(l-1)]\delta + \frac{\{kl + bl(l-1)P(l) - bl^2[P(l-1)]^2\}\theta}{2}\delta^2 \qquad \text{(A91)} \\ &= -bl\alpha_3\delta + \frac{bl\{P(l-1) + \alpha_3 + (l-1)P(l) - l[P(l-1)]^2\}\theta}{2}\delta^2 \qquad \text{(A92)} \\ &= \left(-bl\alpha_3 + \frac{bl\{P(l-1) + \alpha_3 + (l-1)P(l) - l[P(l-1)]^2\}\theta\delta}{2}\right)\delta. \qquad \text{(A93)} \end{aligned}$$

Also replace k in eq. 53 by its value in terms of α_3 to obtain

$$\begin{aligned} \delta &= \frac{\alpha_3 b}{b[lP(l) + Q(l)] - b[P(l-1) + \alpha_3]lP(l-1)} \\ &\qquad \times \left(1 + \frac{b\alpha_3\{b - b[P(l-1)+\alpha_3]\}l(l-1)P(l)\theta/2}{\{b[lP(l)+Q(l)] - b[P(l-1)+\alpha_3]l[P(l-1)]\}^2}\right)^{-1} \qquad \text{(A94)} \\ &= \frac{\alpha_3}{lP(l) + Q(l) - l[P(l-1) + \alpha_3]P(l-1)} \\ &\qquad \times \left(1 + \frac{\alpha_3[1 - P(l-1) - \alpha_3]l(l-1)P(l)\theta/2}{\{lP(l)+Q(l) - l[P(l-1)+\alpha_3][P(l-1)]\}^2}\right)^{-1}. \qquad \text{(A95)} \end{aligned}$$

Since δ and α_3 go to zero together, this can be written as

$$\delta = \frac{\theta\alpha_3}{lP(l) + Q(l) - l[P(l-1)]^2}. \tag{A96}$$

Plugging the value of δ into the expression for X (eq. A93) gives

$$X = \left(-bl\alpha_3 + \frac{bl\{P(l-1) + \alpha_3 + (l-1)P(l) - l[P(l-1)]^2\}\theta\delta}{2}\right)\delta \tag{A93}$$

$$= \left(-bl\alpha_3 + \frac{bl\{P(l-1) + (l-1)P(l) - l[P(l-1)]^2 + \alpha_3\}\theta\alpha_3}{2\{P(l-1) + (l-1)P(l) - l[P(l-1)]^2\}}\right)\frac{\theta\alpha_3}{P(l-1) + (l-1)P(l) - l[P(l-1)]^2} \tag{A97}$$

$$= \left(-bl\alpha_3 + \frac{bl\theta\alpha_3}{2} + \frac{bl\theta\alpha_3^2}{2\{P(l-1) + (l-1)P(l) - l[P(l-1)]^2\}}\right)\frac{\theta\alpha_3}{P(l-1) + (l-1)P(l) - l[P(l-1)]^2} \tag{A98}$$

$$= -\frac{bl\theta\alpha_3^2}{2\{P(l-1) + (l-1)P(l) - l[P(l-1)]^2\}}. \tag{A99}$$

Thus,

$$F_l \le e^{-bl\theta\alpha_3^2/(2\{P(l-1)+(l-1)P(l)-l[P(l-1)]^2\})}. \tag{55}$$

Eq. 56. The derivation of eq. 55 requires that $\delta = o(1)$. The step from eq. A98 to eq. A99 requires that α_3 be small compared to some other terms. Both conditions imply

$$\alpha_3 = \{lP(l) + Q(l) - l[P(l-1)]^2\}o(1). \tag{56}$$

Eq. 58. By inclusion-exclusion, the sum for the region that defines R_k is equal to the sum over the entire area ($r_k(b,l,m,n,0)$), minus the sums over the various regions where a single j is required to be outside of R_k's region (l copies of $r_k(b,l,m,n,1)$, plus the sums over regions where two j's are required to be outside of R_k's region, etc.

Eq. 60.

$$r_k(b,l,m,n,h) = \sum_{\substack{j_1<k \\ j_2<k \\ \cdots \\ j_h<k \\ j_{h+1},\ldots,j_l}} \binom{b-j_1}{j_2,\ldots,j_l,b-j_1-\cdots-j_l}[Q(m)]^{j_1+\cdots+j_l}[1-P(m)-nQ(m)]^{b-j_1-\cdots-j_l} \tag{59}$$

$$= \sum_{\substack{j_1<k \\ j_2<k \\ \cdots \\ j_h<k \\ j_{h+1},\ldots,j_{l-1}}} \binom{b-j_1}{j_2,\ldots,j_{l-1},b-j_1-\cdots-j_{l-1}}[Q(m)]^{j_1+\cdots+j_{l-1}}[1-P(m)-(n-1)Q(m)]^{b-j_1-\cdots-j_{l-1}} \tag{A100}$$

$\ldots$

$$= \sum_{\substack{j_1<k \\ j_2<k \\ \cdots \\ j_h<k}} \binom{b-j_1}{j_2,\ldots,j_h,b-j_1-\cdots-j_h}[Q(m)]^{j_1+\cdots+j_h}[1-P(m)-(n-l+h)Q(m)]^{b-j_1-\cdots-j_h}. \tag{60}$$

Bound on first term of eq. 63. For

$$\sum_{j_0<k}\binom{b}{j_0}[P(l)]^{j_0}[1-P(l)]^{b-j_0}$$

use the Chernoff bound from eq. 25.

Eq. 64.

$$\sum_{j_0<k}\binom{b}{j_0}[P(l)]^{j_0}\sum_{j<k}\binom{b-j_0}{j}[Q(l)]^j[1-P(l)-Q(l)]^{b-j_0-j}$$

$$\leq\sum_{j_0,j}\binom{b}{j_0}[P(l)]^{j_0}\binom{b-j_0}{j}[Q(l)]^j[1-P(l)-Q(l)]^{b-j_0-j}x^j y^{j_0} \tag{A101}$$

$$\leq x^{-kl}y^{-k+1}[1+P(l)(x^l y-1)+(x-1)Q(l)]^b. \tag{A102}$$