

Average Case Performance of the Apriori Algorithm

Paul Purdom Dirk Van Gucht
Computer Science Department
Indiana University

Abstract: The Apriori Algorithm examines baskets of items to determine which subsets of the items occur in lots of baskets. Suppose we wish to determine which items sets occur in at least k baskets. The algorithm considers item sets of size l in the order $l = 1, 2, \dots$. The only way this algorithm can determine that a set occurs at least k times is to count the k occurrences, but it sometimes determines (without counting) that a set occurs less than k times by noticing that some subsets of the l items occur less than k times. For algorithms that require explicit counting to verify the k occurrences, it is useful to separate the total time into the “success time”; that is used to verify k occurrences, and the “failure time”; that is used to process sets which have less than k occurrences. This paper derives both exact and asymptotic formulas for both success and failure times in the case where the baskets are filled randomly with probability p (each shopper independently buys each item).

The Apriori Algorithm considers almost every possible set of l items for those l where $k \leq bp^l$ and almost no sets for larger l . For most applications the largest l such that $k \leq bp^l$ is not very large. When it is less than one half of the number of items (essentially the only case of interest), the work associated with this largest such l dominates the running time. The probability that a particular set needs processing approaches zero at a rate that is a negative exponential function of the square of the difference $bp^l - k$ when k is above bp^l . When k is large compared to 1, the probability that the set needs processing approaches 1 at a similar negative exponential rate.

1. Introduction

The Apriori Algorithm [1, 2, 3, 8] 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 [1, 2, 3, 8, 12]. Other data mining problems based on the Apriori Algorithm are discussed in [7, 8, 10, 13, 14].

Let J_l be a subset of size $l \geq 1$ that is selected from the $|I|$ items. For a particular set 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 *pretest*. For those sets J_l that pass the pretest, the algorithm examines the list of basket contents and counts the number of baskets which contain the set J_l to determine whether the set of items occurs in at least k baskets. This counting and comparing with the threshold is called the *final test*. The algorithm remembers the results of the final test for use by the pretests that occur when l is increased.

In this paper we do 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 study the expected value of S_l [9]. In real life, the Apriori Algorithm is used to analyze data that is more complex than this. Presumably, no one is interested in running the algorithm on truly random data. Rather, they are 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 some of the main features of the performance of the algorithm. The Apriori Algorithm is designed to take advantage

of random properties of the data rather than to take advantage of any fixed structure that the data might have. In particular, for worst-case data, the pretest of the Apriori Algorithm is not effective.

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. The disadvantage is that experimental studies are needed to verify the extent to which performance on model data predicts the performance on real data. Since different sources of real data usually have different characteristics, theoretical studies can interact with experimental studies to suggest different types of data that should be studied.

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 passes the final test and
2. *Failure rate*: the probability that a set passes the pretest but fails the final 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, but it is hard to see any other way that would be efficient on the types of data where the Apriori Algorithm is used. 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 . One can start with the largest set (the set of all items) and work down to the smaller ones. This leads to an algorithm that needs to count which sets occur less than k times. The Apriori Algorithm is efficient when the sets that pass the test are small, the complementary algorithm is efficient when they are large. Neither the Apriori Algorithm nor its complement is efficient when the maximal sets with k occurrences have about half of the items.

Any algorithm for the problem solved by the Apriori Algorithm needs exponential time in the worst case. If every shopper buys every item, the algorithm must output each subset of the I items. The problem, however, remains hard even when the required output is small. Thus, a subpart of the problem, determining whether or not there are any sets of size l that occur k times, is NP-complete because the Balanced Complete Bipartite Subgraph Problem [5] reduces to it. The appendix has the details of proofs for this and many other statements. (Most proofs are derivations for numbered equations.)

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 final 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 [1] for more discussion of the techniques used in good implementations.

Let S_l be the probability that the set consisting of items 1 to l passes the final test, and F_l be the probability that the same set passes the pretest but fails the final test. 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

$$\sum_{1 \leq l \leq |I|} \binom{|I|}{l} S_l, \quad (1)$$

and the expected number of failures is

$$\sum_{1 \leq l \leq |I|} \binom{|I|}{l} F_l. \quad (2)$$

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

$$\sum_{1 \leq l \leq |I|} \binom{|I|}{l} (S_l + F_l). \quad (3)$$

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

$$\sum_{1 \leq l \leq |I|} (l + 1) \binom{|I|}{l} (S_l + F_l). \quad (4)$$

Most of the memory use in the algorithm is for the data (the basket contents) and for the success sets (also called frequent item sets). Each success set for level l can be stored in $(l + 1)$ words of $\lg |I|$ bits. Although the algorithm needs to output all the success sets, it needs to remember those only for the current level and the previous level. For quick access to this information, a hash table should be used, resulting in perhaps two more words per success set. If we allow h words per item set for hashing overhead, then this leads to a memory requirement bound for storing success sets of $\max_l \left\{ (l + h) \binom{|I|}{l-1} S_{l-1} + (l + h + 1) \binom{|I|}{l} S_l \right\}$ words, but usually this is not much less than

$$\sum_l (l + h + 1) \binom{|I|}{l} S_l, \quad (5)$$

which is the space needed if each answers is retained during the whole computation.

3. Exact calculations

Define the following conditions with respect to a single basket:

- M_0 : the basket has all the items 1 to l and
- M_h ($1 \leq h \leq l$): 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 , $0 \leq h \leq l$.

The probability that a randomly filled basket obeys condition M_0 is

$$P(l) = p^l. \quad (6)$$

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

$$Q(l) = p^{l-1} (1 - p). \quad (7)$$

Note that

$$P(l - 1) = P(l) + Q(l). \quad (8)$$

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 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

$$S_l = \sum_{j \geq k} \binom{b}{j} [P(l)]^j [1 - P(l)]^{b-j} = 1 - \sum_{j < k} \binom{b}{j} [P(l)]^j [1 - P(l)]^{b-j}. \quad (9)$$

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

$$\binom{b}{j_0, \dots, j_l, b - j_0 - \dots - j_l} [P(l)]^{j_0} [Q(l)]^{j_1 + \dots + j_l} [1 - P(l) - lQ(l)]^{b - j_0 - \dots - j_l}, \quad (10)$$

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, \dots , set l has j_l baskets, and $b - j_0 - \dots - j_l$ baskets are not in any of the $l + 1$ sets.

The item set $\{1, \dots, l\}$ will be tested on level l if and only if each condition M_h ($1 \leq h \leq l$) is satisfied by at least k baskets. Condition M_1 is passed when $j_0 + j_1 \geq k$. 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, set $\{1, \dots, l\}$ will be tested in just those cases where the conditions

$$j_0 + j_1 \geq k, j_0 + j_2 \geq k, \dots, j_0 + j_l \geq k \quad (11)$$

are all true. The probability that the set $\{1, \dots, l\}$ is tested is the above probability (eq. 10) summed over those cases that satisfy the conditions (eq. 11),

$$T_l = \sum_{\substack{j_0 \\ j_1 \geq k - j_0 \\ j_2 \geq k - j_0 \\ \dots \\ j_l \geq k - j_0}} \binom{b}{j_0, \dots, j_l, b - j_0 - \dots - j_l} [P(l)]^{j_0} [Q(l)]^{j_1 + \dots + j_l} [1 - P(l) - lQ(l)]^{b - j_0 - \dots - j_l}. \quad (12)$$

Since the set $\{1, \dots, j_l\}$ either does or does not occurs in at least k baskets, the probability that the set $\{1, \dots, l\}$ passes the pretest but fails the final test is

$$\begin{aligned} F_l &= T_l - S_l \\ &= \sum_{\substack{j_0 < k \\ j_1 \geq k - j_0 \\ j_2 \geq k - j_0 \\ \dots \\ j_l \geq k - j_0}} \binom{b}{j_0, \dots, j_l, b - j_0 - \dots - j_l} [P(l)]^{j_0} [Q(l)]^{j_1 + \dots + j_l} [1 - P(l) - lQ(l)]^{b - j_0 - \dots - j_l}. \end{aligned} \quad (13)$$

3.1. 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. 9) 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. 13 is $O(kb^l)$. However, using the recurrence equations below, F can be computed in polynomial time.

Write eq. 13 as

$$F_l = \sum_{j_0 < k} \binom{b}{j_0} [P(l)]^{j_0} R_{k-j_0}(b - j_0, l, l, l) \quad (14)$$

where

$$R_k(b, l, m, n) = \sum_{\substack{j_1 \geq k \\ j_2 \geq k \\ \dots \\ j_l \geq k}} \binom{b}{j_1, \dots, j_l, b - j_1 - \dots - j_l} [Q(m)]^{j_1 + \dots + j_l} [1 - P(m) - nQ(m)]^{b - j_1 - \dots - j_l}. \quad (15)$$

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

$$R_k(b, l, m, n) = \sum_{j \geq k} \binom{b}{j} [Q(m)]^j R_k(b - j, l - 1, m, n) \quad (16)$$

with boundary condition

$$R_k(b, 0, m, n) = [1 - P(m) - nQ(m)]^b. \quad (17)$$

With these equations, a particular $R_k(b, l, m, n)$ can be computed from the various $R_k(c, l - 1, m, n)$ ($k \leq c \leq b$) in $O(b)$ operations. To compute R by repeated application of eq. 17, 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 $O(lb^2)$.

4. Approximations

4.1. Chernoff bounds

The sums for S_l and F_l are incomplete binomial sums. They do not have closed forms (implied by [6]), but, as we show below, Chernoff techniques [4] lead to useful approximations. For

$$L(i) = \begin{cases} 1, & i \leq k, \\ 0, & i > k; \end{cases} \quad \text{and} \quad U(i) = \begin{cases} 0, & i < k, \\ 1, & i \geq k; \end{cases} \quad (18)$$

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

$$\sum_{0 \leq i \leq k} a_i = \sum_{0 \leq i \leq n} a_i L(i) \quad \text{and} \quad \sum_{k \leq i \leq n} a_i = \sum_{0 \leq i \leq n} a_i U(i). \quad (19)$$

In addition, when each $a_i \geq 0$, replacing $L(i)$ (or $U(i)$) with a pointwise upper bound gives an upper bound on the sum. Chernoff [4] noticed that useful bounds for partial binomial sums result when one uses

$$L(i) = x^{-k+i} \quad \text{with} \quad x \leq 1 \quad \text{and} \quad U(i) = x^{-k+i} \quad \text{with} \quad x \geq 1, \quad (20)$$

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

The Chernoff bound for S is

$$S_l \leq x^{-k} \sum_j \binom{b}{j} [xP(l)]^j [1 - P(l)]^{b-j} = x^{-k} [1 + (x - 1)P(l)]^b \quad (21)$$

for any $x \geq 1$.

A Chernoff bound for R is

$$R_k(b, l, m, n) \leq x^{-kl} \sum_{j_1, \dots, j_l} \binom{b}{j_1, \dots, j_l, b - j_1 - \dots - j_l} [xQ(m)]^{j_1 + \dots + j_l} [1 - P(m) - nQ(m)]^{b - j_1 - \dots - j_l} \quad (22)$$

$$\leq x^{-kl} [1 - P(m) - (n - lx)Q(m)]^b \quad (23)$$

for any $x \geq 1$.

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

$$F_l \leq y^{-k+1} \sum_j \binom{b}{j} [yP(l)]^j x^{-(k-j)l} [1 - P(l) + l(x - 1)Q(l)]^{b-j} \quad (24)$$

$$\leq x^{-kl} y^{-k+1} [1 + (x^l y - 1)P(l) + l(x - 1)Q(l)]^b, \quad (25)$$

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

4.2. Regions and borders for S_l

To find the optimum value for x in eq. 21 we start by taking the derivative of the bound with respect to x , setting the result to zero, and solving for x . This gives

$$x = \frac{k[1 - P(l)]}{(b - k)P(l)}. \quad (26)$$

In addition, we check whether the resulting x is strictly within range ($x > 1$). This is the case when

$$k > bP(l). \quad (27)$$

This completes the first stage of finding the Chernoff approximation to S_l . The second stage, which is done in Section 4.4.1, 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 inside the boundary. Thus, S_l is extremely small inside the region defined by eq. 27 except near the boundary. We show in Section 4.4.2 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 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 or large). Sections 4.4.1 and 4.4.2 are needed to determine the details (just how small or large).

4.3. Regions and borders for F_l

To find the optimum value for x and y in eq. 25 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

$$(b - k)P(l)x^l y + (b - kl)Q(l)x - k[1 - P(l) - lQ(l)] = 0. \quad (28)$$

We want the y that satisfies

$$(b - k + 1)P(l)x^l y - (k - 1)[1 - P(l) + lQ(l)(x - 1)] = 0. \quad (29)$$

Consider when x and y are strictly within range ($x > 1, y < 1$). Logically, there are four regions to investigate.

1. eq. 28 with $y = 1$, optimizing $x > 1$ (pretest effective);
2. eq. 29 with $x = 1$, optimizing $y < 1$ (final test easy);
3. eqs. 28 and 29, optimizing $x > 1$, optimizing $y < 1$ (both effects); and
4. $x = 1, y = 1$ (slow).

In eqs. 28 and 29, x is associated with the effectiveness of the pretest and y is associated with the probability of a set failing the final test. When both x and y are 1 the bound for F_l is the trivial bound of 1; 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 4.5 that the bound on F_l is an exponential function of the square of the distance of x or y from the boundary, so F_l rapidly becomes extremely small as x or y moves away from the boundary. In Region 1, F_l is small because the pretest is effective. In Region 2, F_l is small because S_l is near one and F_l can never be larger than $1 - S_l$ (failure requires passing the pretest and failing the final test, which does not happen when the final test is passed). In Region 3, F_l is small for both reasons at the same time. Region 3 includes all points in the intersection of Regions 1 and 2. It also includes some points that are in the union of Regions 1 and 2, but it does not include any points outside of their union. Region 4 is everything that is outside of Region 1 and 2, so no separate study is needed to find its boundary. We now find the boundaries of the regions.

4.3.1. Pretest effective

When the optimizing $x > 1$, the pretest in the Apriori Algorithm (Step 2) is effective. To find when this occurs, notice that eq. 28 is satisfied by $x = 1, y = 1$ when

$$k = b[P(l) + Q(l)] = bP(l - 1). \quad (30)$$

As b decreases, the x that solves eq. 28 increases. This implies that, for $y = 1$, $x > 1$ when

$$k > bP(l-1). \quad (31)$$

4.3.2. Final test easy

When the optimizing $y < 1$, the Apriori Algorithm is efficient in the sense that few item sets fail the test. When $x = 1$, the solution to eq. 29 is

$$y = \frac{(k-1)[1-P(l)]}{(b-k+1)P(l)}. \quad (32)$$

This results in $y < 1$ when

$$k < bP(l) + 1. \quad (33)$$

For most parameters values, the regions of eq. 31 and eq. 33 do not overlap. However, subtracting the right side of eq. 33 from the right side of 31, we find that they do overlap when

$$bQ(l) < 1. \quad (34)$$

This happen both when p^{l-1} is small ($1/b$ is small enough) and also when $1-p$ is small ($1/b$ is small enough). When eq. 34 is true, 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 (one where S_l is near 1) are discussed in section 4.5.4.

4.3.3. Both effects

To find values for the parameters such that $x > 1$ and $y < 1$ optimize the bound, we need to satisfy eqs. 28 and 29 simultaneously. This results in the values

$$x = \frac{1 - P(l) - lQ(l)}{(b - k - l + 1)Q(l)}, \quad (35)$$

$$y = (k-1) \left(\frac{(b-k-l+1)Q(l)}{1-P(l)-lQ(l)} \right)^{l-1} \frac{Q(l)}{P(l)}. \quad (36)$$

We have $x > 1$ when

$$k+l-1 < b < k+l-1 + \frac{1-P(l)-lQ(l)}{Q(l)}. \quad (37)$$

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

All solutions to eq. 37 are in the union of Regions 1 (eq. 31) and 2 (eq. 33). The smallest k that satisfies eq. 31 is k just above $bP(l-1)$. This value for k satisfies eq. 37 when

$$b < \frac{1}{Q(l)}. \quad (38)$$

Eq. 38 is true under the same conditions that eq. 34 is true. Thus, 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, they include all k values.

For $k = 1$, eq. 36 implies that $y = 0$, which is less than 1. For $l = 1$ eq. 36 has no solutions. For $k \geq 2$ and $l \geq 2$, eq. 36 implies $y < 1$ when

$$b < k+l-1 + \frac{1-P(l)-lQ(l)}{Q(l)} \left(\frac{P(l)}{(k-1)Q(l)} \right)^{1/(l-1)}. \quad (39)$$

For parameter values to be in Region 3, both eqs. 37 and 39 must be satisfied.

The upper bound on b from eq. 39 is greater than the lower bound from eq. 37. The upper bound on b from eq. 39 is less than the upper bound from eq. 37 when

$$k > \frac{1}{1-p}. \quad (40)$$

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

Since

$$\frac{1 - P(l) - lQ(l)}{Q(l)} \left(\frac{P(l)}{(k-1)Q(l)} \right)^{1/(l-1)} > 0, \quad (41)$$

any $k \geq b - l + 1$ always satisfies eq. 39. For $l = 2$, this rightmost term from eq. 39 reduces to

$$\frac{1}{k-1}, \quad (42)$$

which is less than 1 for $k \geq 2$. Thus, for $l = 2$, the only solution to eq. 39 is $k \geq b - l + 1$.

The left most term of the right side of eq. 39 (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. 41) 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.

4.4. Exponents for S_l

Section 4.2 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).

4.4.1. 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. 26 into the bound from eq. 21 we obtain

$$S_l \leq \left(\frac{P(l)}{k} \right)^k \left(\frac{1 - P(l)}{b - k} \right)^{b-k} b^b \quad (43)$$

so long as $x \geq 1$. By eq. 27 the condition $x > 1$ is equivalent to $k > bP(l)$, so we will define α_1 by

$$k = b[P(l) + \alpha_1]. \quad (44)$$

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

$$S_l \leq e^{-b\alpha_1^2/\{2P(l)[1-P(l)]\} + O(b\alpha_1^3[1-P(l)]^{-2})} \quad (45)$$

when $\alpha_1 > 0$.

4.4.2. 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. 9. Shift the relation between k and α_1 by one so that α_2 is defined by

$$k = b[P(l) - \alpha_2] - 1. \quad (46)$$

We can now modify the derivation of eq. 45 (with $x < 1$) to obtain

$$S_l \geq 1 - e^{-b\alpha_2^2/\{2P(l)[1-P(l)]\} + O(b\alpha_2^3[1-P(l)]^{-2})} \quad (47)$$

when $\alpha_2 > 0$.

4.5. Exponents for F_l

Section 4.3 found the boundary between the region where F_l is small and where it is large. We now compute just how small (with an upper bound) or small (with a lower bound).

4.5.1. Region 1

When $k > bP(l-1)$ we are in Region 1 of Section 4.3 and the pretest is effective. We now give an upper bound on F_l to show that it is near 0 in this case.

By eq. 25 with $y = 1$

$$F_l \leq x^{-kl} [1 + (x^l - 1)P(l) + l(x - 1)Q(l)]^b. \quad (48)$$

(Note that bounds on F_l obtained with $y = 1$ are also bounds on $T_l = F_l + S_l$. The definition for T_l (eq. 12) 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. 28. Solve eq. 28 (with $y = 1$) for x with $x = 1 + \delta$ and small δ . Let θ stand for any 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.)

$$\delta = \frac{k - bP(l) - bQ(l)}{b[lP(l-1)] - kl[P(l-1)]} \left(1 + \frac{[k - bP(l) - bQ(l)](b - k)l(l-1)P(l)\theta/2}{\{b[lP(l) + Q(l)] - kl[P(l-1)]\}^2} \right)^{-1}. \quad (49)$$

Define α_3 by

$$k = b[P(l-1) + \alpha_3]. \quad (50)$$

In eq. 48 replace k by its value in terms of α_3 and in plug the value of x implied by eq. 49 to obtain

$$F_l \leq e^{-bl\theta\alpha_3^2 / (2\{P(l-1) + (l-1)P(l) - l[P(l-1)]^2\})} \quad (51)$$

when α_3 is small enough, i.e.,

$$\alpha_3 = \{lP(l) + Q(l) - l[P(l-1)]^2\}o(1). \quad (52)$$

4.5.2. Region 2

When $k < bP(l) + 1$ we are in Region 2 of Section 4.3 and by eq. 47 nearly all item sets pass the final test. Since an item set must first pass the pretest and then fail the final test, F_l can be no larger than $1 - S_l$, which (by eq. 47) gives the bound

$$F_l \leq e^{-b\alpha_2^2 / \{2P(l)[1-P(l)] + O(\alpha_2^3 b[1-P(l)]^{-2}\}}, \quad (53)$$

where α_2 is defined by $k = b[P(l) - \alpha_2] - 1$ (eq. 46).

4.5.3. 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.

4.5.4. Region 4

When $bP(l) < k < bP(l-1)$ we are in Region 4 of Section 4.3. The pretest is not effective and also very few item sets pass the final test. We now give a lower bound on F_l to show that there are cases where it is near 1.

In eq. 15, the quantity R_k is defined by sums where each $j_i \geq k$ (for $1 \leq i \leq l$). Using inclusion-exclusion arguments, an alternate way to compute R_k is

$$R_k(b, l, m, n) = \sum_h (-1)^h \binom{l}{h} r_k(b, l, m, n, h), \quad (54)$$

where

$$\begin{aligned}
r_k(b, l, m, n, h) &= \sum_{\substack{j_1 < k \\ j_2 < k \\ \dots \\ j_h < k \\ j_{h+1}, \dots, j_l}} \binom{b - j_1}{j_2, \dots, j_l, b - j_1 - \dots - j_l} [Q(m)]^{j_1 + \dots + j_l} [1 - P(m) - nQ(m)]^{b - j_1 - \dots - j_l} \\
&= \sum_{\substack{j_1 < k \\ j_2 < k \\ \dots \\ j_h < k}} \binom{b - j_1}{j_2, \dots, j_l, b - j_1 - \dots - j_l} [Q(m)]^{j_1 + \dots + j_h} [1 - P(m) - (n - l + h)Q(m)]^{b - j_1 - \dots - j_h}.
\end{aligned} \tag{55}$$

$$\tag{56}$$

The $h = 0$ term of eq. 54 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 were 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.

$$R_k(b, l, m, n) \geq r_k(b, l, m, n, 0) - lr_k(b, l, m, n, 1) \tag{57}$$

$$\geq [1 - P(m) - (n - l)Q(m)]^b - l \sum_{j < k} \binom{b}{j} [Q(m)]^j [1 - P(m) - (n - l + 1)Q(m)]^{b - j}. \tag{58}$$

By eq. 14 we have

$$F_l \geq \sum_{j_0 < k} \binom{b}{j_0} [P(l)]^{j_0} \left([1 - P(l)]^{b - j_0} - l \sum_{j < k} \binom{b - j_0}{j} [Q(l)]^j [1 - P(l) - Q(l)]^{b - j_0 - j} \right). \tag{59}$$

A lower bound on the sum that comes from the first term in the large parentheses is given by eq. 47. The reasoning that leads to eq. 25 gives the following bound for the sum coming from the second term

$$\sum_{j_0 < k} \binom{b}{j_0} [P(l)]^{j_0} \sum_{j < k} \binom{b - j_0}{j} [Q(l)]^j [1 - P(l) - Q(l)]^{b - j_0 - j} \leq x^{-k} y^{-k+1} [1 + P(l)(xy - 1) + (x - 1)Q(l)]^b. \tag{60}$$

By setting $y = 1$ and using eq. 8 the bound becomes $x^{-k} [1 + (x - 1)P(l - 1)]^b$, which is the upper bound on S_{l-1} (eq. 21). Thus, combining this with eqs. 47 and 45, we obtain

$$F_l \geq 1 - e^{-b\alpha_1^2 / \{2P(l)[1 - P(l)]\} + O(b\alpha_1^3 [1 - P(l)]^{-2})} - le^{-b\alpha_4^2 / \{2P(l-1)[1 - P(l-1)]\} + O(b\alpha_4^3 [1 - P(l-1)]^{-2})} \tag{61}$$

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.,

$$k = \frac{bP(l) + bP(l - 1) - 1}{2} + \eta \tag{62}$$

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

$$\alpha_1 = \frac{bP(l - 1) - bP(l) - 1}{2} + \eta, \quad \alpha_4 = \frac{bP(l - 1) - bP(l) - 1}{2} - \eta. \tag{63}$$

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. 61 (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. 61 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.)

5. Total work

Eq. 61 shows that for random data there are many cases where the Apriori Algorithm has one bad level, i.e., a level where the pretest is not effective even though few item sets pass the final test. Eq. 34 shows that there are other cases for random data where the Apriori Algorithm has no bad levels. The Apriori Algorithm has a reputation for being effective in practice [1, 3, 12]. 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. 3 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. 3:

$$\binom{|I|}{l+1} / \binom{|I|}{l} \frac{|I|-l}{l} \approx \frac{|I|}{l}, \quad (64)$$

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. 51 results in α_3 near 0 (from eq. 52) and F_l near 1. Now consider eq. 51 with l one larger. For this l , we have $\alpha = P(l-1) - P(l)$ and eq. 51 gives the bound

$$F_l \leq e^{-b(l+1)[P(l-1)-P(l)]^2\theta/(2\{P(l)+lP(l+1)-(l+1)[P(l)]^2\})}. \quad (65)$$

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

$$e^{-b(l+1)p^{l-2}/2} \approx e^{-(l+1)k/(2p)}. \quad (66)$$

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

$$\frac{|I|}{l} e^{-(l+1)k/(2p)}. \quad (67)$$

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.

6. 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 where 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. 45 and 47 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. 51, 53, and 61. 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 (upto 100,000 in [1]) with the need to keep the computing time reasonable. Also, 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. 45. The next rightmost is the actual value from eq. 9. The least rightmost in the group of three is the lower bound from eq. 47. 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. 14, 15, and 17. 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 from eqs. 51, 53, and 61. 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.

7. Discussion

Most people using the Apriori Algorithm are probably interested in applying it to data generated by nonrandom shoppers. It is a question for experiments to determine how closely our formulas for random data capture the basic properties of the algorithm when run on such data. 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. We expect that real data will be similar except that it will have subsets with various probabilities. If so, the transition to small work will be smeared out, but otherwise things will be qualitatively similar.

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

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 .

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

$$S_l = \sum_{j \geq k} \binom{b}{j} [P(l)]^j [1 - P(l)]^{b-j}. \quad (9a)$$

From the binomial theorem we have

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

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 .

so

$$\sum_{j \geq k} \binom{b}{j} [P(l)]^j [1 - P(l)]^{b-j} = 1 - \sum_{j < k} \binom{b}{j} [P(l)]^j [1 - P(l)]^{b-j}. \quad (9b)$$

Eq. 10. 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 , \dots , $[Q(l)]^{j_l}$ is the probability that the next j_l baskets obey condition M_l , and $[1 - P(l) - lQ(l)]^{b-j_0-\dots-j_l}$ is the probability that the remaining baskets obey none of the conditions M_0, \dots, M_l . (Notice that each basket obeys at most one of the conditions M_h ($0 \leq h \leq l$.) The multinomial coefficient counts the number of other sequences that lead to the same result (they all have the same probability). Thus, we need j_0 baskets that obey condition M_0 , \dots , j_l baskets that obey condition M_j , but they don't need to be the first j_0 baskets for condition M_0, \dots , and the next j_l baskets for condition M_l . 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 $b - j_0 - \dots - j_l$ baskets do not obey any of the conditions is

$$\binom{b}{j_0, \dots, j_l, b - j_0 - \dots - j_l} [P(l)]^{j_0} [Q(l)]^{j_1 + \dots + j_l} [1 - P(l) - lQ(l)]^{b - j_0 - \dots - j_l}. \quad (10)$$

Eq. 16.

$$R_k(b, l, m, n) = \sum_{\substack{j_1 \geq k \\ j_2 \geq k \\ \vdots \\ j_l \geq k}} \binom{b}{j_1, \dots, j_l, b - j_1 - \dots - j_l} [Q(m)]^{j_1 + \dots + j_l} [1 - P(m) - nQ(m)]^{b - j_1 - \dots - j_l} \quad (15)$$

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}	$1.0 - 3.4 \times 10^{-2}$
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}	$1.0 - 1.0 \times 10^{-3}$	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}	$1.0 - 3.4 \times 10^{-7}$	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 .

$$\begin{aligned}
&= \sum_{j_1} \binom{b}{j_1} [Q(m)]^{j_1} \\
&\quad \times \sum_{\substack{j_2 \geq k \\ j_3 \geq k \\ \dots \\ j_l \geq k}} \binom{b-j_1}{j_2, \dots, j_l, b-j_1-j_2-\dots-j_l} [Q(m)]^{j_2+\dots+j_l} [1-P(m)-nQ(m)]^{(b-j_1)-j_2-\dots-j_l}
\end{aligned} \tag{A2}$$

$$= \sum_{j \geq k} \binom{b}{j} [Q(m)]^j R_k(b-j, l-1, m, n). \tag{16}$$

Eq. 17. The boundary condition is just eq. 15 with l replaced with 0.

Eq. 25. Using the binomial theorem on the j_l sum in eq. 23, we have

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

k	F_1	F_2	F_3	F_4	F_5
1	2.0×10^{-29}	1.8×10^{-2}	7.3×10^{-1}	1.9×10^{-3}	6.4×10^{-10}
2	1.4×10^{-27}	9.1×10^{-2}	7.3×10^{-1}	2.9×10^{-5}	6.3×10^{-13}
3	4.8×10^{-26}	2.4×10^{-1}	4.5×10^{-1}	2.3×10^{-7}	3.0×10^{-16}
4	1.1×10^{-24}	4.3×10^{-1}	2.0×10^{-1}	1.2×10^{-9}	9.8×10^{-20}
5	1.9×10^{-23}	6.3×10^{-1}	6.3×10^{-2}	4.7×10^{-12}	2.4×10^{-23}
22	1.5×10^{-10}	$1.0 - 6.0 \times 10^{-10}$	7.1×10^{-23}	3.9×10^{-61}	5.6×10^{-93}
32	2.0×10^{-6}	$1.0 - 4.1 \times 10^{-6}$	1.6×10^{-40}	1.7×10^{-93}	2.0×10^{-137}
33	4.3×10^{-6}	$1.0 - 8.7 \times 10^{-6}$	2.1×10^{-42}	8.1×10^{-97}	6.0×10^{-142}
45	4.2×10^{-3}	$1.0 - 8.4 \times 10^{-3}$	2.1×10^{-68}	1.1×10^{-137}	2.7×10^{-197}
57	1.7×10^{-1}	6.9×10^{-1}	2.4×10^{-92}	5.1×10^{-180}	1.2×10^{-274}
58	2.0×10^{-1}	6.4×10^{-1}	1.4×10^{-94}	1.3×10^{-183}	1.7×10^{-287}
64	4.8×10^{-1}	2.7×10^{-1}	3.4×10^{-108}	2.8×10^{-205}	
65	5.3×10^{-1}	2.2×10^{-1}	1.7×10^{-110}	6.4×10^{-209}	
91	$1.0 - 5.8 \times 10^{-4}$	3.3×10^{-7}	3.9×10^{-173}		
120	$1.0 - 5.3 \times 10^{-11}$	2.8×10^{-21}	4.0×10^{-250}		
121	$1.0 - 2.6 \times 10^{-11}$	6.7×10^{-22}	3.5×10^{-253}		
128	$1.0 - 1.4 \times 10^{-13}$	2.1×10^{-26}	9.7×10^{-276}		
129	$1.0 - 6.3 \times 10^{-14}$	4.4×10^{-27}	4.1×10^{-279}		
186	1.0	7.0×10^{-77}			
247	1.0	1.3×10^{-150}			
248	1.0	5.8×10^{-152}			
256	1.0	5.1×10^{-163}			
257	1.0	2.0×10^{-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 $b = 1024$, $p = 1/16$.

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

$$\begin{aligned}
R_k(b, l, m, n) &\leq x^{-kl} \sum_{j_1, \dots, j_l} \binom{b}{j_1, \dots, j_l, b - j_1 - \dots - j_l} [xQ(m)]^{j_1 + \dots + j_l} [1 - P(m) - nQ(m)]^{b - j_1 - \dots - j_l} \\
&\leq x^{-kl} [1 - P(m) - (n - lx)Q(m)]^b.
\end{aligned} \tag{23}$$

Eq. 26. Less algebra is needed to minimize the logarithm of the bound, and it leads to the same result.

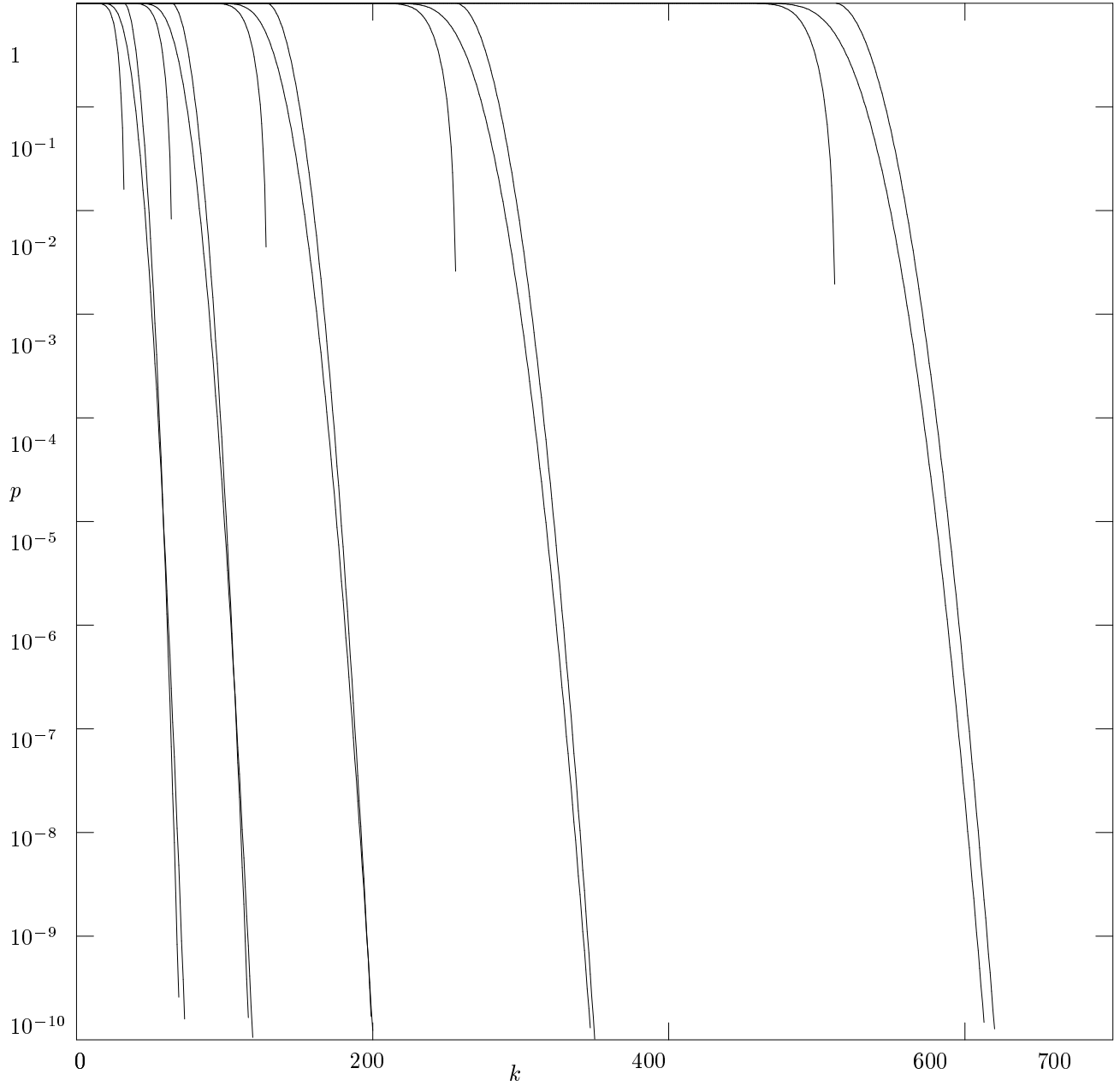


Figure 1. The value of S_l and approximations to S_l for $p = 1/2$ and $1 \leq l \leq 5$. The rightmost curve is the upper bound on S_1 (eq. 45 with the big O term omitted). The next rightmost curve is the actual value of S_1 . The next rightmost curve is the lower bound on S_1 (eq. 47 with the big O 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 .

Start with the derivative of the logarithm of eq. 21.

$$\frac{d\{-k \ln x + b \ln[1 + (x-1)P(l)]\}}{dx} = \frac{-k}{x} + \frac{bP(l)}{1 + P(l)(x-1)} = 0, \quad (\text{A4})$$

$$-kP(l)(x-1) - k + bP(l)x = 0. \quad (\text{A5})$$

$$x = \frac{k[1 - P(l)]}{(b - k)P(l)}. \quad (\text{26})$$

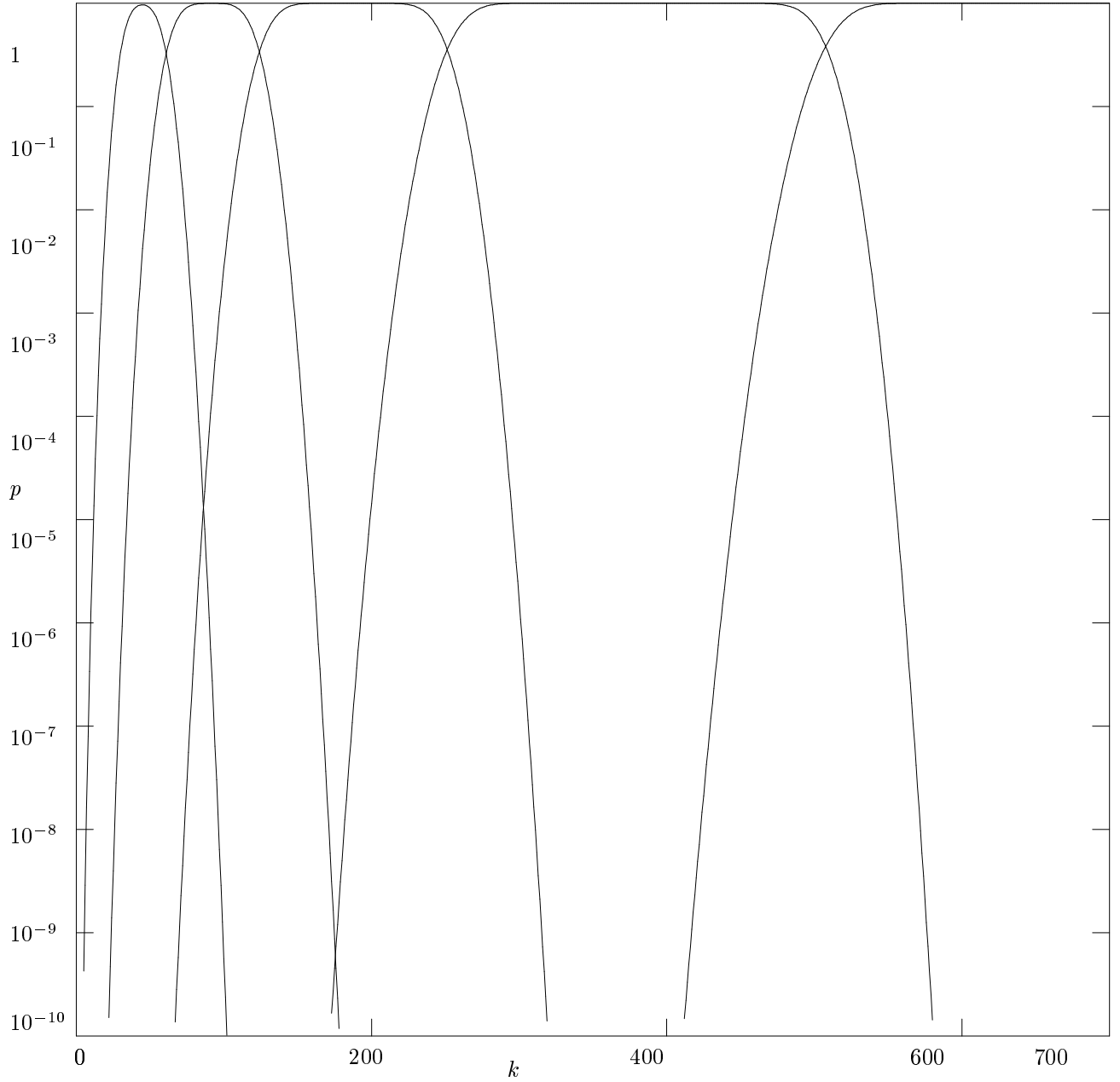


Figure 2. The value of F_l for $p = 1/2$ and $1 \leq l \leq 5$. The left most hump is the curve for $l = 5$, the next leftmost hump is for $l = 4$, etc.

Eq. 27.

$$\frac{k[1 - P(l)]}{(b - k)P(l)} > 1, \quad (\text{A6})$$

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

(since $b > k$)

$$k > bP(l). \quad (\text{27})$$

Eq. 28. The derivative of the logarithm of the bound on F (eq. 25) with respect x is

$$\frac{d[-kl \ln x - (k - 1) \ln y + b \ln[1 + (x^l y - 1)P(l) + l(x - 1)Q(l)]]}{dx}$$

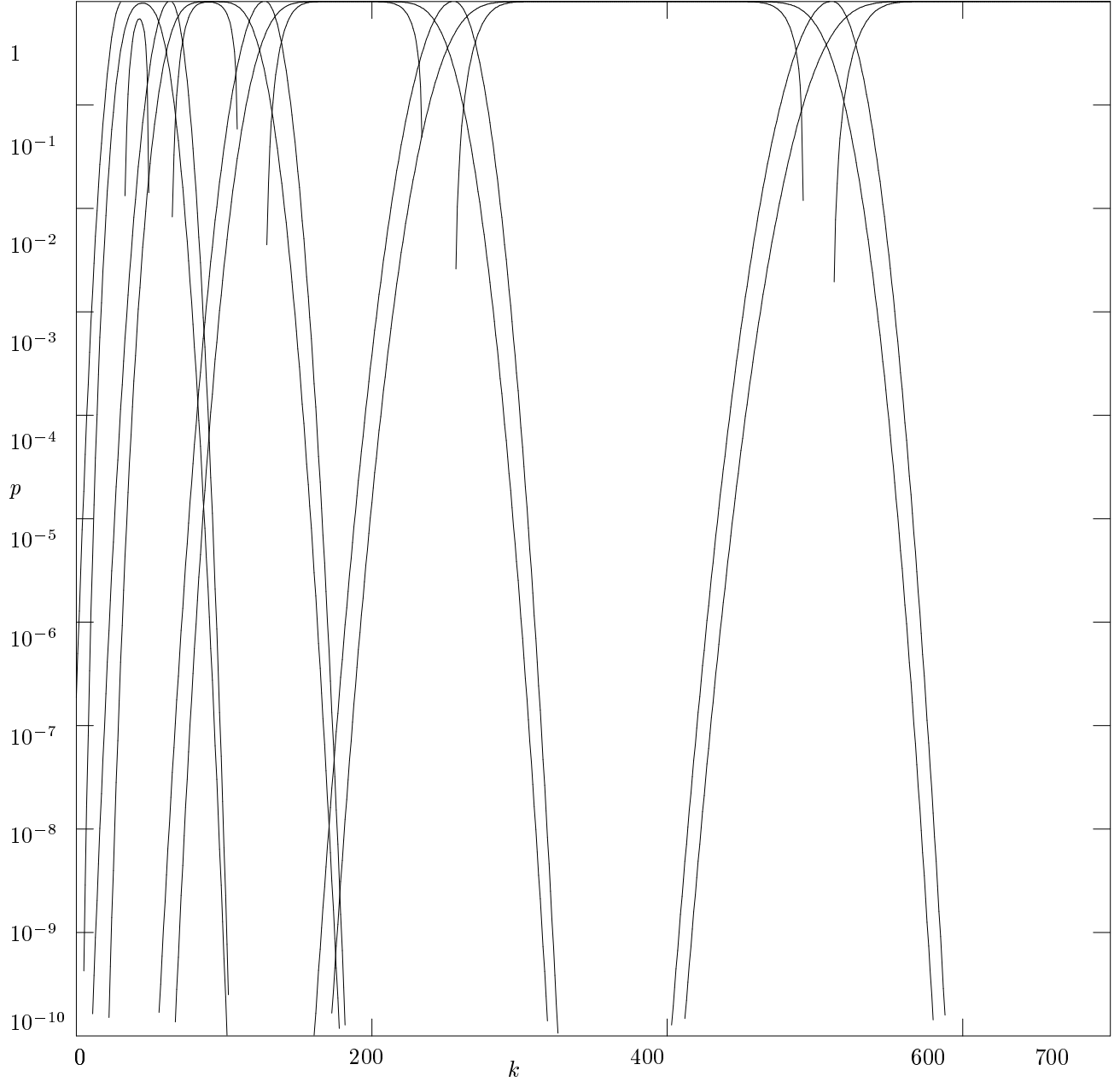


Figure 3. The value of F_l for $p = 1/2$ and $1 \leq l \leq 5$ along with upper and lower bound.

$$= \frac{-kl}{x} + \frac{b[lx^{l-1}yP(l) + lQ(l)]}{1 + (x^l y - 1)P(l) + l(x - 1)Q(l)}. \quad (\text{A8})$$

Setting this to zero gives

$$-kl[1 + (x^l y - 1)P(l) + l(x - 1)Q(l)] + bx[lx^{l-1}yP(l) + lQ(l)] = 0, \quad (\text{A9})$$

$$-kl - klP(l)x^l y + klP(l) + kl^2Q(l) - kl^2Q(l)x + blP(l)x^l y + blQ(l)x = 0, \quad (\text{A10})$$

$$(b - k)P(l)x^l y + (b - kl)Q(l)x - k[1 - P(l) - lQ(l)] = 0. \quad (\text{28})$$

Eq. 29. The derivative of the logarithm bound (eq. 25) with respect to y gives

$$\frac{d[-kl \ln x - (k - 1) \ln y + b \ln[1 + (x^l y - 1)P(l) + l(x - 1)Q(l)]}{dy} \quad (\text{A11})$$

$$= -\frac{k-1}{y} + \frac{bx^l P(l)}{1 + (x^l y - 1)P(l) + l(x-1)Q(l)}. \quad (\text{A12})$$

Setting this to zero gives

$$-(k-1)[1 + (x^l y - 1)P(l) + l(x-1)Q(l)] + bx^l y P(l) = 0, \quad (\text{A13})$$

$$(b-k+1)P(l)x^l y - (k-1)[1 - P(l) + lQ(l)(x-1)] = 0. \quad (29)$$

Eq. 30. Eq. 28 with $x = 1$, $y = 1$ is

$$(b-k)P(l) + (b-kl)Q(l) - k[1 - P(l) - lQ(l)] = 0, \quad (\text{A14})$$

$$k = b[P(l) + Q(l)] = bP(l-1). \quad (30)$$

Eq. 31. By implicit differentiation of eq. 28 (with $y = 1$) we have

$$\frac{d\{(b-k)P(l)x^l + (b-kl)Q(l)x - k[1 - P(l) - lQ(l)]\}}{db} = 0, \quad (\text{A15})$$

$$P(l)x^l + Q(l)x + [(b-k)lP(l)x^{l-1} + (b-kl)Q(l)]\frac{dx}{db} = 0, \quad (\text{A16})$$

$$\frac{dx}{db} = -\frac{P(l)x^l + Q(l)x}{(b-k)lP(l)x^{l-1} + (b-kl)Q(l)} \quad (\text{A17})$$

When x solves eq. 28 (with $y = 1$) we have

$$(b-k)P(l)x^l + (b-kl)Q(l)x = k[1 - P(l) - lQ(l)], \quad (\text{A18})$$

which is positive. For such an x

$$(b-k)P(l)lx^{l-1} + (b-kl)Q(l) = \frac{(b-k)P(l)x^l + (b-kl)Q(l)x}{x} + (l-1)(b-k)P(l)x^{l-1} \quad (\text{A19})$$

is also positive (because $x > 0$, $b-k > 0$, and $l \geq 1$). Thus, dx/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. 32. Eq. 29 with $x = 1$ is

$$(b-k+1)P(l)y - (k-1)[1 - P(l)] = 0. \quad (\text{A20})$$

$$y = \frac{(k-1)[1 - P(l)]}{(b-k+1)P(l)}. \quad (32)$$

Eq. 33.

$$\frac{(k-1)[1 - P(l)]}{(b-k+1)P(l)} < 1, \quad (\text{A21})$$

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

$$k-1 < bP(l). \quad (\text{A23})$$

$$k < bP(l) + 1. \quad (33)$$

Eq. 35. From eq. 28

$$P(l)x^l y = \frac{k[1 - P(l) - lQ(l)] - (b-kl)Q(l)x}{b-k}. \quad (\text{A24})$$

From eq. 29

$$P(l)x^l y = \frac{(k-1)[1-P(l)+l(x-1)Q(l)]}{b-k+1}. \quad (\text{A25})$$

Setting the two right sides equals and clearing fractions gives

$$k(b-k+1)[1-P(l)-lQ(l)] - (b-kl)(b-k+1)Q(l)x = (k-1)(b-k)[1-P(l)+lQ(l)x-lQ(l)]. \quad (\text{A26})$$

$$(b^2-bk+b-bkl+k^2l-kl)Q(l)x = (bk-k^2+k)[1-P(l)-lQ(l)] - (bk-k^2-b+k)[1-P(l)+lQ(l)x-lQ(l)]. \quad (\text{A27})$$

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

$$(b^2-bk+b-bl)Q(l)x = b[1-P(l)-lQ(l)]. \quad (\text{A29})$$

$$x = \frac{1-P(l)-lQ(l)}{(b-k-l+1)Q(l)}. \quad (\text{35})$$

Eq. 36. Plugging the value of x from eq. 35 into eq. 28 gives

$$(b-k) \left(\frac{1-P(l)-lQ(l)}{(b-k-l+1)Q(l)} \right)^l P(l)y + \frac{(b-kl)[1-P(l)-lQ(l)]}{b-k-l+1} - k[1-P(l)-lQ(l)] = 0, \quad (\text{A30})$$

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

$$y = \frac{(b-k-l+1)^{l-1}Q(l)^l \{k(b-k-l+1)[1-P(l)-lQ(l)] - (b-kl)[1-P(l)-lQ(l)]\}}{(b-k)[1-P(l)-lQ(l)]^l P(l)}. \quad (\text{A32})$$

$$y = \frac{(b-k-l+1)^{l-1}Q(l)^l (bk-k^2-kl+k-b+kl)[1-P(l)-lQ(l)]}{(b-k)[1-P(l)-lQ(l)]^l P(l)}, \quad (\text{A33})$$

$$y = \frac{(b-k-l+1)^{l-1}Q(l)^l (k-1)[1-P(l)-lQ(l)]}{[1-P(l)-lQ(l)]^l P(l)}, \quad (\text{A34})$$

$$y = (k-1) \left(\frac{(b-k-l+1)Q(l)}{1-P(l)-lQ(l)} \right)^{l-1} \frac{Q(l)}{P(l)}. \quad (\text{36})$$

Eq. 37. To have $x > 1$ we need

$$\frac{1-P(l)-lQ(l)}{(b-k-l+1)Q(l)} > 1. \quad (\text{A35})$$

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

$$1-P(l)-lQ(l) > (b-k-l+1)Q(l), \quad (\text{A36})$$

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

$$b < k-1 + \frac{1-P(l)}{Q(l)}. \quad (\text{A38})$$

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

$$k+l-1 < b < k+l-1 + \frac{1-P(l)-lQ(l)}{Q(l)}. \quad (\text{37})$$

Suppose $b < k+l-1$. Then from eq. A35 we have

$$1-P(l)-lQ(l) < (b-k-l+1)Q(l), \quad (\text{A39})$$

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

$$b > k - 1 + \frac{1 - P(l)}{Q(l)}, \quad (\text{A41})$$

$$k + l - 1 > b > k - 1 + \frac{1 - P(l)}{Q(l)}. \quad (\text{A42})$$

For this range to be non-empty, we need

$$k + l - 1 > k - 1 + \frac{1 - P(l)}{Q(l)}, \quad (\text{A43})$$

$$0 > \frac{1 - P(l)}{Q(l)} - l, \quad (\text{A44})$$

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

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 $0 \leq p \leq 1$) the sum of some of the terms is no more than 1, so the right side of eq. A45 is nonnegative. Thus, the range is always empty.

Eq. 38.

$$b < bP(l-1) + l - 1 + \frac{1 - P(l) - lQ(l)}{Q(l)}. \quad (\text{A46})$$

$$b[1 - P(l-1)] < \frac{1 - P(l) - lQ(l)}{Q(l)}. \quad (\text{A47})$$

$$b[1 - P(l-1)] < \frac{1 - P(l-1)}{Q(l)}. \quad (\text{A48})$$

$$b < \frac{1}{Q(l)}. \quad (\text{A49})$$

Eq. 39. To have $y < 1$ we need

$$(k-1) \left(\frac{(b-k-l+1)Q(l)}{1-P(l)-lQ(l)} \right)^{l-1} \frac{Q(l)}{P(l)} < 1, \quad (\text{A50})$$

For $l \geq 2$

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

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

$$b < k + l - 1 + \frac{1 - P(l) - lQ(l)}{Q(l)} \left(\frac{P(l)}{(k-1)Q(l)} \right)^{1/(l-1)}. \quad (\text{39})$$

The upper bound on b from eq. 39 is greater than the lower bound from eq. 37.

$$k + l - 1 + \frac{1 - P(l) - lQ(l)}{Q(l)} \left(\frac{P(l)}{(k-1)Q(l)} \right)^{1/(l-1)} > k + l - 1, \quad (\text{A53})$$

$$\frac{1 - P(l) - lQ(l)}{Q(l)} \left(\frac{P(l)}{(k-1)Q(l)} \right)^{1/(l-1)} > 0. \quad (\text{A54})$$

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

Eq. 40. We now consider when the upper bound on b from eq. 39 is less than the upper bound from eq. 37.

$$k + l - 1 + \frac{1 - P(l) - lQ(l)}{Q(l)} \left(\frac{P(l)}{(k-1)Q(l)} \right)^{1/(l-1)} < k + l - 1 + \frac{1 - P(l) - lQ(l)}{Q(l)}, \quad (\text{A55})$$

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

$$0 < \frac{1 - P(l) - lQ(l)}{Q(l)} \left[1 - \left(\frac{P(l)}{(k-1)Q(l)} \right)^{1/(l-1)} \right]. \quad (\text{A57})$$

As shown above (below eq. A45), the first factor is always positive. For the second factor to be positive we need

$$1 > \left(\frac{P(l)}{(k-1)Q(l)} \right)^{1/(l-1)}, \quad (\text{A58})$$

$$1 > \frac{P(l)}{(k-1)Q(l)}, \quad (\text{A59})$$

$$(k-1)(1-p)p^{l-1} > p^l, \quad (\text{A60})$$

$$(k-1)(1-p) > p, \quad (\text{A61})$$

$$k - kp - 1 + p > p, \quad (\text{A62})$$

$$k(1-p) > 1, \quad (\text{A63})$$

$$k > \frac{1}{1-p}. \quad (40)$$

Eq. 43. Plugging eq. 26 into eq. 21 gives

$$S_l \leq \left(\frac{k[1 - P(l)]}{(b-k)P(l)} \right)^{-k} \left[1 + \left(\frac{k[1 - P(l)]}{(b-k)P(l)} - 1 \right) P(l) \right]^b, \quad (\text{A64})$$

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

$$S_l \leq \left(\frac{P(l)}{k} \right)^k \left(\frac{1 - P(l)}{b-k} \right)^{b-k} b^b. \quad (43)$$

Eq. 45. Replace k in eq. 43 with its value in terms of α_1 (eq. 44)

$$S_l \leq \left(\frac{P(l)}{b[P(l) + \alpha_1]} \right)^{b[P(l) + \alpha_1]} \left(\frac{1 - P(l)}{b - b[P(l) + \alpha_1]} \right)^{b - b[P(l) + \alpha_1]} b^b, \quad (\text{A65})$$

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

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

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

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

$$X = \ln \left[\left(\frac{1}{1 + \alpha_1/P(l)} \right)^{b[P(l) + \alpha_1]} \left(\frac{1}{1 - \alpha_1/[1 - P(l)]} \right)^{b[1 - P(l) - \alpha_1]} \right] \quad (\text{A69})$$

$$= -b[P(l) + \alpha_1] \ln \left(1 + \frac{\alpha_1}{P(l)} \right) - b[1 - P(l) - \alpha_1] \ln \left(1 - \frac{\alpha_1}{[1 - P(l)]} \right). \quad (\text{A70})$$

Dividing by b , we have

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

$$= -[P(l) + \alpha_1] \left[\left(\frac{\alpha_1}{P(l)} \right) - \frac{1}{2} \left(\frac{\alpha_1}{P(l)} \right)^2 + O \left(\left(\frac{\alpha_1}{P(l)} \right)^3 \right) \right] \quad (\text{A72})$$

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

$$= -\alpha_1 + \frac{\alpha_1^2}{2P(l)} - O \left(\frac{\alpha_1^3}{P(l)^2} \right) - \frac{\alpha_1^2}{P(l)} + \frac{\alpha_1^3}{2P(l)^2} - O \left(\frac{\alpha_1^4}{P(l)^3} \right) \\ + \alpha_1 + \frac{\alpha_1^2}{2[1 - P(l)]} + O \left(\frac{\alpha_1^3}{[1 - P(l)]^2} \right) - \frac{\alpha_1^2}{1 - P(l)} - \frac{\alpha_1^3}{2[1 - P(l)]^2} - O \left(\frac{\alpha_1^4}{[1 - P(l)]^3} \right) \quad (\text{A74})$$

$$= -\frac{\alpha_1^2}{2P(l)} - \frac{\alpha_1^2}{2[1 - P(l)]} + O \left(\frac{\alpha_1^3}{[1 - P(l)]^2} \right) - O \left(\frac{\alpha_1^3}{P(l)^2} \right), \quad (\text{A75})$$

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,

$$S_l \leq e^{-b\alpha_1^2/\{2P(l)[1-P(l)]\} + O(b\alpha_1^3[1-P(l)]^{-2})}. \quad (\text{A5})$$

Eq. 49. Eq. 28 with $y = 1$ is

$$(b - k)P(l)x^l + (b - kl)Q(l)x - k[1 - P(l) - lQ(l)] = 0. \quad (\text{A76})$$

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.

$$(b - k)P(l) \left(1 + l\delta + \frac{l(l-1)\delta^2\theta}{2} \right) + (b - kl)Q(l)(1 + \delta) = k[1 - P(l) - lQ(l)]. \quad (\text{A77})$$

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

$$\delta = \frac{k[1 - P(l) - lQ(l)] - (b - k)P(l) - (b - kl)Q(l)}{l(b - k)P(l)[1 + (l-1)\theta\delta/2] + (b - kl)Q(l)}, \quad (\text{A79})$$

$$\delta = \frac{k - bP(l) - bQ(l)}{b[lP(l) + Q(l)] - kl[P(l) + Q(l)] + (b - k)l(l-1)P(l)\theta\delta/2}, \quad (\text{A80})$$

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

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

$$\delta = \frac{k - bP(l - 1)}{b[lP(l) + Q(l)] - klP(l - 1)} \left(1 + \frac{[k - bP(l - 1)](b - k)l(l-1)P(l)\theta/2}{\{b[lP(l) + Q(l)] - klP(l - 1)\}^2} \right)^{-1}. \quad (\text{A9})$$

Eq. 51. Write eq. 48 as

$$F_l \leq e^X \quad (\text{A83})$$

with

$$X = \ln\{x^{-kl}[1 + (x^l - 1)P(l) + l(x - 1)Q(l)]^b\} \quad (\text{A84})$$

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

Replace x with $1 + \delta$.

$$X = -kl \ln(1 + \delta) + b \ln\{1 + [(1 + \delta)^l - 1]P(l) + lQ(l)\delta\} \quad (\text{A86})$$

Expanding X in a power series to second order gives

$$\begin{aligned} X &= -kl \ln(1 + \delta) + b \ln\left(1 + l\delta P(l) + \frac{l(l-1)P(l)\theta\delta^2}{2} + lQ(l)\delta\right) \\ &= -kl\delta + \frac{kl\theta\delta^2}{2} + b\left(lP(l)\delta + \frac{l(l-1)\delta^2 P(l)\theta}{2} + lQ(l)\delta\right) \\ &\quad - \frac{b}{2}\left(lP(l)\delta + \frac{l(l-1)P(l)\theta\delta^2}{2} + lQ(l)\delta\right)^2 \theta \end{aligned} \quad (\text{A87})$$

$$= -l[k - bP(l) - bQ(l)]\delta + \frac{kl + bl(l-1)P(l) - bl^2[P(l) + Q(l)]^2}{2}\theta\delta^2 \quad (\text{A88})$$

$$= -l[k - bP(l-1)]\delta + \frac{kl + bl(l-1)P(l) - bl^2[P(l-1)]^2}{2}\theta\delta^2. \quad (\text{A89})$$

Replace k by its definition in terms of α_3 (eq. 50) to obtain

$$X = -l[k - bP(l-1)]\delta + \frac{\{kl + bl(l-1)P(l) - bl^2[P(l-1)]^2\}\theta}{2}\delta^2 \quad (\text{A90})$$

$$= -bl\alpha_3\delta + \frac{bl\{P(l-1) + \alpha_3 + (l-1)P(l) - l[P(l-1)]^2\}\theta}{2}\delta^2 \quad (\text{A91})$$

$$= \left(-bl\alpha_3 + \frac{bl\{P(l-1) + \alpha_3 + (l-1)P(l) - l[P(l-1)]^2\}\theta}{2}\right)\delta. \quad (\text{A92})$$

Also replace k in eq. 49 by its value in terms of α_3 to obtain

$$\begin{aligned} \delta &= \frac{\alpha_3 b}{b[lP(l) + Q(l)] - b[P(l-1) + \alpha_3]lP(l-1)} \\ &\quad \times \left(1 + \frac{b\alpha_3\{b - b[P(l-1) + \alpha_3]\}l(l-1)P(l)\theta/2}{\{b[lP(l) + Q(l)] - b[P(l-1) + \alpha_3]l[P(l-1)]\}^2}\right)^{-1} \end{aligned} \quad (\text{A93})$$

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

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

$$\delta = \frac{\theta\alpha_3}{lP(l) + Q(l) - l[P(l-1)]^2}. \quad (\text{A95})$$

Plugging the value of δ into the expression for X (eq. A92) gives

$$X = \left(-bl\alpha_3 + \frac{bl\{P(l-1) + \alpha_3 + (l-1)P(l) - l[P(l-1)]^2\}\theta}{2}\right)\delta. \quad (\text{A92})$$

$$= \left(-bl\alpha_3 + \frac{bl\{P(l-1) + (l-1)P(l) - l[P(l-1)]^2 + \alpha_3\}\theta}{2\{P(l-1) + (l-1)P(l) - l[P(l-1)]^2\}}\right) \frac{\theta\alpha_3}{P(l-1) + (l-1)P(l) - l[P(l-1)]^2} \quad (\text{A96})$$

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

$$= \frac{bl\theta\alpha_3^2}{2\{P(l-1) + (l-1)P(l) - l[P(l-1)]^2\}}. \quad (\text{A98})$$

Thus,

$$F_l \leq e^{-bl\theta\alpha_3^2/(2\{P(l-1)+(l-1)P(l)-l[P(l-1)]^2\})}. \quad (51)$$

Eq. 52. The derivation of eq. 51 requires that $\delta = o(1)$. The step from eq. A97 to eq. A98 requires that α_3 be small compared to some other terms. Both conditions imply

$$\alpha_3 = \{lP(l) + Q(l) - l[P(l-1)]^2\}o(1). \quad (52)$$

Eq. 54. 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. 56.

$$r_k(b, l, m, n, h) = \sum_{\substack{j_1 < k \\ j_2 < k \\ \dots \\ j_h < k \\ j_{h+1}, \dots, j_l}} \binom{b-j_1}{j_2, \dots, j_l, b-j_1-\dots-j_l} [Q(m)]^{j_1+\dots+j_l} [1-P(m)-nQ(m)]^{b-j_1-\dots-j_l} \quad (55)$$

$$= \sum_{\substack{j_1 < k \\ j_2 < k \\ \dots \\ j_h < k \\ j_{h+1}, \dots, j_{l-1}}} \binom{b-j_1}{j_2, \dots, j_{l-1}, b-j_1-\dots-j_{l-1}} [Q(m)]^{j_1+\dots+j_{l-1}} [1-P(m)-(n-1)Q(m)]^{b-j_1-\dots-j_{l-1}} \quad (A99)$$

$$\dots$$

$$= \sum_{\substack{j_1 < k \\ j_2 < k \\ \dots \\ j_h < k}} \binom{b-j_1}{j_2, \dots, j_h, b-j_1-\dots-j_h} [Q(m)]^{j_1+\dots+j_h} [1-P(m)-(n-l+h)Q(m)]^{b-j_1-\dots-j_h}. \quad (56)$$

Bound on first term of eq. 59. 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. 21.

Eq. 60.

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

$$\leq \sum_{j_0, j} \binom{b}{j_0} [P(l)]^{j_0} \binom{b-j_0}{j} [Q(l)]^j [1-P(l)-Q(l)]^{b-j_0-j} x^j y^{j_0} \quad (A100)$$

$$\leq x^{-kl} y^{-k+1} [1+P(l)(x^l y - 1) + (x-1)Q(l)]^b. \quad (A101)$$

References

1. 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.

2. R. Agrawal, R. Srikant, *Fast algorithms for mining association rules*, in Proceedings of the 1994 Very Large Data Bases Conference, (1994) pp 487–499.
3. 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.
4. 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.
5. 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.
- 7 Laks V.S. Lakshmanan, Raymond 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.
6. Michael Karr, *Summations in Finite Terms*, J. ACM **28** (1981) pp 305–350.
8. H. Mannila, *Methods and problems in data mining*, in Proceedings of the International Conference on Database Theory, pp 41–55, 1997.
9. 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.
10. 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.
11. Paul Walton Purdom, Jr. and Cynthia A. Brown, *The Analysis of Algorithms*, Holt, Rinehart and Winston, New York (1985).
12. 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.
13. Hannu Toivonen, *Discovery of Frequent Patterns in Large Data Collections*, PhD Thesis, Report A-1996-5, University of Helsinki, Department of Computer Science, November 1996.
14. 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.