# Growth Projections and Assortment Planning of Commodity Products across Multiple Stores: A Data Mining and Optimization Approach

Xue Bai
Sudip Bhattacharjee
Fidan Boylu
Ram Gopal

[1]Department of Operations and Information Management
2100 Hillside Road, U-1041
School of Business
University of Connecticut
Storrs, CT 06269-1041

{xue.bai, sbhattacharjee, fidan.boylu, ram.gopal}@business.uconn.edu

# Online Supplement

## Online Supplement A. Conflict resolution and pruning algorithm: estimating number of iterations

Our focus here is to determine an upper bound of the number of iterations needed to prune the explosion of ARM itemsets.

**Step 1:**

An equivalent formulation for the conflict resolution and pruning algorithm is as follows: Choose the itemset that has the highest *lift*. A tie can be broken using *support* values, and any rare occurrence of a subsequent tie can be broken randomly. For the itemset chosen, identify all transactions that contain the frequent itemset. Drop those frequent itemsets from those transactions, keeping any other products present in those transactions. Choose the next itemset and continue the process. Stop when there are no itemsets which match the itemsets of remaining transaction elements.

Intuitively, assume that transactions are initially stacked in bins numbered with the number of products in each transaction, e.g. bin 1, 2, and 3 contain 1, 2, and 3 products per transaction respectively. If a two-product itemset is the dominant itemset, then transactions that match 2 products will subsequently move from bin 3 to bin 1 (one product remaining), or bin 2 to out. We model the process below.

Let P: total number of products sold, and $p \in \{1, \ldots, P\}$

Let products be combined in all possible manner, so:

P: possible number of 1 product combinations

${}^{P}C_2$: possible number of 2 product combinations

${}^{P}C_3$: possible number of 3 product combinations

…

${}^{P}C_P$: possible number of P product combinations

Let there be $w_1$ single product transactions, $w_2$ two-product transactions, etc.

This is a multinomial distribution in general. The probability mass function for this is given by:

$$\frac{(w_1 + w_2 + \cdots + w_p)!}{w_1!\, w_2! \ldots w_P!} {z_1}^{w_1} {z_2}^{w_2} \ldots {z_P}^{w_P}$$

where

$z_1$ = probability of finding a one product transaction

$z_2$ = probability of finding a two product transaction, etc.

From above,

$z_1 = P/(P + {}^{P}C_2 + {}^{P}C_3 + \ldots + {}^{P}C_P)$

$z_2 = {}^{P}C_2 /( P + {}^{P}C_2 + {}^{P}C_3 + \ldots + {}^{P}C_P)$, etc.

Worst case scenario: Since our focus is on finding an upper bound for the number of iterations, we choose the **worst case** possible: a two-product itemset, which is the smallest possible ARM itemset and which would arguably take the longest time to converge.

Now let us take each bin – 1, 2, 3, …, P and try to find the probability of finding a two-product combination (say $p_1p_2$) in each:

Bin 1: Prob($p_1p_2$) = 0

Bin 2: Prob($p_1p_2$) = $1/{}^{P}C_2$

Bin 3: Prob($p_1p_2$) = $(P\text{-}2)/{}^{P}C_3$

Bin 4: Prob($p_1p_2$) = ${}^{(P-2)}C_2/{}^{P}C_4$, etc.

In general, the probability of finding a two-product combination (say $p_1p_2$) in any bin x is given by:

Bin x, (where $P \geq x \geq 2$): Prob($p_1p_2$) = ${}^{(P-2)}C_{(x-2)}/{}^{P}C_x = x(x\text{-}1)/(P(P\text{-}1))$ (E1)

Now, assume the following bins and the number of items in each bin as in Table A1. Let us focus only on the even numbered bins. We can alternately focus on odd numbered bins as well. Since our focus is to use a two-product itemset $\{p_1, p_2\}$(worst case scenario), each iteration will make the transactions from an even-numbered bin go to the next even-numbered bin (e.g. bin 6 to bin 4, bin 4 to bin 2), and those from odd numbered bins go to other odd numbered bins. We choose even-numbered bins here since bin 2 being cleared of all transactions means that the conflict resolution and pruning algorithm has converged and stopped.

| Bin size (number of products in transaction) | Number of initial transactions | Probability of finding products $\{p_1,p_2\}$ in this bin* | Expected number of transactions with $\{p_1,p_2\}$ | Number of transactions after round 1 (for a two-product itemset, drop frequent itemsets from transaction) |
|---|---|---|---|---|
| **1** | $w_1$ | $z_1=0$ | | |
| **2** | $w_2$ | $z_2$ | $w_2z_2$ | $w_2(1-z_2)+w_4z_4$ |
| **3** | | | | |
| **4** | $w_4$ | $z_4$ | $w_4z_4$ | $w_4(1-z_4) + w_6z_6$ |
| **5** | | | | |
| **6** | $w_6$ | $z_6$ | $w_6z_6$ | $w_6z_6(1-z_6) + w_8z_8$ |
| **7** | | | | |
| **8** | $w_8$ | $z_8$ | $w_8z_8$ | $w_8(1-z_8)$ |

**Table A1: Bins and transactions**

*Probability of finding products $\{p_1, p_2\}$in a bin is given by (E1) above:
(x(x-1))/(P(P-1)), where x = bin size, P = total number of products

**Continuing on, number of transactions after round 2:**

Bin 2: $[w_2(1-z_2)+w_4z_4](1-z_2) + [w_4(1-z_4) + w_6z_6]z_4$

Bin 4: $[w_4(1-z_4) + w_6z_6](1-z_4) + [w_6z_6(1-z_6) + w_8z_8]z_6$

Bin 6: $[w_6z_6(1-z_6) + w_8z_8](1-z_6) + w_8(1-z_8)z_8$

Bin 8: $w_8(1-z_8)(1-z_8)$

Focusing on bin 2, and continuing iteratively, we find for $i^{th}$ round of iteration:

$$w_2^i = w_2^0(1-z_2)^i + w_4^0 z_4^{i-2}\left[(1-z_2)^{i-1} + (1-z_2)^{i-2}(1-z_4)^{i-2} + (1-z_4)^{i-1}\right] + w_6^0 z_4 z_6\left[(1-z_2)^{i-2} + (1-z_4)^{i-2} + (1-z_6)^{i-2}\right] + w_8^0 z_4 z_6 z_8$$

We can similarly write the functions for all other bins in a recursive manner. It is evident that these recursive functions are not closed form solutions. Hence we enumerate them numerically to see their properties.

**Step 2:**

**We enumerate under the following scenario:**

- Transactions are distributed in a Poisson distribution across all bins. The probability mass function for each bin is given by:

  $f(x) = = \frac{\lambda^x e^{-\lambda}}{x!}$, where x is the bin size.

We conducted an empirical study based on the recursive formulation for an upper bound on the resulting number of conflict-free itemsets. In particular, our interest centered on empirically establishing

the rate of growth of the number of conflict-free itemsets as a function of the number of transactions (*B*), total number of products (*P*), and the average number of products in a transaction ($\lambda$ for a Poisson distribution of products). We estimated the following functional specification:

$$\text{No. of conflict free rules} = AN^{\alpha 1}P^{\alpha 2}\lambda^{\alpha 3}$$

The data were generated from the recursive formulation by varying the values of *N* from 100 to 500,000; *P* from 20 to 100; and $\lambda$ from 2 to 10. A total of 222,850 data points were created and in each case the total number of conflict-free itemsets was determined from the recursive formulation. The regression results from a log-transformed specification are reported in Table A2.

| $\log(No.\ of\ conflict\ free\ rules) = \log(A) + \alpha 1 \log(N) + \alpha 2 \log(P) + \alpha 3 \log(\lambda)$ | | |
|---|---|---|
| Independent Variable | **Estimate** | **Standard Error** |
| **log(*A*)** | 0.3554795*** | 0.000385 |
| **log(*N*)** | 0.1015078*** | 2.688E-5 |
| **log(*P*)** | 2.0256472*** | 5.263E-5 |
| **log($\lambda$)** | 0.0292048*** | 5.263E-5 |
| Model Fit Statistics<br>**F-Value = 4.985E+8, $R^2$ = 0.999851, Adjusted $R^2$ = 0.999851 *** $p$ <0.00001** | | |

**Table A2: Empirical estimation of algorithm convergence**

The overall estimation results strongly indicate a near-perfect fit for the power-form specification of the number of conflict-free itemsets. The number of conflict-free itemsets increases at a rate of one-tenth power to the total number of transactions, as square of the total number of products, and is relatively impervious to the number of products in a transaction.

To perform a comparative analysis of the estimated upper bound on the number of conflict-free itemsets with the actual number that result, we analyzed the data set of customer invoices of an industry leading plastics manufacturer and distributor in the United States (further and more detailed analysis of this data set is presented in section 4). Four client industry segments were considered, and the results of the Poisson distribution of number of products in transactions are reported in Table A3.

| **Industry Segment** | ***N* (transactions)** | ***P* (number of products sold)** | **$\lambda$ (average number of products in transaction) - rounded** |
|---|---|---|---|
| **1** | 90 | 62 | 5 |
| **2** | 560 | 62 | 5 |
| **3** | 305 | 62 | 5 |
| **4** | 177 | 62 | 5 |

**Table A3: Poisson fit of data from firm invoices**

Table A4 presents the number of conflict-free itemsets estimated from the empirical fit, along with the actual number of itemsets that result from the execution of the conflict resolution strategy.

| **Industry Segment** | ***Estimated number of conflict-free itemsets*** | ***Actual number of conflict-free itemsets*** |
|---|---|---|
| **1** | 7542 | 104 |
| **2** | 10998 | 512 |
| **3** | 9849 | 215 |
| **4** | 8820 | 187 |

**Table A4: Estimated and Actual Number of Conflict-free Itemsets**

Together, the results indicate the overall effectiveness of frequent itemset pruning strategy in significantly reducing the number of itemsets for consideration.

## Online Supplement B. Computational steps for the Comparative results of total number of products in optimal portfolio and optimal revenue generated by proposed approach and Top-K approach In Table 7 (for "average" revenue capture ratio).

1) We determine the "average" revenue capture ratio (RCR) of a product $p$ for each segment. This way, we incorporate the global knowledge across stores to compute the RCR for all products. (We also computed the "90$^{th}$ percentile" RCR; it produced similar results overall.)

   Formula: $\Phi(p,\emptyset) = average\left(\frac{r_{i,p}}{\pi_i}, \forall i: \{p\} \cap a_j = \emptyset, \forall a_j \epsilon \theta_{t_i}\right)$ (equation 6 in the paper)

2) For each store, we compute the size of each industry segment around it.

   Formula: $\Upsilon(p,\emptyset) = \frac{\sum_{i:\{p\}\cap a_j=\emptyset, \forall a_j \epsilon \theta_{t_i}} \pi_i}{\sum_{\forall i} \pi_i}$ (equation 5 in the paper)

3) For each industry segment around each store, we estimate the revenue from product $p$ sold in that store.

   Formula: $\Upsilon(p,\emptyset)\Phi(p,\emptyset)\Pi x_p$ (equation 9 in the paper)

   This provides estimated revenues from each product sold in a given store. Note here that the concept of frequent itemsets is not used in this computation.

4) For each store, choose the Top-K products based on revenue ignoring sales associations among products.

5) Compute the revenue of each store using the objective function (equation 14 in the paper):

   $\sum_{p=1}^{P} \Upsilon(p,\emptyset)\Phi(p,\emptyset)\Pi x_p + \sum_{p=1}^{P}\sum_{j=1}^{|A|} \Upsilon(p,a_j)\Pi\left[\Phi(p,\emptyset)(x_p - y_j) + \Phi(p,a_j)y_j\right]$ (14)

   The rationale is that, although we ignore those associations during product portfolio selection, the beneficial as well as cannibalizing influence of products will be present in the real sales. Ignoring these patterns during revenue computations would result in a biased estimate. Therefore we use equation 14 to compute the revenue for the selected portfolio.