

# Product selection for promotion planning

Yinghui Yang · Chunhui Hao

Received: 20 August 2009 / Revised: 26 February 2010 / Accepted: 13 July 2010 /

Published online: 21 July 2010

© The Author(s) 2010. This article is published with open access at Springerlink.com

**Abstract** This paper addresses a very important question—how to select the right products to promote in order to maximize promotional benefit. We set up a framework to incorporate promotion decisions into the data-mining process, formulate the profit maximization problem as an optimization problem, and propose a heuristic search solution to discover the right products to promote. Moreover, we are able to get access to real supermarket data and apply our solution to help achieve higher profits. Our experimental results on both synthetic data and real supermarket data demonstrate that our framework and method are highly effective and can potentially bring huge profit gains to a marketing campaign.

**Keywords** Data mining · Marketing · Promotion planning · Optimization · Market basket analysis

## 1 Introduction

Some of the most common reasons why retailers have sales involve getting rid of merchandise, introducing new products and selling seasonal products (e.g. Christmas and Valentine's day). With a majority of stores keeping track of their sales records nowadays, it is becoming increasingly feasible and important to tailor promotions and sales according to the knowledge learned from the transactions. Through these transactions, customers' purchasing behavior can be learned, and promotions can be designed according to such behavior in order to bring maximum profit to the marketing campaign.

One of the most popular data mining techniques used for analyzing customer shopping-basket data is the association rule discovery technique, and much research has been dedicated

---

Y. Yang (✉)

Graduate School of Management, University of California, Davis, CA 95616, USA  
e-mail: yiyang@ucdavis.edu

C. Hao

Chinese Academy of Sciences, 100190 Beijing, China  
e-mail: chunhui.hao@ia.ac.cn

to the development of more efficient and effective association rule algorithms ([2,3,5,9,11,14–16,23]). This technique discovers associations among products, from which we can infer whether a product or a set of products can positively affect the sales of other products. The association rule extraction process often generates an overwhelming number of rules, which are very difficult for managers to comprehend, let alone put into action ([1]). Utilizing these rules for a given decision-making task can be very challenging. Directly incorporating managers' goals into the data-mining process can potentially provide more value to managers.

In this paper, the decision-making task we consider is how to select the right products for store promotions. For a retailer (either online or offline), a very common decision is to select products for sale so that the money spent on the promotion can achieve the maximum benefit. We propose an approach that can directly utilize the results of the association rule analysis. The approach will take the rules as input and generate a list of products the store can promote. Integrating the results of association rule analysis into the product-selection process can provide potentially significant value for retailers.

In this paper, we provide data-mining solutions for the product-selection problem for store promotions, which is a very real and important problem faced by supermarkets, grocery stores, online retailers, etc. No previous research has studied the exact problem we define in this paper. In the following paragraphs, we discuss the research which is related to the problem we study. The most relevant data-mining research is the study of product assortment, which is a fundamentally different problem ([6–8,20,22,21]). The core decision in the product assortment problem is to decide what items a store should carry, given limited retail space. And the techniques used for product assortment cannot be directly applied to the product-promotion problem. In [6,7] and [8], the assortment first needs to be consistent with the store's image by including some basic products, and the added products should be selected to maximize cross-sales potential with the basic products. In [22] and [21], the value of the products that are not carried by the store will be considered as a loss to the store. Their goal is to minimize such loss. Our focus is to select products with the highest promotional effect. Therefore, our problem has a totally different objective function to optimize.

In the marketing literature, research has not been focusing on selecting the best products to promote out of all the products available. Related research has studied various aspects of the problem, such as cross-selling effects ([13]), complementary products ([4]), purchase connections among multiple product categories ([18]), etc. For example, [17] studies the following two hypotheses: (a) Cross-effects of retail promotions depend on the strength of the complementary relationship; (b) Retail promotions of a product have a positive effect on the purchase quantity of the complementary product. [10] investigates the timing behavior of households in two product categories for which the decision of a household to make a purchase in one category at a given point in time influences purchases in the other category.

The main contributions of this paper are as follows. First, the paper addresses a very important question—how to select the right products to promote in order to maximize promotional benefit. Second, we set up a framework to incorporate managers' promotion decisions into the data-mining process, formulate an optimization problem, and propose a heuristic search solution to discover the right products to promote. This research incorporates profit maximization into the data-mining process and thus contributes to the actionability of data mining. Moreover, we are able to access real supermarket data and apply our solution to help achieve higher profits.

The remainder of the paper is organized as follows. Section 2 defines the problem and introduces several definitions of the value of a set of products. Section 3 develops a method

to find the top products to maximize profit. Section 4 reports the experimental results on both synthetic data and real supermarket data, and Sect. 5 concludes with discussion on future work.

## 2 Problem definition

Product selection for the purpose of store promotions is to select a (small) set of products that will have a strong promotional effect on other products. The assumption is that with the promotion of these selected products, people are more willing to buy them, and thus the associated products as well. This can potentially increase profits for the store. The extent of a product's promotional effect is derived from purchase transactions where multiple products are purchased. The decision the store needs to make is which  $n$  products to promote in order to maximize the campaign profit, given that the purchases of products are correlated.

We consider a market basket transaction database that contains  $m$  purchase transactions  $T = \{t_1, t_2, \dots, t_m\}$ , each of which contains a subset of the following  $N$  products or items  $I = \{i_1, i_2, \dots, i_N\}$  (we use item and product interchangeably in this paper). The profit of a selected set of items  $S \in I$  is calculated over  $T$ . Given the number of items to be selected,  $n$ , the goal is to select the set so that the profit of the promotion is maximized.

An item should be selected if it can strongly promote the sales of other items. To see whether an item or a set of items has a positive promotional effect on another item, we define the following term called Promotional Effect.

**Definition 1** (*Promotional effect*) The promotional effect of a set of items ( $L$ ) on another item ( $R$ ), denoted as  $\text{Promo}(L \rightarrow R)$ , is defined as  $P(R|L) - P(R)$ , where  $P(R)$  is the probability of the purchase of  $R$  across all transactions, and  $P(R|L)$  is the probability of the purchase of  $R$  among transactions containing  $L$ .  $\square$

For a certain association rule  $L \rightarrow R$  with a single item on its right-hand side,  $P(R|L)$  corresponds to the confidence of the rule, and  $P(R)$  is the support of the item  $R$ . If  $L$  has a positive promotional effect on  $R$ ,  $\text{Promo}(L \rightarrow R) > 0$ ; otherwise,  $\text{Promo}(L \rightarrow R) < 0$ . In the following equation, we define the value of a single item based on the promotional effect of a set of rules.

**Definition 2** (*Value of a single item*)(denoted as  $V_1()$ )

$$V_1 = \sum_{\text{rules in } S_1} [NT * UP_{RHS} * \text{Promo}(\text{rule})]$$

$\square$

Among all the association rules discovered from the purchase transactions,  $S_1$  is the set of rules with the item on promotion on the left-hand side and one single item on the right-hand side (e.g.  $A \rightarrow B$ ). For rules with multiple items on the right-hand side, we can simply consider them separately. For example,  $A \rightarrow \{B, C\}$  can be separated into  $A \rightarrow B$  and  $A \rightarrow C$ .  $NT$  is the number of transactions supporting the rule.  $UP_{RHS}$  is the unit profit of the item on the right-hand side of the rule.  $RHS$  is short for the item on the right-hand side of the rule. The *lift* of the rule is defined as confidence of the rule, divided by the support of the  $RHS$ , and it can also be used to measure whether the purchase of the item on the left-hand side is positively affecting the purchase of the item on the right-hand side.

Because lift value is always positive, we cannot directly use it in Definition 2. The reason is that for rules with lift smaller than 1, we do not want it to positively contribute to the value of the item on the left-hand side. Therefore, we use the promotional effect of a rule, which is  $\text{Promo}(\text{rule})$  we defined in Definition 1. This value will be negative if the lift is smaller than 1, and will be positive if it is greater than 1. Its value will range between  $[-1, 1]$ .

The intuition behind Definition 2 is as follows. The value of an item in promoting the sales of other items is calculated based on the rules with this item on the left-hand side. For each such rule, we will also look at the extent of its impact through the number of transactions where it holds, and the unit profit of the item on the right-hand side of the rule. The higher the number of transactions and the unit profit of the item derived from the item on sale (i.e. the item on the left hand side of the rule), the higher the promotional value of the item on the left-hand side. The promotional value of some of the rules in this set will be positive, and some will be negative. The aggregated value of this item across all rules is eventually used as the value of this item.

The problem with one item on promotion is fairly simple. We can calculate the value of each item based on Definition 2, and pick the item with the highest value to promote. However, promotions are often run on multiple items simultaneously. A naïve approach would be to rank the items according to their values defined in Definition 2, and to pick the top  $n$  items to promote. However, this solution may generate a far less profitable result. The major reason is because of the cross-selling effect among items. For example, items A, B, C are the top three items, the value of A and B combined together is not necessarily higher than the value of A and C combined. The added value of B may not be as high after A is on promotion. One reason is because item A might encourage the purchase of item B, and item B's coupon may not be as useful, even though this item has high value on its own. Another reason is that with multiple items, some promotional effect comes from rules with multiple items on the left-hand side (e.g.  $\{A, C\} \rightarrow D$ ), and we need to incorporate this into the value calculation process. How to select multiple items so that their potential benefit is maximized is very important. In the following paragraph, we define the Exclusive Value of a set of items, and the (Inclusive) Value of a set of items.

**Definition 3** (*Exclusive value of an itemset*)  $V_E()$

$$V_E = \sum_{\text{rules in } S_E} [NT * UP_{RHS} * \text{Promo}(\text{rule})]$$

□

For a set of items  $L$ , the exclusive value is defined as the value generated from the set of rules ( $S_E$ ) with  $L$  on its left-hand side and a single item on its right-hand side. Again, rules with multiple items on the right-hand side can be broken into several rules, each with a single item on the right-hand side. The reason why this is called exclusive value is because we only consider the value of the rules with all the items in the set on its left-hand side, and we exclude the value of the rules with a subset of the items on its left-hand side.

When multiple items are promoted, we should not only consider the value derived from the rules with the entire itemset on the left-hand side, but also the value derived from the rules with the subset of the itemset on the left-hand side. For example, when two items are being promoted, we should also include the value derived from the rules with  $A$  or  $B$  on the left-hand side, i.e.  $V_E(A) + V_E(B)$ . Since both the transactions with  $A$  and transactions with  $B$  contain transactions with  $\{A, B\}$ , we need to exclude the double-counted transactions. Therefore, the

true value of these two items will be the summary of the value of each item excluding the over counted value (i.e.  $V_E(A) + V_E(B) - V_E(A, B)$ ). In the following paragraph, we defined the (inclusive) value of an itemset.

**Definition 4** (*Value of an itemset*)  $V(\cdot)$  For an itemset  $G = \{i_1, i_2, \dots, i_k\}$  with  $k$  items, its value is calculated as following

$$V(G) = \sum_{j_1=1}^k V_E(i_{j_1}) - \sum_{1 \leq j_1 < j_2 \leq k} V_E(i_{j_1}, i_{j_2}) + \sum_{1 \leq j_1 < j_2 < j_3 \leq k} V_E(i_{j_1}, i_{j_2}, i_{j_3}) \\ + \dots + (-1)^{k-1} V_E(i_1, i_2, \dots, i_k)$$

□

Since there are not a lot of significant rules with a large number of items on the left-hand side, the size of the subset of the itemset that we consider can be limited (e.g. four). For example, when we limit the number of items on the left-hand side of a rule to four, the value of an itemset with five items  $\{A, B, C, D, E\}$  is as follows.

$$V(A, B, C, D, E) = V_E(A) + V_E(B) + V_E(C) + V_E(D) + V_E(E) - V_E(A, B) \\ - V_E(A, C) - V_E(A, D) - V_E(A, E) - V_E(B, C) - V_E(B, D) \\ - V_E(B, E) - V_E(C, D) - V_E(C, E) - V_E(D, E) + V_E(A, B, C) \\ + V_E(A, B, D) + V_E(A, B, E) + V_E(B, C, D) + V_E(B, C, E) \\ + V_E(C, D, E) - V_E(A, B, C, D) - V_E(A, B, C, E) - V_E(B, C, D, E)$$

The value of a single item is the same as its exclusive value. To expedite computation, we can calculate the value of  $k + 1$  items based on the value of  $k$  items, if it is already known.

**Theorem 1** (*Recurrent calculation of the value of an itemset*) For an itemset  $G_1 = \{i_1, i_2, \dots, i_k, A\}$  with  $k+1$  items, its value can be calculated based on the value of  $G = \{i_1, i_2, \dots, i_k\}$ .

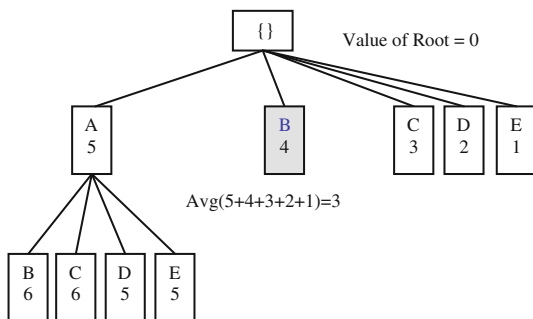
$$V(G_1) = V(G) + V_E(A) - \sum_{1 \leq j_1 \leq k} V_E(i_{j_1}, A) + \sum_{1 \leq j_1 < j_2 \leq k} V_E(i_{j_1}, i_{j_2}, A) \\ + \dots + (-1)^k V_E(i_1, i_2, \dots, i_k, A)$$

□

This recurrent property is very useful in the method we use to select the top products to promote, because the process mostly involves calculating the value of  $k + 1$  items based on the value of  $k$  items (see Sect. 3 for more details). Once we define the value of multiple items, the goal is to select the top items with the highest value.

### 3 Select the top products

The purchase transactions include  $N$  distinctive items. Among these  $N$  items, we select  $n$  items which can potentially generate the highest promotional value. The problem can be written into the following optimization problem.

**Fig. 1** Example for search

$$\begin{aligned} & \text{Max } V(S) \\ & \text{s.t. } \begin{cases} I = \{i_1, i_2, \dots, i_N\} \\ S \in I \\ |S| = n \\ n < N \end{cases} \end{aligned}$$

$I$  is the set of all the  $N$  products we consider, and  $S$  is the set of  $n$  products we select to promote in order to maximize  $V(S)$ . Since we cannot try every possible combination of  $n$  items and calculate the value, we present a tree-based heuristic search to try and find the top  $n$  items out of the  $N$  distinctive products the seller carries.

The intuition behind the heuristic search is as follows. Assume that we have a complete tree of depth  $n$ . The root of the tree is null, and each child node of the root corresponds to a unique product (so there are  $N$  children for the root node). For each of the other non-leaf nodes in the tree, its children include all the items that have not appeared in the path leading from the root to that node. The tree has  $n$  levels because we only select  $n$  items to promote. The optimal solution will be one of the branches in the tree. Assume that we follow a strict greedy depth first search. For any non-leaf node, we choose the child of the node if the value of the set of items on the path from the root to this child is the greatest. If we follow this greedy strategy to search, we will essentially go down one branch and never backtrack, which is very likely to lead to an inferior solution. Therefore, we designed a mechanism to allow backtracking when the marginal profit of going down a branch is not high enough.

Take Fig. 1 as an example. We first select A because  $V(A)$ , which is 5, is the greatest among the children of the root node. We then follow the branch of A and check the children of A.  $V(A, B)$  and  $V(A, C)$ , which are 6, are the greatest. However, the marginal profit  $V(A, B) - V(A)$ , which is  $6 - 5$ , is quite low. We compare this margin with the average margin of the previous level, which is the average of  $V(A)$ ,  $V(B)$ ,  $V(C)$ ,  $V(D)$  and  $V(E)$  ( $= 3$ ). If this margin is low ( $< 3$ ), then this branch is not worth further exploration, and we move back to the highlighted node B. To simplify computation (which does not change our underlying intuition about marginal profit), we adjust the value of a node to be the value of the set of items leading from the root to that node (e.g.  $V(A, B)$ ) minus the average value of its parent and the parent's siblings. Then we can simply compare all the adjusted values of all the leaf nodes, and move to the leaf node with the greatest adjusted value for future searches.

Figure 2 below gives the detailed search algorithm (we use TP (Top Product selection) to refer to this algorithm) and the example presented in Fig. 3a–d further illustrates the search process.

The path leading from the root node to a leaf node of the tree represents the items we can select. We compare the value of all the leaf nodes and pick the node with the highest value

**Inputs:**

1. Dataset  $D$  with  $m$  purchase transaction. Among these transactions, there are  $N$  distinctive items,  $I$ .
2. Association rule discovery algorithm,  $R$ , that discovers rules which can be used to calculate value of a set of items.

**Output:**

1. A set of  $n$  selected items,  $S$ . The algorithm's goal is to maximize the value of  $S$ .

**Procedure:**

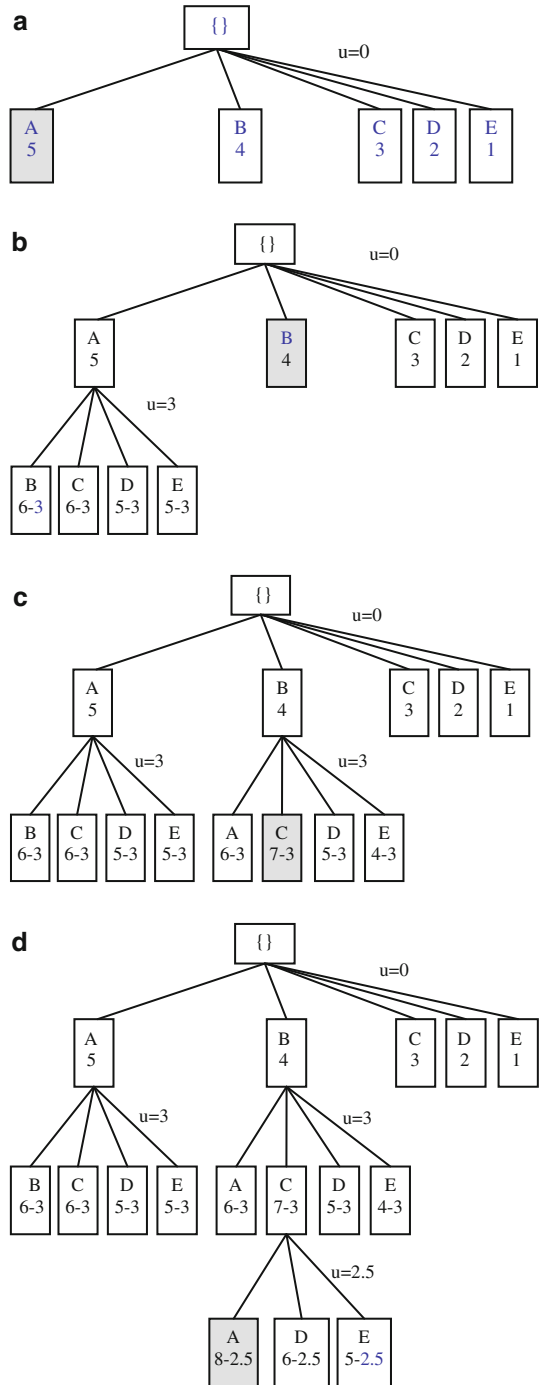
1.  $S = \emptyset$ , output.
2. Generate a set of association rules  $P$  by applying  $R$  to  $D$ . Store the promotional effect of the rules.
3. Initiate a search tree by creating the root of the tree and label it as "null".  $ANODE$  = Root node of the tree. We assign the value 0 to the root.  $ANODE$  stands for active node.
4.  $ExtendTree(ANODE, u)$ .  $S$  = Items on the path from the root of the tree to  $ANODE$ . We extend the tree by adding children nodes to  $ANODE$ . Each child node corresponds to an item in  $I-S$ . And for each child node  $CNode$ , we use Lemma 1 to calculate its value  $(V(S \cup \text{item in } CNode) - u)$  based on  $V(S)$ .  $u$  is the average value of the parent node and the parent's sibling nodes. Here for the root node,  $u=0$ .
5.  $ANODE = GetMaxLeafChild()$ . Compare the value of all the leaf nodes of the tree, select the node with the highest value.
6.  $S = S \cup (\text{the Item in } ANODE)$
7. While  $|S| < n$  do {
8.      $ExtendNode(ANODE, u)$ .
9.      $ANODE1 = ANODE$
10.     $ANODE = GetMaxLeafChild()$ .
11.    If  $ANODE1$  is the parent of  $ANODE$  {
12.        $S = S \cup (\text{the Item in } ANODE)$
13.    else
14.        $S = \text{Items on the path from Root to } ANODE$ .
15.    }endIf
16. }
17. Output  $S$ .

**Fig. 2** Top product selection—TP method

to further grow the tree. We introduce a factor  $u$  to make sure that the benefit of growing the tree further down a branch is higher than the marginal benefit we can achieve by growing down other branches.

In the following paragraph, we use an example to illustrate the steps ( $N = 5$ ,  $n = 3$ ).

- Step 1. Initiate the tree. The root is null. We extend the tree by adding child nodes to the root. Each child corresponds to an item and the value of each item is written underneath. The values of all the leaf nodes are compared, and  $A$  is the node with the largest value.
- Step 2. We extend the tree by adding child nodes to node  $A$ . Each child node corresponds to an item which is not on the path to node  $A$ . We calculate  $u$  by taking the

**Fig. 3** Search example



average of the values of the nodes that share the same parent with node  $A$  (including  $A$ ),  $u = (5+4+3+2+1)/5 = 3$ . The value of each child node is the value of the set of items on the path from the root to that node, minus  $u$  (e.g. the value of node  $B$  is  $V(A, B) - u$ , where  $u = 3$ ). The values of all the leaf nodes are compared, and highlighted node  $B$  on the first level of the tree has the highest value (4), then we traverse to that  $B$  node.

- Step 3. Similar to step 2, we extend the tree from the highlighted  $B$  node. Among all the leaf nodes, the highlighted  $C$  node has the highest value.
- Step 4. The tree is further extended, and the highlighted  $A$  node has the highest value among all leaf nodes. Since the number of items on the path to the highlighted  $A$  node has reached the desired number ( $n = 3$ ), we stop and output the selected items  $\{B, C, A\}$ .

In a way,  $u$  represents the opportunity cost of searching down a certain node. The additional benefit needs to be high enough in order to pursue the direction further.

## 4 Experiments

We conduct experiments to evaluate our approach. We compare our method with two other methods, the naïve method, which picks the top items based on their standalone value, and the frequency-based method, which selects the most frequently purchased items to promote.

### 4.1 Synthetic data

In our experiment, we use the IBM synthetic data generator ([3]) to generate data sets with different sets of parameter values. On top of the IBM generator, we added a unit profit generator to randomly generate unit profit for individual items. We follow the following formula to generate unit profit,  $\text{Profit} = 1 + \text{Unifrom}(0,1) * 1000$  (where  $\text{uniform}(0,1)$  returns a value between 0 and 1). The association rule discovery algorithm we used is FP-tree ([12]). Table 1 presents the results, varying the total number of items ( $N$ ) and the number of items to be selected for promotion ( $n$ ). The average number of items per transaction is set at 10, and the average number of items per frequent itemset is 4).

As shown in Tables 1 and 2, our method significantly outperformed the naïve method and the frequency-based method.

### 4.2 Results from real supermarket shopping data

We acquired a shopping transaction data set from a major supermarket. The supermarket has a large grocery store as well as several floors of high-end merchandise including

**Table 1** Profit improvement over naïve method (in percentage)

	$N = 50$ (%)	$N = 100$ (%)	$N = 150$ (%)	$N = 200$ (%)
$n = 5$	64.06	67.60	67.60	67.60
$n = 10$	187.10	308.10	308.10	308.10
$n = 15$	129.53	232.66	228.81	230.93
$n = 20$	57.09	128.72	128.09	127.45
$n = 25$	16.51	188.13	191.81	189.85

**Table 2** Profit improvement over frequency-based method (in percentage)

	$N = 50$ (%)	$N = 100$ (%)	$N = 150$ (%)	$N = 200$ (%)
$n = 5$	94.89	77.39	76.28	76.28
$n = 10$	152.49	165.41	161.50	161.50
$n = 15$	133.21	146.05	140.58	142.14
$n = 20$	106.63	193.52	189.47	188.67
$n = 25$	19.17	558.94	526.60	511.96

**Table 3** Improvements over two other methods

	TP over naïve (%)	TP over frequency (%)
$n = 5$	18.76	132.36
$n = 10$	52.64	204.11
$n = 15$	28.82	219.66
$n = 20$	19.35	156.67
$n = 25$	29.57	190.80
$n = 30$	23.05	200.91

clothing, jewelry, office supplies and other household items. Hence, the product selection is very broad. The data contains very detailed information about shopping transactions generated by 5,000 customers, including shopper card numbers as unique identification, products purchased in each shopping transaction, time and amount of a transaction, product purchase price, etc.

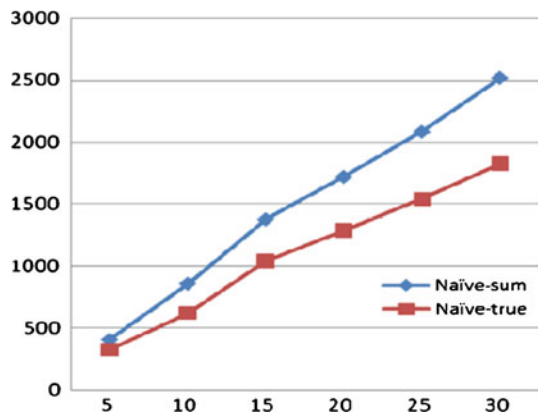
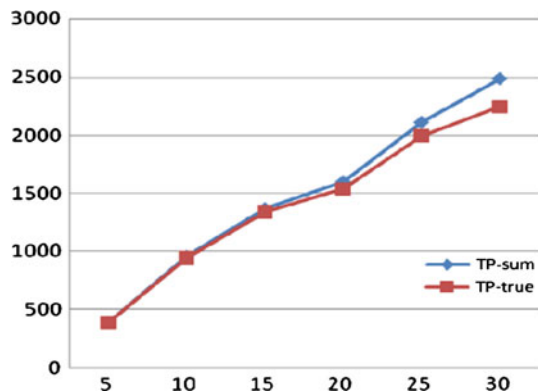
We processed one year's data from year 2006, and there are 1,20,536 transactions in total. Among these transactions, there are 76,879 unique products. The average number of products per transaction is 4.14. We set the profit of each product to be 10% of its price. We again applied FP-tree ([12]) on all the transactions to discover association rules. We set the support threshold to be 0.01% (around 12 transactions). We set the confidence threshold to be 0, because rules with low confidence value could possibly have a high promotional effect (confidence of the rule—support of the RHS of the rule). Among all the 76,879 unique products, 7,307 of them satisfy the minimum support threshold. We further filtered out the items whose single item value,  $V_1()$ , is negative. This leaves us with 976 items to consider (i.e.  $N = 976$ ).

Again, we implemented three methods, the naïve method, the frequency-based method and our method (TP). We use 5-fold cross-validation to evaluate the methods using hold-out data. For each run, we randomly select 20% of the transactions as the hold-out transactions. We use the 80% training data to select the top  $n$  items, then we use the 20% hold-out data to calculate the value of the  $n$  items ( $V()$ ) selected using the training data. The results presented in Tables 3, 4 and Figs. 4, 5, 6 are the average of 5 runs and they are all results on the hold-out data. Table 3 lists the improvements of our TP method over the naïve method and the frequency-based method.

Table 4 below lists the summary of single item values for the selected products (-sum) and the true value of the selected set of products (-true). For example, if  $\{A, B, C\}$  are the selected products, then -sum is  $V_1(A) + V_1(B) + V_1(C)$  and -true is  $V(A, B, C)$ . As we can tell from Table 4 and Figs. 4, 5, the Naïve method has a much higher difference between the -sum

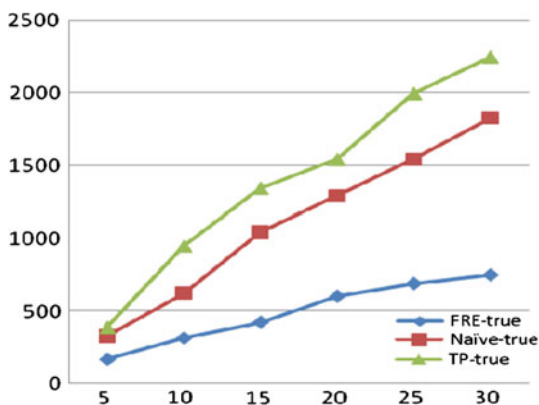
**Table 4** Sum of single item values and the true value of the set of items

	TP		Naïve		Frequency	
	-sum	-true	-sum	-true	-sum	-true
$n = 5$	386	386	408	325	167	166
$n = 10$	959	948	860	621	313	312
$n = 15$	1368	1342	1377	1041	423	420
$n = 20$	1602	1539	1717	1289	605	599
$n = 25$	2114	1994	2082	1539	693	686
$n = 30$	2490	2246	2516	1825	767	746

**Fig. 4** Naïve method—sum of single item values and the true value of the set of items**Fig. 5** TP—sum of single item values and the true value of the set of items

value and the -true value. This indicates that there is quite a lot of overlapping promotional effect among the products selected by the Naïve method. As for the TP-method, the difference is very small, indicating the effectiveness of the TP-method in selecting products whose promotional effects do not overlap as much. The difference between the FRE-sum and FRE-true is also low. This is because the items with the highest frequency are not chosen based on the associations they have with other items, thus there are very little overlaps between the promotional effects of different items chosen based on frequency. Figure 6 further illustrates

**Fig. 6** Comparisons between the naïve method, frequency-based method and TP—the true value of the set of items



that the value of the set of items selected by TP (TP-true) is much high than that selected by the Naïve method (Naïve-true) and the frequency-based method (FRE-true).

## 5 Conclusion

In this paper, we address the problem of selecting the right products for promotions, which is a very relevant and important problem for many businesses including supermarkets, grocery stores and other offline/online retailers. Our solution is highly actionable for marketing managers, since we directly incorporate a manager's decisions into the problem. We define an optimization function and provide a heuristic search to discover a good solution. The experimental results on both synthetic data and real supermarket data demonstrate that our framework and method are highly effective and can potentially bring huge profit gains to marketing campaigns.

While we demonstrated promising results in this paper, there are also limitations, which we plan to address in future research. First, we only consider the short-term associations between products by looking at associations at the transaction level. As pointed out in ([19]), the longitudinal aspect of promotion should be studied alongside of short-term effects. In the future, we plan to look at long-term promotional effects. [19] also points out that associated products are not necessarily complements. In the future, we can conduct more research to validate whether the products selected truly help the stores achieve higher profit. A good way for validating this is to persuade the store to do trial promotions according to the products we have selected to observe the true effect of a promotion. Furthermore, we only consider product selection in this paper. In the future, we can consider incorporating customer selection together with product selection in a promotional campaign (i.e. targeted marketing—which products a store should promote to different customers).

**Acknowledgments** This research is partly supported by the National Natural Science Foundation of China through grants 90924302 and 70890084, and the Ministry of Science and Technology of China through grant 2006AA010106.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

## References

1. Adomavicius G, Tuzhilin A (1997) Discovery of actionable patterns in databases: the action hierarchy approach. In: Proceedings of the third international conference on knowledge discovery and data mining, pp 111–114
2. Agrawal R, Imielinski T, Swami A (1993) Mining association rules between sets of items in large databases. In: Proceedings of ACM SIGMOD conference on management of data, pp 207–216
3. Agrawal R, Srikant R (1994) Fast algorithms for mining association rules. In: Proceedings of the 20th international conference on very large databases, pp 487–499
4. Betancourt R, Gautschi D (1990) Demand complementarities, household production, and retail assortments. *Mark Sci* 9(2):146–161
5. Boley M, Grosskreutz H (2009) Approximating the number of frequent sets in dense data. *Knowl Inf Syst* 21(1):65–89
6. Brijs T, Swinnen G, Vanhoof K et al (1999) Using association rules for product assortment decisions: a case study. In: Proceedings of the fifth international conference on knowledge discovery and data mining, pp 254–260
7. Brijs T, Goethals B, Swinnen G et al (2000) A data mining framework for optimal product selection in retail supermarket data: the generalized PROFSET model. In: Proceedings of ACM SIGKDD, pp 300–304
8. Brijs T, Swinnen G, Vanhoof K et al (2004) Building an association rules framework to improve product assortment decisions. *Data Min Knowl Discov* 8:7–23
9. Brin S, Motwani R, Ullman J et al (1997) Dynamic itemset counting and implication rules for market basket data. In: Proceedings ACM SIGMOD international conference on management of data, pp 255–264
10. Chintagunta P, Haldar S (1998) Investigating purchase timing behavior in two related product categories. *J Mark Res* 35:43–53
11. Hämmäläinen W (2010) StatApriori: an efficient algorithm for searching statistically significant association rules. *Knowl Inf Syst* 23(3):373–399
12. Han J, Pei J, Yin Y et al (2004) Mining frequent patterns without candidate generation: a frequent-pattern tree approach. *Data Min Knowl Discov* 8:53–87
13. Li S, Sun B, Wilcox R (2005) Cross-selling sequentially ordered products: an application to consumer banking services. *J Mark Res* 42(2):233–239
14. Liu H, Lin Y, Han J (2009) Methods for mining frequent items in data streams: an overview. *Knowl Inf Syst* (Online First)
15. Papapetrou P, Kollios G, Sclaroff S et al (2009) Mining frequent arrangements of temporal intervals. *Knowl Inf Syst* 21(2):133–171
16. Park J, Chen M, Yu P (1995) An effective hash based algorithm for mining association rules. In: Proceedings of the ACM SIGMOD international conference on management of data, pp 175–186
17. Poel D, Schamphelaere J, Wets G (2004) Direct and indirect effects of retail promotions on sales and profits in the do-it-yourself market. *Expert Syst Appl* 27(1):53–62
18. Seetharaman P, Chib S, Ainslie A et al (2005) Models of multi-category choice behavior. *Mark Lett* 16(3–4):239–254
19. Vindevogel B, Poel D, Wets G (2005) Why promotion strategies based on market basket analysis do not work. *Expert Syst Appl* 28(3):583–590
20. Wang K, Su M (2002) Item selection by “hub-authority” profit ranking. In: Proceedings of the eighth ACM SIGKDD, pp 652–657
21. Wong R, Fu A, Wang K (2005) Data mining for inventory item selection with cross-selling considerations. *Data Min Knowl Discov* 11:81–112
22. Wong R, Fu A, Wang K (2003) MPIS: Maximal-profit item selection with cross-selling considerations. In: Proceedings of ICDM, pp 371–378
23. Zaki M, Parthasarathy S, Ogihara M et al (1997) New algorithms for fast discovery of association rules. In: Proceedings of the third international conference on knowledge discovery and data mining, pp 283–286

## Author Biographies



**Yinghui Yang** is an assistant professor of the Graduate School of Management at University of California, Davis. She received her Ph.D. in Operations and Information management from The Wharton School at the University of Pennsylvania, and B.E. in Management Information Systems from The School of Economics and Management at Tsinghua University. Her research is interdisciplinary between data mining and marketing. Her research has been published in top data mining journals (e.g. IEEE Transactions on Knowledge and Data Engineering, ACM Transactions on Database Systems, and Knowledge and Information Systems) and Marketing journals (e.g. Marketing Science).



**Chunhui Hao** is a Ph.D. student in the Institute of Automation at Chinese Academy of Sciences.