A Provable Algorithmic Approach to Product Selection Problems for Market Entry and Sustainability

Silei Xu, Yishi Lin, Hong Xie, and John C.S. Lui
Department of Computer Science and Engineering
The Chinese University of Hong Kong
New Territories, Hong Kong
{slxu, yslin, hxie, cslui}@cse.cuhk.edu.hk

ABSTRACT

Given the globalized economy, how to process the heterogeneous web data so to extract customers' purchase behavior is crucial to manufacturers who want to enter or sustain in a competitive market. To maximize the sales, manufacturers not only need to decide what products to produce so to meet diverse customers' requirements, but at the same time, compete with competitors' products. In this paper, we present a general framework for the following product selection problems: (1) k-**BSP** problem, which is for a manufacturer to enter a competitive market, and (2) k-**BBP** problem, which is for a manufacturer to sustain in a competitive market. We propose several product adoption models to describe the complex purchase behavior of customers, and formally show that these problems are *NP-hard* in general. To tackle these problems, we propose computationally efficient greedy-based approximation algorithms. Based on the submodularity analysis, we prove that our algorithms can guarantee a $(1-1/e)$-approximation ratio as compared to the optimal solutions. We perform large scale data analysis to show the efficiency and accuracy of our framework. In our experiments, we observe 1,300 to 250,000 times speedup as compared to the exhaustive algorithms, and our solutions can achieve on average 96% of solution quality as compared to the optimal solutions. Finally, we apply our algorithms on web dataset to show the impact of customers' different purchase behavior on the results of product selection.

Categories and Subject Descriptors

F2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

Keywords

Product selection; market entry; market sustainability; submodular set function; approximation algorithm

SSDBM '14, June 30 - July 02 2014, Aalborg, Denmark

http://dx.doi.org/10.1145/2618243.2618250

1. INTRODUCTION

In the modern globalized and fast evolving economy, consumer markets are becoming very dynamic. This leads to fierce competitions among manufacturers who compete for potential customers. Manufacturers need to find effective means to enter a new market, while existing manufacturers in a market need to consider how to sustain so as to keep up with the competition. To succeed in a market, manufacturers need to create better products over their competitors by considering not only existing products in the market, but also customers' behavior, perception and preference.

The popularity of the Internet has revolutionalized customers' product adoption behavior and the way manufacturers introduce new products. Customers can now share their opinions on products in the form of ratings or product reviews through various web services, e.g., Amazon, TripAdvisor. Potential customers can access these ratings or reviews, and purchase products based on ratings, reviews and their own preference, instead of solely relying on the sales pitch from salesmen or traditional advertisements. Furthermore, manufacturers can use these ratings or reviews (i.e., heterogeneous web data) to gain a better understanding of customers' preference or requirement so to guide their manufacturing decisions. This leads to new challenges on how to effectively introduce new products into the market.

To introduce new products, a manufacturer normally has a set of candidate products that they may produce [10, 13]. The key constraint is the production budget, say the manufacturer can only afford to produce a subset of these candidate products. The goal of a manufacturer is to select a subset of these candidate products that may bring the highest profit by considering customers' preferences or requirements, as well as the competition of existing products in the market.

In this work, we consider the scenario that each product can be described by a finite number of attributes. Customers' requirement on each attribute can be obtained by means of mining product ratings or reviews on the web. We call this as the "*product selection problem*", where a set of existing products, a set of candidate products, and customers' requirement on each products' attribute are available. Given a production budget $k \geq 1$, the objective is to select k most marketable products from the candidate products so that the manufacturer can either enter or sustain in the market with the largest sales.

The problem of selecting k most marketable products is challenging. Firstly, many human factors may affect cus-

tomers' product adoption behavior, which have a significant impact on the selection result. However, there is a lack of a formal model and analysis of product adoption behavior. Secondly, finding the optimal k products to produce is computationally expensive. For example, even for the simplest product adoption model [10], is *NP-hard*. And the problem becomes more challenging when we have to consider the complicated customers' behavior in the real world market [4]. The aim of this paper is to tackle these two challenges by proposing formal models to analyze the impact of various factors on market entry or sustainability, and present efficient algorithms on these generalized product selection problems. To the best of our knowledge, this is the first paper that provides formal models and analysis of such problems. Our contributions are:

- We formulate product selection problems for a new manufacturer to enter a market (or the k-**BSP** problem), and how to sustain in a competitive market (or the k-**BBP** problem).
- We propose three new and general product adoption models to capture various human factors that may affect customers' behavior in adopting products.
- We formally prove these product selection problems are *NP-hard* in general. We propose computationally efficient approximation algorithms for the product selection problems. By proving the submodularity property, we show our algorithms can provide a high theoretical performance guarantee: $(1-1/e)$-approximation as compared to the optimal solutions.
- We perform experiments using both synthetic data and real-world web data (i.e., RateBeer.com) to validate our framework and to examine factors that may affect the product selection. The results of these experiments show the efficiency and accuracy of our algorithms and the significant impact of different adoption models.

This is the outline of this paper. In Section 2, we present three general product adoption models to describe the customers' behavior. We also define the expected sales of products and formulate product selection problems for both market entry (k-**BSP**) and sustainability (k-**BBP**). In Section 3, we present efficient exact algorithms for the case that $k=1$ and prove that finding the exact solutions is *NP-hard* when $k>1$. In Section 4, we present approximation algorithms for the market entry and sustainability problems, and show they are not only computationally efficient but also with theoretical performance guarantee. In Section 5 and 6, we present experimental results on synthetic data and web data respectively. Related work is given in Section 7, and Section 8 concludes.

2. MATHEMATICAL MODELS & PROBLEM FORMULATION

In this section, we first present the model of a market. Then we describe various product adoption models (i.e., how customers decide which products to purchase) and the expected sales of a set of products given the current market condition (e.g., available products and customers). Finally, we formulate the market entry and market sustainability problems.

2.1 Model for a market

We consider a market which consists of l customers $\mathcal{C} = \{c_1, c_2, \ldots, c_l\}$ and there are m existing products $\mathcal{P}_E = \{p_1, p_2, \ldots, p_m\}$. Let M represent a manufacturer and let $\mathcal{P}_M \subseteq \mathcal{P}_E$ denote all the existing products produced by M which are in the market. The remaining products in the market, denoted by $\mathcal{P}_C = \mathcal{P}_E \setminus \mathcal{P}_M$, are from competitors of M, or $\mathcal{P}_E = \mathcal{P}_M \cup \mathcal{P}_C$, and $\mathcal{P}_M \cap \mathcal{P}_C = \emptyset$. Suppose the manufacturer M wants to produce some new products that will maximize its utility and it needs to take into account the current market condition. Manufacturer M has a budget constraint and it can only select $k \geq 1$ products from n candidate new products to produce, we denote these candidate new products as $\mathcal{P}_N = \{p_{m+1}, p_{m+2}, \ldots, p_{m+n}\}$. Note that all the products in $\mathcal{P}_N$ are *new* to the market, i.e., $\mathcal{P}_N \cap \mathcal{P}_E = \emptyset$. Formally, when $\mathcal{P}_N$ is given, the manufacturer M needs to select k most marketable products out from $\mathcal{P}_N$.

2.2 Models for Product Adoption

Each product in $\mathcal{P}_E \cup \mathcal{P}_N$ is associated with d attributes denoted by $\mathcal{A} = \{A_1, A_2, \ldots, A_d\}$. Each attribute is represented by a non-negative real number and higher value implies higher quality. We can use A_i to represent various attributes, e.g., durability, attractiveness, or inverse of price. The qualities of a product can be described by a d-dimensional vector. Specifically, the qualities of the product p_j are described by the vector $\boldsymbol{q}_j = (q_j[1], q_j[2], \ldots, q_j[d])$ where $q_j[t] \in [0, \infty)$ indicates the quality of p_j on the attribute A_t, $\forall t = 1, \ldots, d$. Similarly, the requirements of a customer can also be described by a d-dimensional vector. Let $\boldsymbol{r}_i = (r_i[1], r_i[2], \ldots, r_i[d])$ denote the requirement vector of customer c_i, where $r_i[t] \in [0, \infty)$ indicates the *minimum* requirement on attribute A_t, i.e., customer c_i requires that the product's quality on A_t is at least $r_i[t]$, or she will not adopt that product.

A customer may adopt a product only if the product satisfies her requirements. We say that a product satisfies a customer's requirements if and only if the product meets the requirements of that customer on *all* attributes. Formally, we define the following.

Definition 1 (**Product satisfiability**). *Consider a customer c_i and a product p_j. We say the product p_j satisfies the requirements of the customer c_i if and only if $q_j[t] \geq r_i[t]$, $\forall t = 1, \ldots, d$. We denote this relationship by $p_j \gtrsim c_i$, and p_j is said to be a satisfactory product of c_i. In other words, c_i is a potential customer of p_j.*

To illustrate, consider that each product has two attributes ($d=2$), and the value of each attribute ranges from 0 to 10. Let's say we have two existing products $\mathcal{P}_E = \{p_1, p_2\}$, two candidate products $\mathcal{P}_N = \{p_3, p_4\}$ and three customers $\mathcal{C} = \{c_1, c_2, c_3\}$. The qualities of the products and the requirements of the customers are depicted in Figure 1, where the horizontal axis represents the attribute A_1 and the vertical axis represents the attribute A_2. The satisfactory relationship is shown on the right of Figure 1. For example, the product p_1 has two potential customers c_1 and c_2, while the customer c_3 has one satisfactory product p_4.

Customers may adopt more than one product. To model this possibility, we let w_i, $w_i \in \mathbb{R}^+$, be the purchasing capacity of customer c_i. We assume that if c_i has some satisfactory products in the market, then c_i will use up all w_i units on these products, otherwise 0 units will be adopted.

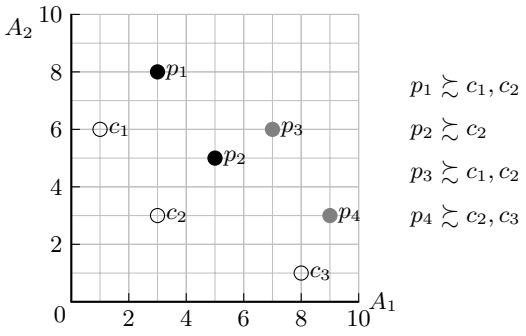


Figure 1: An illustration for product satisfiability and adoption

It is important to note that if c_i only has one satisfactory product, say p_j, then c_i will adopt w_i units of p_j, and if c_i has multiple satisfactory products, then these w_i units of purchased products will be a *combination* of all satisfactory products. Because customers may have different behavior in adopting different satisfactory products, let us present *three* models to describe some representative product adoption behavior: *persistent adoption model*, *opportunistic adoption model*, and *mixed adoption model*.

• Persistent Adoption Model

Customers may prefer some products over others. A customer's preference over a product can be described by the probability that the customer will adopt that product. Higher probability implies higher preference. Let $\Pr(i, j|\mathcal{P})$ denote the probability that a customer c_i adopts the product p_j when a set $\mathcal{P}$ of products are available in the market. Note that any change on other products may change the value of the probability $\Pr(i, j|\mathcal{P})$, since other products may become more (or less) attracted to the customer. The *relative preference* of the customer c_i over the products p_j and $p_{j'}$ can be expressed by the ratio $\Pr(i, j|\mathcal{P})/\Pr(i, j'|\mathcal{P})$. We say a customer's preference is *persistent* if given any two products, that customer's *relative preference* over these two products remains the same even when there are changes among available products in the market. Concretely, suppose a customer prefers the product p_j over the product $p_{j'}$, then she will always prefer p_j over $p_{j'}$. Mathematically, the ratio $\Pr(i, j|\mathcal{P})/\Pr(i, j'|\mathcal{P})$ is a constant, or

$$\frac{\Pr(i, j|\mathcal{P})}{\Pr(i, j'|\mathcal{P})} = \frac{\Pr(i, j|\mathcal{P}')}{\Pr(i, j'|\mathcal{P}')}, \quad \forall \mathcal{P}, \mathcal{P}'. \tag{1}$$

It is important to note that in the case of $\Pr(i, j'|\mathcal{P}) = 0$, the value of $\Pr(i, j|\mathcal{P})/\Pr(i, j'|\mathcal{P})$ is defined to be 1 if $\Pr(i, j|\mathcal{P}) = 0$, otherwise it is defined to be ∞.

Example 1. To illustrate, consider the products and customers depicted in Figure 1. Suppose the market only contains the existing products $\mathcal{P}_E = \{p_1, p_2\}$, one can observe that both of p_1 and p_2 are c_2's satisfactory products. Assume c_2 prefers p_1 over p_2 and c_2 will adopt p_1 and p_2 with probability 0.8 and 0.2 respectively, or $\Pr(2, 1|\mathcal{P}_E) = 0.8$, $\Pr(2, 2|\mathcal{P}_E) = 0.2$. Hence the "relative preference" of c_2 over p_1 and p_2 can be expressed as $\Pr(2, 1|\mathcal{P}_E)/\Pr(2, 2|\mathcal{P}_E) = 4$. Suppose now we introduce a new product p_3 into the market. From Figure 1, one can see that p_3 is also a satisfactory product of c_2. Let's say p_3 is so attracted to c_2 that c_2 will adopt it with probability 0.9. After p_3 is introduced into the market, the probability c_2 adopting p_1 and p_2 will change, but under the persistent adoption model, their ratio remains the same, say $\Pr(2, 1|\mathcal{P}_E \cup \{p_4\})/\Pr(2, 2|\mathcal{P}_E \cup \{p_4\}) = 4$. In other words, the relative preference of c_2 over p_1 and p_2 remains unchanged. Based on that ratio, we can update the probability c_2 adopts p_1 and p_2 as $\Pr(2, 1|\mathcal{P}_E \cup \{p_4\}) = 0.08$ and $\Pr(2, 2|\mathcal{P}_E \cup \{p_4\}) = 0.02$.

In this work, we consider two representative instances of the persistent adoption model: *uniform model* and *distance-proportional model*.

Uniform Model (UM)

The *uniform model* is one of the most widely used models which assume that when a customer has multiple satisfactory products, the customer will adopt them with equal probability. Mathematically,

$$\frac{\Pr(i, j|\mathcal{P})}{\Pr(i, j'|\mathcal{P})} = 1, \quad \forall p_j, p_{j'} \in \mathcal{SP}(c_i|\mathcal{P}), \tag{2}$$

where $\mathcal{P}$ is the set of products in the market, $\mathcal{SP}(c_i|\mathcal{P}) \subseteq \mathcal{P}$ is the set of c_i's satisfactory products in $\mathcal{P}$.

Distance-proportional Model (DM)

In a real world market, the higher a product's quality is, the more likely customers may adopt it. We use a distance measure between a customer and a product to quantify how likely a customer may adopt that product. Mathematically, a customer adopts a satisfactory product with probability proportional to the distance between her requirement vector and that product's quality vector. There are a number of distance metrics we can use, e.g., l_1 or l_2 norm. Let $\mathcal{P}$ be the available products in the current market, and $d(i, j)$ be the distance between the requirement vector of customer c_i and the quality vector of product p_j. Then we have

$$\frac{\Pr(i, j|\mathcal{P})}{\Pr(i, j'|\mathcal{P})} = \frac{d(i, j)}{d(i, j')}, \quad \forall p_j, p_{j'} \in \mathcal{SP}(c_i|\mathcal{P}). \tag{3}$$

• Opportunistic Adoption Model

We say a customer's preference is opportunistic if her relative preference over two given products may be influenced by the change of the market condition. We consider two representative instances of the *opportunistic adoption model: single-attribute-based model* and *all-attributes-based model*.

Single-attribute-based Model (SM)

In a realistic market, one typical behavior is if a customer has multiple satisfactory products, she will simply adopt the one with the *lowest* price. We describe this kind of behavior by the *SM*. Under the *SM*, customers always adopt a satisfactory product which has the highest quality on a particular attribute (say the inverse of the price). We call this attribute *decisive attribute*. If there are multiple products having the same highest quality on the customers' decisive attribute, then that customer will randomly select one. Formally, let a_i be the index of the decisive attribute of c_i. Let $\mathcal{P}$ be the set of products in the current market, S_i be the set of products which has the highest quality on attribute A_{a_i} among $\mathcal{SP}(c_i|\mathcal{P})$, then the choice of c_i can be modeled as follows.

$$\Pr(i, j|\mathcal{P}) = \begin{cases} \frac{1}{|S_i|} & \text{if } q_j[a_i] = \max\limits_{p_t \in \mathcal{SP}(c_i|\mathcal{P})} q_t[a_i], \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

All-attributes-based Model (AM)

Similar with the *DM*, customers following the *AM* also adopt products according to the distance between products' quality vector and their requirement vector. However, under the *AM*, customers only adopt the satisfactory product which has the *longest* distance (or highest quality among all attributes). If there are multiple products having the same longest distance, then the customers will randomly select one. Let $\mathcal{P}$ be the set of products in the current market, $d(i,j)$ be the distance between the requirement vector of customer c_i and the quality vector of product p_j. Let S'_i be the set of products which have the longest distance to c_i. Then the probability that the c_i adopts p_j can be modeled as follows.

$$\Pr(i,j|\mathcal{P}) = \begin{cases} \frac{1}{|S'_i|} & \text{if } d(i,j) = \max\limits_{p_t \in \mathcal{SP}(c_i|\mathcal{P})} d(i,t), \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

• **Mixed Adoption Model**

In the *mixed adoption model*, some customers follow the *persistent adoption model* while other customers follow the *opportunistic adoption model*. Let $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ denote the fraction of customers that follow the *UM*, *DM*, *SM*, and *AM* respectively, where $\sum_{i=1}^{4} \alpha_i = 1$. We call this the $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$-*Mixed Adoption Model*. To simplify notations, we denote $(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$-*Mixed Adoption Model* as the ***Mean-Mixed Adoption Model (MM)***.

Example 2. Consider the products and customers depicted in Figure 1. Suppose the market only contains the products in $\mathcal{P}_E \cup \{p_4\}$, where $\mathcal{P}_E = \{p_1, p_2\}$. One can observe that all these three products are satisfactory products of c_2. Let's consider the probability c_2 will adopt p_4. Under the *UM*, c_2 will adopt p_1, p_2 and p_4 with the same probability, or 1/3. Under the *DM*, assume that we use the l_1 norm to compute the distance. We have $d(2,1) = 5$, $d(2,2) = 4$, $d(2,4) = 6$. Hence, c_2 will adopt p_4 with probability $6/(5+4+6) = 2/5$. Under the *SM*, assume that A_2 is the *decisive attribute* of c_2. Since p_1 has the highest quality on the attribute A_2, c_2 will adopt p_1 with probability 1.0, p_2 and p_4 with probability 0. Under the *AM*, assume that we also use the l_1 norm to compute the distance. Since p_4 has the longest distance from c_2, c_2 will adopt p_4 with probability 1.0. Under the *MM*, if we still use the l_1 norm for all distance measure and $a_2 = 2$. Then c_2 will adopt p_4 with probability $\frac{1}{4}(1/3+2/5+0+1) \approx 0.43$.

2.3 Model for Product Sales

We formulate the expected sales of a set of products P. Given a product p_j, let $\mathcal{PC}(p_j) = \{c_i | c_i \in \mathcal{C}, p_j \succsim c_i\}$ be the set of potential customers of p_j. Let P be a set of products we consider and $\mathcal{P}_E$ is the set of existing products in the market. The expected sales of P are defined as:

$$Sale(P) = \sum_{p_j \in P} \sum_{c_i \in \mathcal{PC}(p_j)} w_i \cdot \Pr(i,j|\mathcal{P}_E \cup P). \quad (6)$$

Example 3. Let us illustrate the expected sales of p_4, or $Sale(\{p_4\})$, by considering the same scenario in Example 2 where all the customers follow the *MM*. In the market, p_4 satisfies c_2 and c_3, or $\mathcal{PC}(p_4) = \{c_2, c_3\}$. Let the weight of c_2 and c_3 be $w_2 = 1$ and $w_3 = 2$, respectively. Consider the customer c_2, from Example 2, we can see that c_2 will adopt p_4 with probability 0.43. Consider the customer c_3, p_4 is the only satisfactory product of c_3, so no matter what adoption model c_3 follows, she will adopt p_4 with probability 1.0. As a result, the total sales of p_4 are $w_2 \times 0.43 + w_3 \times 1.0 = 2.43$.

2.4 Problem Formulation

Consider a new manufacturer M who wants to enter the market. The goal of M is to attract as many customers as possible to adopt its new entrant products. Given its production budget, M seeks to select $k \geq 1$ products from the new product set $\mathcal{P}_N$ so to maximize the sales.

Since M is a new manufacturer, it has no existing product in the market, i.e., $\mathcal{P}_M = \emptyset$. The goal is to find k products from $\mathcal{P}_N$ which can achieve the highest expected sales. We call this market entrance as the *top-k best-selling products* (k-**BSP**) problem, which is formally defined as follows.

Definition 2 (k-**BSP**). *Given a set $\mathcal{C}$ of customers, a set $\mathcal{P}_E$ of existing products in the market, and a set $\mathcal{P}_N$ of candidate products by the manufacturer M, select k products from $\mathcal{P}_N$ so to maximize $Sale(P)$, where P is the set of k products the manufacturer M selects from $\mathcal{P}_N$.*

Now assume that the manufacturer M is in the market and has some products in $\mathcal{P}_E$. Then a natural question to ask is what new products M needs to produce so to sustain and maximize its business in the market? Note that this is different from the k-**BSP** problem since $\mathcal{P}_M \neq \emptyset$. M needs to consider not only what new products to select from P_N, but also how these new products may affect its existing products in the market. We call this the *top-k best-benefit products* (k-**BBP**) problem, and it can be defined as follows.

Definition 3 (k-**BBP**). *Given a set $\mathcal{C}$ of customers, a set $\mathcal{P}_E = \mathcal{P}_C \cup \mathcal{P}_M$ (with $\mathcal{P}_M \neq \emptyset$) of existing products in the market, and a set $\mathcal{P}_N$ of candidate products by the manufacturer M, select k products from $\mathcal{P}_N$ so to maximize $Sale(P \cup \mathcal{P}_M)$, where P is the set of the k products M selects.*

Note that both the k-**BSP** and k-**BBP** problems are functions of the product adoption models we defined earlier. In the following, we proceed to explore the algorithmic design and their computational complexity. In Section 3, we present the exact algorithms for the top-1 **BSP** and **BBP** problems which have polynomial running time. However, we formally prove that finding the exact solutions for k-**BSP** and k-**BBP** is *NP-hard* for $k > 1$. To tackle this challenge, in Section 4, we present efficient greedy-based approximation algorithms based on the top-1 exact algorithms. By proving the submodularity of the sales function $Sale(\cdot)$, we formally prove that our approximation algorithms can provide a high performance guarantee on the quality of the solutions.

3. EXACT ALGORITHMS AND THEIR HARDNESS

In this section, we first present the exact algorithms for the top-1 **BSP** and **BBP** problems under the three adoption models introduced in Section 2. This serves as the foundation of our approximation algorithms in Section 4. Then we provide the formal proof of the *NP-hardness* of the k-**BSP** and k-**BBP** problems, for $k > 1$.

3.1 Top-1 Exact Algorithms

One way to find the exact solutions of the top-1 **BSP** and **BBP** problems is via exhaustive search: for all candidate products in $\mathcal{P}_N$, calculate the expected sales and select

the one with the largest. The computational complexity of exhaustive search is $O(mnld)$, where $m = |\mathcal{P}_E|$, $n = |\mathcal{P}_N|$, $l = |\mathcal{C}|$, and $d = |\mathcal{A}|$. In the following, we will propose enhanced top-1 exact algorithms with lower computational complexity, $O((m+n)ld)$, for the three adoption models we introduced in Section 2.

• **Persistent Adoption Model**

Let us first assume that all customers follow the *persistent adoption model*. Let $\mathcal{P}_j$ denote the set of products $\mathcal{P}_E \cup \{p_j\}$. According to the definition of persistent adoption model, we can calculate the sales of product p_j as follows.

$$Sale(\{p_j\}) = \sum_{c_i \in \mathcal{PC}(p_j)} w_i \cdot (\Pr(i,j|\mathcal{P}_j) / \sum_{p_t \in \mathcal{SP}(c_i|\mathcal{P}_j)} \Pr(i,t|\mathcal{P}_j)). \quad (7)$$

Recall that the ratio $\Pr(i,j|\mathcal{P}_j) : \Pr(i,j'|\mathcal{P}_j)$ will not change and all products we consider are in the set $\mathcal{P}_E \cup \mathcal{P}_N$. Let $b_{i,j} = \Pr(i,j|\mathcal{P}_E \cup \mathcal{P}_N)$, thus for any $p_j \in \mathcal{P}_N$, we have

$$Sale(\{p_j\}) = \sum_{c_i \in \mathcal{PC}(p_j)} w_i \cdot (b_{i,j} / \sum_{p_t \in \mathcal{SP}(c_i|\mathcal{P}_j)} b_{i,j}). \quad (8)$$

Let us consider the top-1 **BSP** problem first. The main idea is that we first calculate b_{ij} for each pair of c_i and p_j. Then for each new product $p_j \in \mathcal{P}_N$ we calculate its expected sales according to Equation (8). The product with the largest expected sales is the solution of the algorithm. The main idea of finding the top-1 **BBP** product is similar with finding top-1 **BSP** product. Finally, we outline the algorithm for finding the top-1 **BSP** or **BBP** product in Algorithm 1. Note that the algorithm is a general algorithm for all *persistent adoption models*, and it can be simplified for specific models. For the *UM*, we can replace line 4 by $b_{ij} \leftarrow 1$, and for the *DM*, by $b_{ij} \leftarrow d(i,j)$.

Algorithm 1 Top-1 Algorithm (*persistent adoption model*)

1: $Sale(\mathcal{P}_M) \leftarrow 0$
2: **for** all $c_i \in \mathcal{C}$ **do**
3: **for** all $p_j \in \mathcal{SP}(c_i|\mathcal{P}_E \cup \mathcal{P}_N)$ **do**
4: $b_{ij} \leftarrow \Pr(i,j|\mathcal{P}_E \cup \mathcal{P}_N)$
5: $sum_i = \sum_{p_j \in \mathcal{P}_E} b_{ij}$
6: $m_i = \sum_{p_j \in \mathcal{P}_M} b_{ij}$
7: $Sale(\mathcal{P}_M) \leftarrow Sale(\mathcal{P}_M) + w_i \cdot \frac{m_i}{sum_i}$
8: $\max Sale \leftarrow 0$
9: **for** all $p_j \in \mathcal{P}_N$ **do**
10: $Sale(p_j) \leftarrow Sale(\mathcal{P}_M)$
11: **for** $c_i \in \mathcal{PC}(p_j)$ **do**
12: $Sale(p_j) \leftarrow Sale(p_j) + w_i \cdot (\frac{m_i + b_{ij}}{sum_i + b_{ij}} - \frac{m_i}{sum_i})$
13: **if** $Sale(p_j) \geq \max Sale$ **then**
14: $res \leftarrow p_j$
15: $\max Sale \leftarrow Sale(p_j)$
16: **return** *res*

Lemma 1 (**Computational complexity**). *The computational complexity of Algorithm 1 is $O((m+n)ld)$ for both* **BSP** *and* **BBP**, *where $m = |\mathcal{P}_E|$, $n = |\mathcal{P}_N|$, $l = |\mathcal{C}|$, $d = |\mathcal{A}|$.*

Proof. From line 2 to 7, we compute b_{ij} for each pair of c_i and p_j, and calculate $Sale(\mathcal{P}_M)$ at the same time. The complexity of these steps is $O((m+n)ld)$. From line 9 to 15, for each product in $\mathcal{P}_N$, it takes $O(ld)$ time to compute the expected sales. Hence, the total computational complexity is $O((m+n)ld) + n \times O(ld) = O((m+n)ld)$. ∎

• **Opportunistic Adoption Model**

Let us present the algorithmic design for the *SM* and the *AM*. The algorithm for finding top-1 **BSP** or **BBP** product corresponding to *AM* is outlined in Algorithm 2. The main idea is that, for each customer, we find the products which have the longest distance from her and record the distance. Note that it may happen that multiple products have the longest distance. We also record the customers who will adopt products in $\mathcal{P}_M$ during the calculation. Then, for each product in $\mathcal{P}_N$, we calculate the expected sales via the data we have recorded. The product with the largest sales will be returned as the solution of the algorithm. The algorithm for finding the top-1 **BSP** or **BBP** product corresponding to the *SM* is similar with that of the *AM*. Please refer to [5] for details.

Lemma 2 (**Computational complexity**). *The computational complexity of Algorithm 2 is $O((m+n)ld)$ for both* **BSP** *and* **BBP**, *where $m = |\mathcal{P}_E|$, $n = |\mathcal{P}_N|$, $l = |\mathcal{C}|$, $d = |\mathcal{A}|$.*

Proof. Similar with the proof of Lemma 1, one can observe that the algorithm takes $O(mld)$ time to do the precalculation and $O(nld)$ time to select the product with maximum expected sales. Hence, the total computational complexity is $O((m+n)ld)$. ∎

Algorithm 2 Top-1 Algorithm (*AM*)

1: $Sale(\mathcal{P}_M) \leftarrow 0$
2: **for** all $c_i \in \mathcal{C}$ **do**
3: Find the set of products S'_i which have the longest distance from c_i in $\mathcal{SP}(c_i|\mathcal{P}_E)$
4: $d_i \leftarrow$ distance between the products in S'_i and c_i
5: $e_i \leftarrow |S'_i|$
6: $m_i \leftarrow |S'_i \cap \mathcal{P}_M|$
7: $Sale(\mathcal{P}_M) \leftarrow Sale(\mathcal{P}_M) + w_i \cdot \frac{m_i}{e_i}$
8: $\max Sale \leftarrow 0$
9: **for** all $p_j \in \mathcal{P}_N$ **do**
10: $Sale(p_j) \leftarrow Sale(\mathcal{P}_M)$
11: **for** all $c_i \in \mathcal{PC}(p_j)$ **do**
12: **if** $d(i,j) > d_i$ **then**
13: $Sale(p_j) \leftarrow Sale(p_j) + w_i \cdot (1 - \frac{m_i}{e_i})$
14: **else if** $d(i,j) = d_i$ **then**
15: $Sale(p_j) \leftarrow Sale(p_j) + w_i \cdot (\frac{m_i+1}{e_i+1} - \frac{m_i}{e_i})$
16: **if** $Sale(p_j) \geq \max Sale$ **then**
17: $res \leftarrow p_j$
18: $\max Sale \leftarrow Sale(p_j)$
19: **return** *res*

• **Mixed Adoption Model**

Suppose that customers follow the $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$-*mixed adoption model*. When we select products with the maximum sales, we need to calculate the sales corresponding to all of the four models for each product in $\mathcal{P}_N$. Let $sale_1$, $sale_2$, $sale_3$, and $sale_4$ denote the expected sales corresponding to the *UM*, *DM*, *SM*, and *AM*. The total sales will be $\sum_{i=1}^{4} \alpha_i \cdot sale_i(P)$. So the algorithm corresponding to the *mixed adoption model* will be a combination of the algorithms corresponding to all of the four models. Hence,

according to Lemma 1 and 2, under $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$-*mixed adoption model*, the computational complexity of the algorithm for finding the top-1 **BSP** or **BBP** product is also $O((m+n)ld)$, where $m=|\mathcal{P}_E|$, $n=|\mathcal{P}_N|$, $l=|\mathcal{C}|$, and $d=|\mathcal{A}|$.

3.2 Top-k Exact Algorithms

A direct yet naive approach to find the exact solutions of the k-**BSP** and k-**BBP** problems is via exhaustive search: Enumerate all possible subsets of size k from $\mathcal{P}_N$, then calculate the expected sales of each subset and select the one with the largest. However, this approach is not scalable since there are exponentially many possible subsets when $k > 1$. The following two theorems state that finding the exact solution for the k-**BSP** and k-**BBP** problems is *NP-hard* when the number of attributes is three or more, respectively.

Theorem 1. *Finding the exact solution for the k-**BSP** problem is NP-hard when the number of attributes $d \geq 3$.*

Proof. Please refer to [5] for details. ∎

Theorem 2. *Finding the exact solution for the k-**BBP** problem is NP-hard when the number of attributes $d \geq 3$.*

Proof. Please refer to [5] for details. ∎

4. DESIGN OF APPROXIMATION ALGORITHMS

In the last section, we formally showed that finding out the exact solutions for the k-**BSP** problem and the k-**BBP** problem is *NP-hard*. However, we also presented efficient exact algorithms for the top-1 **BSP** and **BBP** problems. In this section, we extend the top-1 algorithms to the top-k problems by a *greedy-based approximation algorithmic framework*. By proving the submodularity of the sales function $Sale(\cdot)$, we show that our approximation algorithms are not only computationally efficient, but can also provide a high theoretical performance guarantee on the quality of the solutions: $(1-1/e)$-approximation.

4.1 Greedy-based Approximation Algorithms

Let us present a *general greedy algorithmic framework* to solve the k-**BSP** and k-**BBP** problems. The main idea can be described as follows. We select k products in k steps: in each step we select the product which is the solution of the top-1 exact algorithm of the corresponding adoption model (proposed in Section 3), then we remove this product from $\mathcal{P}_N$ and add it into $\mathcal{P}_M$. We outline the framework in Algorithm 3. According to Lemma 1 and 2, the computational complexity of Algorithm 3 is $O(k(m+n)ld)$.

Algorithm 3 Top-k Greedy Algorithm

```
1: P ← ∅
2: while |P| < k do
3:     p_new ← top-1 exact algorithm
4:     P ← P ∪ {p_new}
5:     P_M ← P_M ∪ {p_new}
6:     P_N ← P_N \ {p_new}
7: return P
```

In the following, we first introduce the notion of *submodularity* [12] and one of its interesting properties: the greedy-based framework on a submodular objective function can provide a performance guarantee on the quality of the solution as compared to the optimal one [6].

Given a finite set U, consider a real-valued set function $f: 2^U \to R$, where 2^U denotes the power set of U. We say f is submodular if for any $S \subseteq U$, the marginal gain of adding an element to S is at least as high as the marginal gain of adding the same element to a superset of S. Formally, the submodular set function is defined as follows.

Definition 4 (**Submodular Set Function**). *Given a finite ground set U, a function f that maps subsets of U to real numbers is called* submodular *if*

$$f(S\cup\{u\})-f(S) \geq f(T\cup\{u\})-f(T),\ \forall S\subseteq T\subseteq U, u\in U. \quad (9)$$

Submodular set functions have numbers of interesting properties. One of them is shown in Theorem 3 and we use it to design approximation algorithms with theoretical performance guarantee.

Theorem 3. *For a non-negative monotone submodular function $f: 2^U \to R$, let $S \subseteq U$ be the set of size k obtained by selecting elements from U one at a time, each time choosing the element that provides the largest marginal increase in the function value. Let $S^* \subseteq U$ be the set that maximizes the value of f over all k-element sets. Then we have $f(S) \geq (1–1/e)\cdot f(S^*)$, where e is the base of the natural logarithm. In other words, S provides a $(1–1/e)$-approximation, or guarantees a lower bound on the quality of solutions as compared to the optimal solutions.*

Applying it to the k-**BSP** and k-**BBP** problems, the ground set is $\mathcal{P}_M \cup \mathcal{P}_N$, the sales function $Sale(\cdot)$ in Equation (6) maps subsects of $\mathcal{P}_M \cup \mathcal{P}_N$ to real numbers. To show our greedy algorithms can provide a $(1–1/e)$-approximation according to Theorem 3, we need to prove that $Sale(\cdot)$ is non-negative monotone submodular, which is shown in Theorem 5 and 6 in the next subsection. Once we prove these properties of $Sale(\cdot)$, we have the following theorem.

Theorem 4 (**Performance guarantee**). *The greedy algorithmic framework as stated in Algorithm 3 provides at least $(1–1/e)$-approximate solutions compared with the optimal ones, where e is the base of the natural logarithm.*

Proof. According to Theorem 5 and 6, the sales function $Sale(\cdot)$ in Equation (6) is non-negative, monotone submodular. According to Theorem 3, Algorithm 3 provides $(1–1/e)$-approximate solutions. ∎

4.2 Submodularity Analysis

Since the sales function $Sale(\cdot)$ defined in Equation (6) is obviously non-negative, we seek to prove that it is monotone submodular for the k-**BSP** and k-**BBP** problems. In particular, we show the monotonicity and submodularity properties holds for the three adoption models we introduced in Section 2: the *persistent adoption model*, the *opportunistic adoption model* and the *mixed adoption model*.

• Analysis for the Persistent Adoption Model

Consider the case where all customers adopt products following the *persistent adoption model*. To prove the sales function $Sale(\cdot)$ is monotone submodular, we first need to

prove some lemmas (as shown in Lemma 3, 4 and 5, and the proof of them please refer to [5]). Based on these lemmas, we can then prove the monotonicity and submodularity, which are stated in Theorem 5 and 6, respectively.

To simplify the expression, we define the following notations. For any set $S \subseteq \mathcal{P}_M \cup \mathcal{P}_N$ of products, let $\mathcal{P}_S = \mathcal{P}_E \cup S$ and $S_j = S \cup \{p_j\}$. Furthermore, let $pr_i(S) = \sum_{p_j \in S} \Pr(i, j|\mathcal{P}_S)$ denote the probability of a customer c_i adopting products in S when a set $\mathcal{P}_S$ of products are available in the market.

Lemma 3. *Let $\mathcal{P}_S$ be the set of products available in the market, by adding another new product p_u into the market, $p_u \in \mathcal{P}_N \setminus \mathcal{P}_S$, the increase of the sales of products in S_u is*

$$Sale(S_u) - Sale(S) = \sum_{c_i \in \mathcal{PC}(p_u)} w_i \cdot (pr_i(S_u) - pr_i(S)). \quad (10)$$

Lemma 4. *Let S and T be two sets of products, $S \subseteq T \subseteq \mathcal{P}_M \cup \mathcal{P}_N$, and p_u be another product in $\mathcal{P}_N$, $p_u \in \mathcal{P}_N \setminus T$. For a customer c_i following the persistent adoption model, if $p_u \succsim c_i$, i.e., $c_i \in \mathcal{PC}(p_u)$, then we have*

$$pr_i(S_u) - pr_i(S) \geq pr_i(T_u) - pr_i(T). \quad (11)$$

Lemma 5. *Let $S \subseteq \mathcal{P}_M \cup \mathcal{P}_N$ be a set of products, and p_u be another product in $\mathcal{P}_N$, $p_u \in \mathcal{P}_N \setminus S$. For a customer c_i following the persistent adoption model„ if $p_u \succsim c_i$, i.e., $c_i \in \mathcal{PC}(p_u)$, then we have*

$$pr_i(S_u) - pr_i(S) \geq 0. \quad (12)$$

Based on Lemma 3, 4 and 5, we now prove the monotonicity and submodularity of the sales function $Sale(\cdot)$ as follows in Theorem 5 and 6, respectively.

Theorem 5. *Suppose all customers adopt products following the persistent adoption model, then the sales function $Sale(\cdot)$ defined in Equation (6) is monotone for the k-**BSP** problem and the k-**BBP** problem.*

Proof. Consider the k-**BSP** problem. To prove the monotonicity property of the sales function, we need to show

$$Sale(S_u) - Sale(S) \geq 0 \quad (13)$$

holds, $\forall S \subseteq \mathcal{P}_N$, $p_u \in \mathcal{P}_N$. Similarly, for the k-**BBP** problem, we need to show

$$Sale(\mathcal{P}_M \cup S_u) - Sale(\mathcal{P}_M \cup S) \geq 0 \quad (14)$$

holds, $\forall S \subseteq \mathcal{P}_N$, $p_u \in \mathcal{P}_N$. Recall that $\mathcal{P}_M \subseteq \mathcal{P}_E$. Examine Inequality (13) and (14), one can observe that to prove Inequality (13) and (14) hold, we only need to prove that Inequality (13) holds for all $S \subseteq \mathcal{P}_N \cup \mathcal{P}_M$ and $p_u \in \mathcal{P}_N$, which can be easily proved by combining the results of Lemma 3 and Lemma 5. ∎

Theorem 6. *Suppose all customers adopt products following the persistent adoption model, then the sales function $Sale(\cdot)$ defined in Equation (6) is submodular for the k-**BSP** problem and the k-**BBP** problem.*

Proof. Consider the k-**BSP** problem. Based on Definition 4, we need to show

$$Sale(S_u) - Sale(S) \geq Sale(T_u) - Sale(T) \quad (15)$$

holds, $\forall S \subseteq T \subseteq \mathcal{P}_N$, $p_u \in \mathcal{P}_N$.

Similarly, for the k-**BBP** problem, we need to show

$$Sale(\mathcal{P}_M \cup S_u) - Sale(\mathcal{P}_M \cup S) \geq Sale(\mathcal{P}_M \cup T_u) - Sale(\mathcal{P}_M \cup T) \quad (16)$$

holds, $\forall S \subseteq T \subseteq \mathcal{P}_N$, $p_u \in \mathcal{P}_N$. Note that $\mathcal{P}_M \subseteq \mathcal{P}_E$. When we examine Inequality (15) and (16), one can observe that it is sufficient to prove Inequality (15) holds for all $S \subseteq T \subseteq \mathcal{P}_N \cup \mathcal{P}_M$ and $p_u \in \mathcal{P}_N$.

In the case of $p_u \in S$, Inequality (15) holds, since both sides of Inequality (15) are equal to 0. In the case of $p_u \in T \setminus S$, the right side of Inequality (15) equals 0, while according to the monotonicity, which has been proved in Theorem 5, the left side is non-negative. Hence Inequality (15) also holds. In the case of $p_u \in \mathcal{P}_N \setminus T$, Inequality (15) can be easily proved by combining the results of Lemma 3 and 4. Thus, Inequality (15) holds $\forall S \subseteq T \subseteq \mathcal{P}_N \cup \mathcal{P}_M$, $p_u \in \mathcal{P}_N$. ∎

• Analysis for the Opportunistic Adoption Model

Here we consider the case where all customers adopt products following the *opportunistic adoption model*. The sales function $Sale(\cdot)$, in general, is not monotone submodular under the *opportunistic adoption model*. However, we will show that for the two adoption models we introduced in Section 2: the *SM* and the *AM*, the sales function $Sale(\cdot)$ is indeed monotone submodular, as shown in Theorem 7.

Theorem 7. *Suppose all customers adopt products following the SM or AM, then the sales function $Sale(\cdot)$ defined in Equation (6) is monotone submodular for both k-**BSP** and k-**BBP** problems.*

Proof. The derivation is similar to Theorem 5 and 6. Please refer to [5] for derivation details. ∎

• Analysis for Mixed Adoption Model

The *mixed adoption model* describes that in a market, some customers may follow the *persistent adoption model*, while the rest of them may follow the *opportunistic adoption model*. We show that the sales function $Sale(\cdot)$ remains monotone submodular in the following theorem.

Theorem 8. *Suppose customers adopt products follow the $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$-mixed adoption model, then the sales function $Sale(\cdot)$ defined in Equation (6) is monotone submodular for the k-**BSP** problem and the k-**BBP** problem.*

Proof. The derivation is based on Theorem 5, 6 and 7. Please refer to [5] for derivation details. ∎

5. EXPERIMENTS ON SYNTHETIC DATA

In this section, we perform experiments on synthetic data to show the efficiency and accuracy of our greedy algorithms. All the algorithms were implemented in C++ and the experiments were performed on a PC with a 16-core 2.4GHz CPU, 30 GB of main memory under 64-bit Debian 6.06.

• Synthetic datasets

We adopt one of the most widely used data generator provided by [3] to generate synthetic datasets. We generate the following three typical types of synthetic datasets to examine the efficiency and accuracy of algorithms.

- **Independent (*ind*)**: the value of each attribute is generated independently using a uniform distribution.
- **Positive correlated (*p-corr*)**: products (customers) that have high quality (requirement) in one attribute tends to have high qualities (requirements) in other attributes.
- **Negative correlated (*n-corr*)**: products (customers) that have high quality (requirement) in one attribute tends to have low qualities (requirements) in at least one attribute.

We set the weight of each customer to be 1.0. In other words, each customer only adopts one unit of products. The list of all the parameters used to generate the datasets are shown in Table 1.

In the following, we examine the impact of various factors. In each experiment, we examine one factor considering the corresponding parameter as a variable and setting other parameters as default value as shown in Table 1. Due to the page limit, we only show the results of the experiments on the first five parameters in Table 1. Please refer to [5] for experiments on the remaining three parameters. We examine the accuracy of our greedy algorithms (*greedy*) by comparing the output with the exhaustive search algorithms (*exh*). Furthermore, we also compare the running time and computationally efficiency of these two algorithms.

Figure 2-6 show the results. In each figure, the horizontal axis shows the corresponding parameter which is considered as a variable, the vertical axis of (a) shows the speedup of our greedy algorithms compared with the exhaustive search algorithms, i.e., the ratio between the running time of these two algorithms, and the vertical axis of (b) shows the expected sales. In Table 2, we show the running time of both algorithms. Since the running time only changes slightly when we change the first three parameters, we only show the running time of the experiments varying k and $|\mathcal{P}_N|$.

Parameters	Range	Default
data distributions	*ind, p-corr, n-corr*	*n-corr*
adoption models	*UM, DM, SM, AM, MM*	*DM*
$\|\mathcal{A}\|$	4, 6, 8, 10	4
k	2, 3, 4, 5	3
$\|\mathcal{P}_N\|$	20, 40, 60, 80	20
$\|\mathcal{P}_E\|$	100, 200, 300, 400	100
$\|\mathcal{P}_M\|$	5, 10, 15, 20	5
$\|\mathcal{C}\|$	1K, 4K, 7K, 10K	1K

Table 1: Parameters of synthetic data

• Experiment 1: Impact of data distribution

We explore the impact of different data distributions on the efficiency and accuracy of our greedy algorithms. In particular, we perform experiments on the datasets, where products' quality vectors and customers' requirement vectors are generated by three distributions: independent, positive correlated, and negative correlated. We run experiments for both the k-**BSP** and k-**BBP** problems, and the speedup and expected sales are shown in Figure 2. Figure 2(a) shows that the running time of the greedy algorithms is *significantly less* (about 1/1400) than that of the exhaustive search algorithms for all three different data distributions, and the speedup of the k-**BSP** problem and the k-**BBP** problem is almost the same. From Figure 2(b), we can see that for the independent and positive correlated data distributions, the greedy algorithms can almost find the optimal solutions since the expected sales corresponding to the greedy algorithms and the exhaustive search algorithms are nearly the same. This implies high accuracy of our greedy algorithms. For the negative correlated data distribution, the expected sales corresponding to the greedy algorithms are slightly smaller than the optimal ones (about 0.9-approximate), but still much better than the theoretical lower bound derived in Theorem 3.

Lessons learned: The greedy algorithms are significantly faster than the exhaustive search algorithms with high accuracy. The efficiency and accuracy of the greedy algorithms are *insensitive* to data distributions.

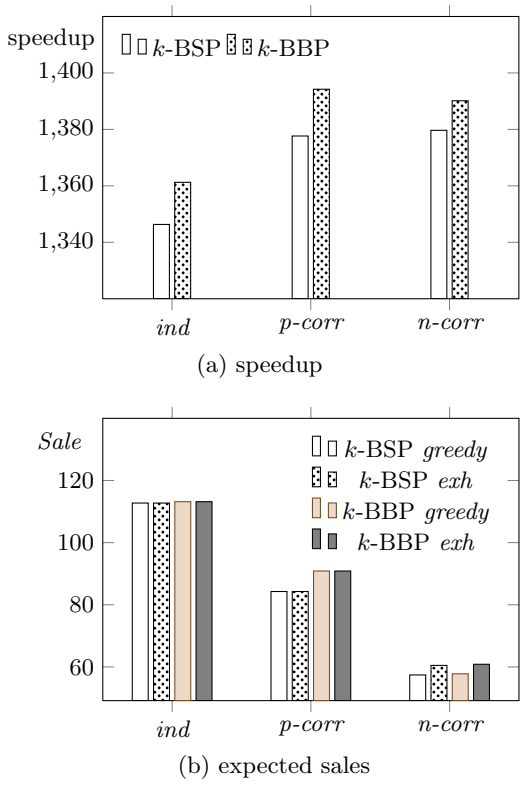
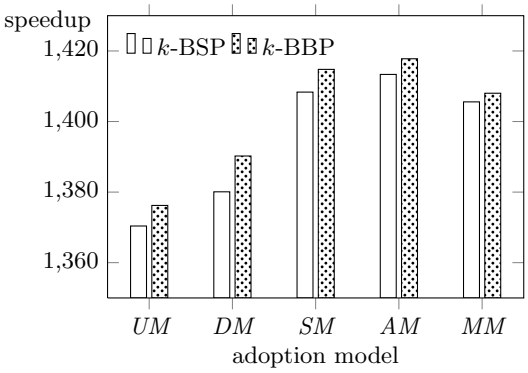


Figure 2: Impact of data distribution

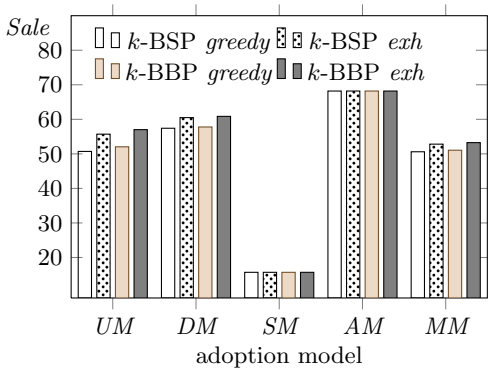
In the real world market, negative correlated distribution is common. Due to the constraint of technology and cost, most products cannot have good qualities on all attributes. Instead, they can preserve high qualities on some attributes but low qualities on some other attributes. At least, if a product is of high qualities, then the price tends to be high, which is not "good" for customers. Hence, in the following experiments, we will set negative correlated distribution as our default data distribution.

• Experiment 2: Impact of adoption models

Here we examine the impact of different adoption models as presented in Section 2 on the efficiency and accuracy of the greedy algorithms. We consider five different product adoption models: the *UM*, *DM*, *SM*, *AM*, and *MM*. Here, we use the l_1 norm to compute the distance in the *DM*, *AM*, and *MM*. We run experiments for both the k-**BSP** and k-



(a) speedup



(b) expected sales

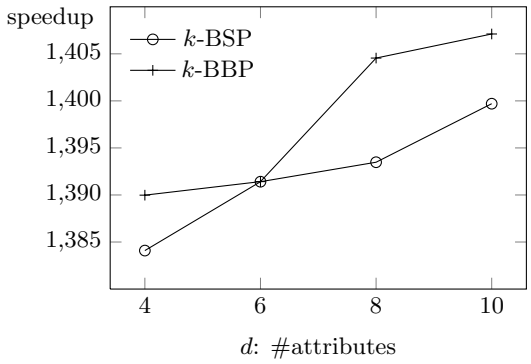
Figure 3: Impact of adoption models

BBP problems. The speedup and expected sales are shown in Figure 3. From Figure 3(a), we observe that the greedy algorithms are *significantly faster* (about 1,400 times faster) than the exhaustive search algorithms for all adoption models. Actually, for the greedy algorithms, the running time is invariant with respect to the product adoption models except the *MM*. The running time of the greedy algorithms under the *MM* is roughly the summation of that under the *UM*, *DM*, *SM*, and *AM*. Because to compute the corresponding sales under *MM*, we need to calculate the corresponding sales under all the other four models. Furthermore, the running time is nearly the same for the k-**BSP** and k-**BBP** problems. This statement also holds for the exhaustive search algorithms. From Figure 3(b), one can observe that the expected sales corresponding to the greedy algorithms and the exhaustive search algorithms are nearly the same. This implies a high accuracy of our greedy algorithms. As we vary the adoption models, the expected sales vary significantly. This shows that the adoption models have a high impact on the expected sales.

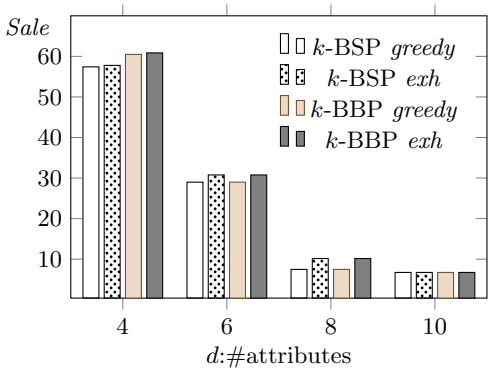
Lessons learned: Our greedy algorithms are remarkably faster than the exhaustive search algorithms and the speedup is invariant with respect to the adoption models. The accuracy of the greedy algorithms is very high and it is insensitive to adoption models. The optimal expected sales, however, are sensitive to adoption models. Customers following different adoption models may cause significant loss or increase in the expected sales.

• **Experiment 3: Impact of $|\mathcal{A}|$ (or d)**

We explore the impact of the number of attributes, or d, on the efficiency and accuracy of the greedy algorithms. We vary the value of d from 4 to 10. We show the speedup and expected sales in Figure 4. From Figure 4(a), one can observe that the speedup increases slightly with the increase of the number of attributes, which ranges from 1380 to 1410. From Figure 4(b), one can observe that the greedy algorithms are with similar high accuracy for each value of d. This implies that the accuracy is insensitive to the number of attributes d. It is interesting to observe that increasing the number of attributes decreases the expected sales. Because larger number of attributes indicates stronger customers' requirements, thus less products will satisfy the requirements of customers.



(a) speedup



(b) expected sales

Figure 4: Impact of d

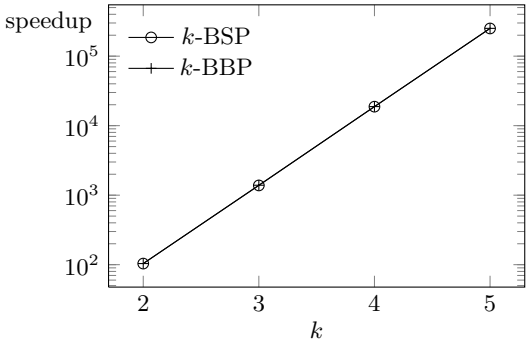
Lessons learned: The running time of both greedy algorithms and exhaustive search algorithms is invariant on d. This statement also holds for the accuracy. Increasing the number of attributes may decreases the expected sales.

• **Experiment 4: Impact of k**

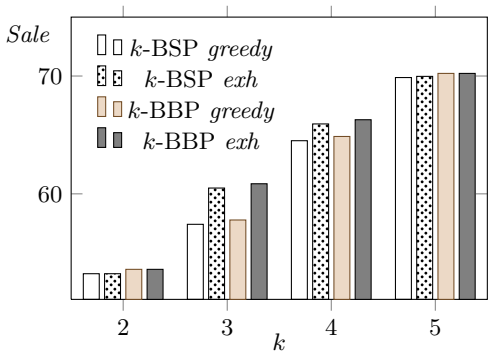
We explore the impact of the number of new products we select, or k, on the efficiency and accuracy of the greedy algorithms. Since the exhaustive search algorithms calculate the expected sales of all k-cardinality subsets of $\mathcal{P}_N$, the increase of k leads to an *exponential increase* in the running time. Due to this computational constraint, we vary k from 2 to 5. We run experiments for both the k-**BSP** and k-**BBP** problems. The speedup and expected sales are shown

in Figure 5. From Figure 5(a), we can see that when $k=5$, the speedup of our greedy algorithms compared with the exhaustive search algorithms is higher than 250,000, where the running time of the exhaustive search is more than 10^4 seconds, but our greedy algorithms still take less than 0.1 seconds as shown in Table 2. As we increase the value of k, we increase the running time of the exhaustive search algorithms remarkably, but the running time of the greedy algorithms only increases slightly. This implies that the greedy algorithms are highly efficient and can be applied to large data sets. Examine Figure 5(b), one can observe that the greedy algorithms provide at least a 0.9-approximation. This implies high accuracy of the greedy algorithms.

Lessons learned: Increasing the number of new products only increases the running time of the greedy algorithms slightly. The greedy algorithms are of high accuracy providing at least a 0.9-approximation. And the accuracy is insensitive to the number of new products k.



(a) speedup

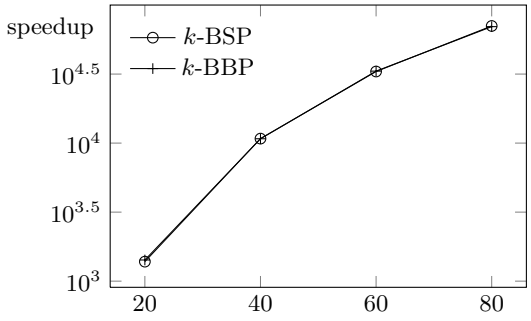


(b) expected sales

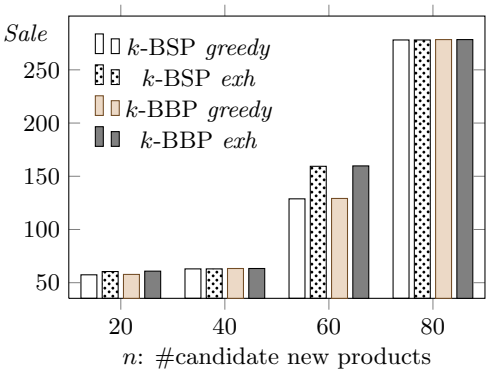
Figure 5: Impact of k

• **Experiment 5: Impact of $|\mathcal{P}_N|$ (or n)**

We explore the impact of the number of candidate products, say n, on the efficiency and accuracy of the greedy algorithms. We vary the number of candidate products n from 20 to 80. We show the speedup and the expected sales in Figure 6. From Figure 6(a), we can observe that as we increase the value of n, the speedup *increases exponentially*, since the running time of the exhaustive search increases exponentially while that of our greedy algorithms only increases linearly with n, as shown in Table 2. From Figure 6(b), we can see that with the increase of n, the greedy algorithms maintain a high level of quality guarantee, or above 0.8-approximation in the worst case.



(a) speedup



(b) expected sales

Figure 6: Impact of n

Lessons learned: The running time of the greedy algorithms increases linearly with the increase of the number of candidate products n. The accuracy of the greedy algorithms is invariant with respect to n.

k	**2**	**3**	**4**	**5**
greedy (s)	0.0223	0.0304	0.0384	0.0465
exh (s)	2.3132	41.9621	720.2301	11632.5061
n	**20**	**40**	**60**	**80**
greedy (s)	0.0303	0.0342	0.0382	0.0433
exh (s)	41.9425	368.5041	1260.4245	3033.1006

Table 2: Running time k-**BSP**

• **Summary and discussion**

For all the above experiments, the greedy algorithms show efficient running time and high accuracy for both of the k-**BSP** and k-**BBP** problems. Our greedy algorithms provide a 0.96-approximation ratio on average, and they are about 1400 times faster than the exhaustive search algorithms under default the parameters in our experiments. When $k=5$, the speedup is around 250,000 as compared to the exhaustive search algorithms. For larger k, say $k=6$ or 7, the exhaustive search algorithms need days or even months, while the running time of the greedy algorithms is still less than 0.1 seconds. This implies that our greedy algorithms can work effectively on large datasets.

6. EXPERIMENTS ON WEB DATA

In this section, we conduct the experiments on real-world web data to show the importance of studying the customers' adoption behavior: different adoption models may lead to totally different product selection strategies. Since the results of the k-**BSP** and k-**BBP** problems are similar, for brevity, we only show experiments on the k-**BSP** problem.

We consider the following problem: Assume customers' requirements, existing products, candidate new products, and k are all given, explore how different adoption models may influence the result of the k-**BSP** problem. Studying this problem helps us gain important insights on how to perform product selection in the real world.

• Web dataset

RateBeer.com *RateBeer* is widely recognized as the most accurate and most-visited source for beer information. Customers share their opinions on beers via expressing ratings on six attributes (e.g., aroma, appearance, taste). We crawl the historical ratings of 357 beers. Table 3 shows the overall statistics of the dataset.

dataset	#products	#customers	#attributes
RateBeer	357	5582	6

Table 3: Parameters of *RateBeer* dataset

• Extract attribute quality and requirement

For a product (beer) in the dataset, we use the *average rating* of an attribute as the true quality of that attribute. A customer's requirement on an attribute is set as the lowest rating she has ever expressed on that attribute. And the weight of each customer is set to be 1. In the dataset, the rating scale on different attributes is different, thus we perform normalization so all attributes are in the same scale, say in the range of $(0, 10)$. *RateBeer* does not provide ratings on attributes like "price", which is important for product adoption. To overcome this deficiency, we manually generate one attribute to represent the *price*. We generate the values on *price* based on negative correlation on other attributes, i.e., the higher the quality on other attributes, the lower quality on *price*. Precisely, the value of the price attribute is inversely proportional to the of sum of values on other attributes. Note that lower quality on *price* implies higher pricing. We also normalize the *price* within the range of $(0, 10)$. Since we cannot get the information about customers' requirements on price, we randomly generate them from 0 to 10.

• Experiments based on *RateBeer*

We select 10 beers from the 357 beers as the candidate new products (or $\mathcal{P}_N$), and we set the remaining 347 beers as the existing products (or P_E). The qualities of the 10 selected beers are shown in Table 4. We use our proposed algorithms to solve the 3-**BSP** problem under 5 different adoption models respectively: the *UM*, *DM*, *SM*, *AM*, and *MM* as defined in Section 2. We adopt l_1 norm to compute the distance and the decisive attribute of all customers is *taste*. The selected beers for each product adoption model are shown in Table 5, which depicts the adoption models, and the corresponding selected products. All the beers are anonymized and represented by ID numbers. Examine Table 5, one can observe that when we use different product adoption models, the selected beers vary significantly. For example, for the *UM*, the selected products are 4,5,9, but for the *AM*, the selected products are 1,2,5. This implies that customers' product adoption behavior can significantly affect the product selection results.

ID	taste	aroma	appearance	preference	palate	overall	price
1	7.111	6.889	7.111	7.667	7.078	6.667	5.664
2	6.620	6.820	6.909	6.967	6.820	6.714	5.896
3	6.962	6.423	6.913	6.462	6.758	6.692	5.989
4	4.375	3.875	4.458	5.333	4.500	5.333	8.640
5	6.750	6.250	7.000	6.500	6.650	6.000	6.152
6	6.417	6.437	6.631	7.241	6.576	6.285	6.084
7	6.194	6.676	6.475	6.504	6.437	6.230	6.253
8	5.822	6.111	5.911	6.578	5.987	5.778	6.655
9	6.404	6.316	6.623	6.632	6.460	6.035	6.261
10	6.822	6.800	6.911	7.289	6.862	6.444	5.856

Table 4: $\mathcal{P}_N$ of *RateBeer* dataset

Models	*ID of selected beers*
UM	4, 5, 9
DM	4, 5, 1
SM	3, 5, 4
AM	5, 2, 1
MM	3, 5, 9

Table 5: Solutions of *RateBeer*

• Summarizations and discussions

Product adoption models can significantly affect the selection results. Different adoption models may result in different product selection strategies. In other words, analysis of customers' adoption behavior is significant for product selection. Hence, manufacturers should put more effort into understanding and discovering customers' product adoption behavior, so as to improve the accuracy of product selection and maximize their sales.

7. RELATED WORK

Kleinberg et al. [7] first advocated using a microeconomic approach on data mining. Since then, a number of works have examined the commercial issues like potential customer identification [1, 17], product feature promotion [16, 11], as well as product positioning [8, 10]. These works show the possibility of helping organizational decision makers to increase their utility.

Authors in [8] extended the concept of dominance, which is used in skyline operators [3], to analyze various forms of relationships among products and customers. By analyzing the dominance relationships, manufacturers can position products effectively while remaining profitable. They extended their results in [9], which took into account not only those min/max attributes (e.g., price, ratings), but also spatial attributes (e.g., location). In these two papers, the authors only considered one manufacturer without competitors while our paper considers this competition. Zhang et al. [18] analyzed the situation that there exist numerous competing companies. They derived the Nash Equilibrium if each manufacturer tries to modify its product in a round robin manner to maximize the market share. Their work only allows each manufacturer to produce one product, which cannot truly reflect the real world market.

Authors in [15] aimed to find the most competitive products using the skyline operator [3]. They could find a group

of candidate products which are not dominated by any competitors. They extended this work by taking customers into account [14, 13], and aimed to find the k most profitable products and k most popular products which can attract the largest number of potential customers. Authors in [2] studied a similar problem named k-most attractive candidates query, where the attractiveness of a product is defined based on the concept of reverse skyline query. The above papers all aimed to find the products which can maximize the *number of potential customers*. In our work, we derive the expected *number of adopters* since the probability a potential customer adopting new products depends on the *number of competitors*. For example, a customer attracted by only one product has a much higher probability to adopt new products than those customers attracted by many products. So the number of potential customers (derived in [15, 3, 14, 13, 2]) is not the same with the expected number of adopters which is derived in our work. Lin et al. [10] aimed to find the products with maximum expected number of total adopters. However, they did not provide any theoretical performance guarantee for their proposed approximation algorithms.

Moreover, none of the previous works considered the complex product adoption behavior of customers and its significant impact. They simply assumed that the customers will randomly select the satisfactory products. Furthermore, all the previous works only considered the market entry and did not address the problem of market sustainability.

8. CONCLUSION

We presented a general framework for the product selection problems: the k-**BSP** problem and the k-**BBP** problem, which are applicable for the market entry and market sustainability, respectively. We mathematically proved that both the k-**BSP** and k-**BBP** problems are *NP-hard* when the number of attributes is three or more. We presented various product adoption models to describe the complex behavior of customers. We proposed approximation algorithms to solve the k-**BSP** and k-**BBP** problems. Our algorithms are computationally efficient, and we also formally proved that they can guarantee a $(1-1/e)$-approximation by the submodularity analysis. We conducted a set of comprehensive experiments using both synthetic datasets and real-world web dataset for quantitative and qualitative analysis of our approximation algorithms. The results showed that our algorithms are remarkably faster than the exhaustive search algorithms and our algorithms can provide high accuracy: much higher than the lower bound that the algorithms guarantee. We also showed that different adoption models will significantly influence the results of the k-**BSP** and k-**BBP** problems, which reflects the importance of considering the behavior of customers for market entry and sustainability.

9. ACKNOWLEDGEMENTS

The work of John C.S. Lui is supported in part by the GRF Grant 415112.

10. REFERENCES

[1] E. Achtert, C. Böhm, P. Kröger, P. Kunath, A. Pryakhin, and M. Renz. Efficient reverse k-nearest neighbor search in arbitrary metric spaces. In *SIGMOD*, pages 515–526, 2006.

[2] A. Arvanitis, A. Deligiannakis, and Y. Vassiliou. Efficient influence-based processing of market research queries. In *CIKM*, pages 1193–1202, 2012.

[3] S. Borzsony, D. Kossmann, and K. Stocker. The skyline operator. In *ICDE*, pages 421–430, 2001.

[4] Y.-J. Chiu, H.-C. Chen, G.-H. Tzeng, and J. Z. Shyu. Marketing strategy based on customer behaviour for the LCD-TV. In *IJMDM*, pages 143–165, 2006.

[5] S. X. et.al. Market entry and sustainability: A provable algorithmic approach to product selection. `http://appsrv.cse.cuhk.edu.hk/~slxu/tr-ps.pdf`.

[6] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137–146, 2003.

[7] J. Kleinberg, C. Papadimitriou, and P. Raghavan. A microeconomic view of data mining. *Data mining and knowledge discovery*, pages 311–324, 1998.

[8] C. Li, B. C. Ooi, A. K. Tung, and S. Wang. Dada: a data cube for dominant relationship analysis. In *SIGMOD*, pages 659–670, 2006.

[9] C. Li, A. K. Tung, W. Jin, and M. Ester. On dominating your neighborhood profitably. In *VLDB*, pages 818–829, 2007.

[10] C.-Y. Lin, J.-L. Koh, and A. L. Chen. Determining k-most demanding products with maximum expected number of total customers. In *TKDE*, pages 1732–1747, 2012.

[11] M. Miah, G. Das, V. Hristidis, and H. Mannila. Standing out in a crowd: Selecting attributes for maximum visibility. In *ICDE*, pages 356–365, 2008.

[12] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of approximations for maximizing submodular set functions - I. *Mathematical Programming*, pages 265–294, 1978.

[13] Y. Peng, R. C.-W. Wong, and Q. Wan. Finding top-k preferable products. In *TKDE*, pages 1774–1788, 2012.

[14] Q. Wan, R. Wong, and Y. Peng. Finding top-k profitable products. In *ICDE*, pages 1055–1066, 2011.

[15] Q. Wan, R. C.-W. Wong, I. F. Ilyas, M. T. Özsu, and Y. Peng. Creating competitive products. In *VLDB*, pages 898–909, 2009.

[16] T. Wu, D. Xin, Q. Mei, and J. Han. Promotion analysis in multi-dimensional space. In *VLDB*, pages 109–120, 2009.

[17] W. Wu, F. Yang, C.-Y. Chan, and K.-L. Tan. Finch: Evaluating reverse k-nearest-neighbor queries on location data. In *VLDB*, pages 1056–1067, 2008.

[18] Z. Zhang, L. V. S. Lakshmanan, and A. K. H. Tung. On domination game analysis for microeconomic data mining. In *TKDD*, pages 18:1–18:27, 2009.