Product Selection Problem: Improve Market Share by Learning Consumer Behavior

SILEI XU and JOHN C. S. LUI, The Chinese University of Hong Kong

It is often crucial for manufacturers to decide what products to produce so that they can increase their market share in an increasingly fierce market. To decide which products to produce, manufacturers need to analyze the consumers' requirements and how consumers make their purchase decisions so that the new products will be competitive in the market. In this paper, we first present a general distance-based product adoption model to capture consumers' purchase behavior. Using this model, various distance metrics can be used to describe different real life purchase behavior. We then provide a learning algorithm to decide which set of distance metrics one should use when we are given some accessible historical purchase data. Based on the product adoption model, we formalize the *k most marketable products (or k-**MMP**)* selection problem and formally prove that the problem is *NP-hard*. To tackle this problem, we propose an efficient greedy-based approximation algorithm with a provable solution guarantee. Using submodularity analysis, we prove that our approximation algorithm can achieve at least 63% of the optimal solution. We apply our algorithm on both synthetic datasets and real-world datasets (TripAdvisor.com), and show that our algorithm can easily achieve five or more orders of speedup over the exhaustive search and achieve about 96% of the optimal solution on average. Our experiments also demonstrate the robustness of our distance metric learning method, and illustrate how one can adopt it to improve the accuracy of product selection.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Product selection, consumer behavior, model learning, submodular set function, approximation algorithm

ACM Reference Format:
Silei Xu and John C. S. Lui. 2016. Product selection problem: Improve market share by learning consumer behavior. ACM Trans. Knowl. Discov. Data 10, 4, Article 34 (June 2016), 25 pages.
DOI: http://dx.doi.org/10.1145/2753764

1. INTRODUCTION

Product competition in the current digital age is becoming increasingly fierce. Consumers can easily access the information about a given product via the internet. Moreover, consumers can share their opinions on products in the form of ratings or reviews via various web services, e.g., Amazon. Therefore, instead of relying on the sales pitch by salesmen or traditional TV advertisements, consumers can now review many competing products before they make their final purchase decision. Manufacturers, on the other hand, can use the web information, such as ratings and reviews, to gain a better understanding of consumers' requirements on various products. This leads to a

The work of John C.S. Lui is supported in part by the GRF Grant 415112.
Authors' addresses: S. Xu and J. C. S. Lui, The Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong; emails: ray.slxu@gmail.com, cslui@cse.cuhk.edu.hk.

DOI: http://dx.doi.org/10.1145/2753764

new challenge on how to discover consumers' preferences, and how these preferences may help manufacturer to select appropriate new products so to compete with other manufacturers in the market.

To introduce new products into a market, a manufacturer usually has a set of *candidate products* to consider. However, due to budget constraints, the manufacturer can only produce a small subset of these candidate products. The objective of a manufacturer is to select a subset of products which can maximize its profit or market share. In this study, we consider the following scenario: In a market consisting of a set of existing products from various manufacturers and a set of consumers, a manufacturer wants to select "*k most marketable products" (k-***MMP***)* from a set of candidate products so as to maximize the market share of all products from this manufacturer (this includes the possibility that some existing products in the market are from the same manufacturer).

One of the major challenges of the "*k*-**MMP**" problem is how to model various consumers' adoption behavior, i.e., how consumers make their purchase decisions. Different adoption behavior may lead to different product selection results. However, there is a lack of formal work of how to model these behaviors using available data. Furthermore, finding the optimal solution to the "*k*-**MMP**" problem can be shown to be *NP-hard* in general.

In this paper, we first model the consumers' adoption behavior with a generalized distance-based model where different distance metrics can be used to describe many different consumers' behaviors. We then propose a method to learn which set of distance metrics one should use when we are given some historical purchase data. We also present a computationally efficient approximation algorithm to solve the *k*-**MMP** problem. To the best of our knowledge, this is the first paper that provides the formal consumers' adoption model and the analysis of product selection. The contributions of this paper are as follows:

- We formulate the problem of finding the *k*-**MMP** for a manufacturer.
- We model the adoption behavior of consumers using a general *distance-based product adoption model* which can take on various different distance metrics.
- Given a set of potential distance metrics, we provide a learning method to determine the appropriate probability distribution of these distance metrics using the historical purchase data on market share, actual sales, or only the ratio of actual sales of a subset of existing products.
- We prove that the *k*-**MMP** problem is *NP-hard* and propose a computationally efficient approximation algorithm. By proving the monotonicity and submodularity properties of the objective function, we show that our approximation algorithm provides a $(1 - 1/e)$-approximation as compared with the optimal solution.
- We perform experiments on synthetic datasets to demonstrate the efficiency and accuracy of our algorithm when varying parameters of the experiments.
- We illustrate the significant impact of different distance metrics and how one can adopt our learning method to improve the market share. We also show that our learning method maintains high accuracy no matter whether we have the perfect information of the potential distance metrics or not.

The outline of the paper is as follows. In Section 2, we propose a general product adoption model which can accommodate different distance metrics to describe the consumers' adoption behavior, and we formulate the *k*-**MMP** problem. In Section 3, we present a learning method to select the appropriate set of distance metrics according to the historical market share of existing products. In Section 4, we propose an exact algorithm for the case of $k = 1$ and prove that finding the exact solution is *NP-hard* with respect to a general k. To tackle the computational challenge, we present an approximation algorithm in Section 5. We show that this algorithm is computationally efficient and also provides a high-quality solution guarantee. In Section 6, we perform experiments on both the synthetic data and the real-world data. Related work is shown in Section 7, and Section 8 concludes.

2. MATHEMATICAL MODELS AND PROBLEM FORMULATION

In this section, we first present a model of a market by considering both products and consumers. Then, we present a *distance-based product adoption model* to describe various consumers' product adoption behaviors. Based on these models, we formulate the k-**MMP** problem.

2.1. Market Model

Let us consider a market which consists of a set of l consumers $\mathcal{C} = \{c_1, c_2, \ldots, c_l\}$ and a set of m existing products $\mathcal{P}_E = \{p_1, p_2, \ldots, p_m\}$. Let M represent a manufacturer in the market, and $\mathcal{P}_M$ denote the set of existing products produced by M, where $\mathcal{P}_M \subseteq \mathcal{P}_E$ and $|\mathcal{P}_M| = m_M$. The remaining products in $\mathcal{P}_E$ are from other manufacturers who are the competitors of M. These competing products are denoted by $\mathcal{P}_C$, where $\mathcal{P}_C \subseteq \mathcal{P}_E$ and $|\mathcal{P}_C| = m_C$. According to these definitions, we have $m = m_M + m_C$, $\mathcal{P}_E = \mathcal{P}_M \cup \mathcal{P}_C$, and $\mathcal{P}_M \cap \mathcal{P}_C = \varnothing$.

Suppose the manufacturer M wants to produce some new products to maximize its utility, i.e., the market share. M has a set of n *candidate new products* to choose from, which we denote by $\mathcal{P}_N = \{p_{m+1}, p_{m+2}, \ldots, p_{m+n}\}$. Note that all the products in $\mathcal{P}_N$ are *new* to the market, in other words, $\mathcal{P}_N \cap \mathcal{P}_E = \varnothing$. Due to the budget, technological and manufacturing constraints, the manufacturer M can only produce $k \leq n$ of these candidate products in $\mathcal{P}_N$.

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 a_i is represented by a non-negative real number, and higher value implies higher quality. One can use a_i to represent various attributes of a given product, e.g., durability, ratings, inverse of price. Hence, the quality of a product can be described by a d-dimensional vector. Specially, the quality of product p_j is described by the vector $\boldsymbol{q}_j = (q_j[1], q_j[2], \ldots, q_j[d])$, where $q_j[t] \in [0, \infty)$, $\forall t \in \{1, 2, .., d\}$ indicates p_j's quality on attribute a_t. Similarly, each consumer in $\mathcal{C}$ is also associated with $\mathcal{A}$ to describe his requirements on different attributes. Let $\boldsymbol{r}_i = (r_i[1], r_i[2], \ldots, r_i[d])$ be the requirement vector of consumer c_i, where $r_i[t] \in [0, \infty)$, $\forall t \in \{1, 2, \ldots, d\}$ indicates c_i's *minimum* requirement on attribute a_t, i.e., c_i requires that the product's quality on attribute a_t is at least $r_i[t]$, or he will not adopt (or purchase) that product.

Example 2.1. To illustrate the notations, we present an example in Figure 1. Consider a market of smart phones where we have two existing products $\mathcal{P}_E = \{p_1, p_2\}$ and three consumers $\mathcal{C} = \{c_1, c_2, c_3\}$. Manufacturer M is considering two candidate products $\mathcal{P}_N = \{p_3, p_4\}$. Let us say each product is described by two attributes: a_1 is the inverse of price (units per thousand dollars, UPM for short) and a_2 is durability (years), and they are represented in the horizontal and the vertical axis, respectively. The quality vectors of products and the requirement vectors of consumers are shown in the figure (with $\mathcal{P}_E$:$\Diamond$, $\mathcal{P}_N$:$\Box$, $\mathcal{C}$:$\circ$). For instance, the quality vector of p_1 is $(2, 6)$, so we can purchase two units of p_1 with one thousand dollars (or the price of p_1 is \$500), and the durability of p_1 is 6 years. Similarly, the requirement vector of c_1 is $(1, 5)$, so consumer c_1 wants a product which is at most \$1,000 and can last for at least 5 years.

2.2. Product Adoption Model

We assume that a consumer may adopt a product if the product satisfies his requirement. We say that a product satisfies a consumer's requirements if and only if the

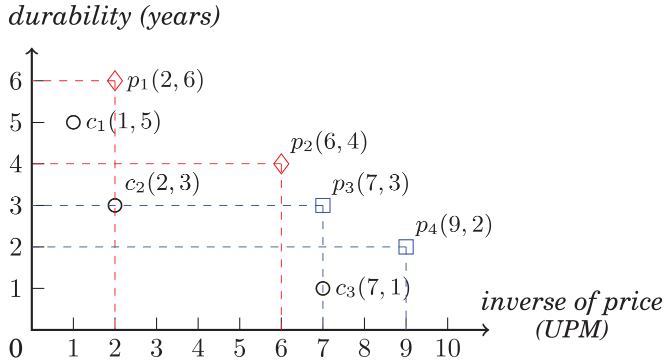


Fig. 1. An illustration of the market model.

product meets the requirements of that consumer on *all* attributes. Formally, we define the product satisfiability condition.

Definition 2.2 (*Product Satisfiability*). Consider a consumer c_i and a product p_j. We say the product p_j satisfies the consumer c_i if and only if $q_j[t] \geq r_i[t]$, $\forall t = 1, \ldots, d$. We denote this relationship as $p_j \succsim c_i$, and p_j is said to be a satisfactory product of c_i, while c_i is a potential consumer of p_j in other words.

For example, consider the products and consumers depicted in Figure 1. One can observe that the quality vector of p_1 is $(2, 6)$ and the requirement vector of c_1 is $(1, 5)$. Since $2 > 1$ and $6 > 5$, so p_1 satisfies c_1, or $p_1 \succsim c_1$. Similarly, we have $p_3 \succsim c_2$ and $p_3 \succsim c_3$.

We assume that if a consumer has some satisfactory products, then he will adopt one unit of product from any of these feasible products. When a consumer c_i has only one satisfactory product, say p_j, then c_i will adopt p_j for sure. However, it becomes complicated when there are multiple satisfactory products. All previous works [Li et al. 2006; Peng et al. 2012; Wan et al. 2011; Zhang et al. 2009] assume that the consumer will *randomly* adopt one of the satisfactory products, but this is not realistic in many situations. In the following, we present the *distance-based adoption model* to describe some realistic and representative product adoption behavior when consumers make their purchase decisions. Our model is very general to model various product adoption behaviors in the real-world scenarios.

In a real-world market, products with higher quality usually attract more consumers. Therefore, we use a distance measure between a product's quality and a consumer's requirement to decide which product the consumer may adopt. Note that consumers will only consider their satisfactory products. Furthermore, larger distance implies better quality. Let $d_{i,j}$ be the distance between the consumer c_i's requirement vector ($\boldsymbol{r}_i$) and the product p_j's quality vector ($\boldsymbol{q}_j$). We assume that c_i will adopt the product p_j which has the largest distance among all his satisfactory products. If there are multiple satisfactory products which have the same largest distance measure with c_i, then c_i will randomly select one of these products. Mathematically, we define the *distance-based adoption model* as follows.

Definition 2.3 (*Distance-Based Adoption Model*). Given a consumer c_i and a set $\mathcal{P}$ of products available in the market, let $F\mathcal{P}(c_i|\mathcal{P})$ be the set of products which have the largest distance between their quality vectors and c_i's requirement vector among all

c_i's satisfactory products. The probability that c_i adopts a product $p_j \in \mathcal{P}$ is

$$\Pr(i, j|\mathcal{P}) = \begin{cases} \frac{1}{|F\mathcal{P}(c_i|\mathcal{P})|} & \text{if } p_j \in F\mathcal{P}(c_i|\mathcal{P}), \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Note that we can use many *distance metrics*, e.g., l_1, l_2, l_∞ norms. For instance, if l_1 norm (or the *Manhattan distance*) is used, then consumers will choose the satisfactory products which have the largest sum of all components' values in the quality vectors. To describe different adoption behaviors of different consumers in a real-world market, we also take into account the weighted distance metrics. Let w_t be the weight of attribute a_t, $w_t \geq 0, \forall a_t \in \mathcal{A}$, then under the l_1 norm, the distance $d_{i,j}$ can be expressed as

$$d_{i,j} = \sum_{a_t \in \mathcal{A}} w_t \cdot (q_j[t] - r_i[t]). \quad (2)$$

It is important to point out that the algorithms we present in this paper are general to all distance metrics. Readers can use other distance metrics when appropriate. In here, we present four representative distance metrics which we use as examples for illustrations and experiments.

—*Discrete metric (DM)*
We define $d_{i,j} = 1$ for consumers c_i and c_i's satisfactory product p_j in the discrete metric. This distance metric simplifies the adoption model that consumers will randomly select one from all his satisfactory products. Using this distance metric, our work subsumes the adoption models of previous works [Li et al. 2006; Peng et al. 2012; Wan et al. 2011; Zhang et al. 2009].

—*Norm metric (NM)*
In this distance metric, we set the weight $w_t = 1.0, \forall a_t \in \mathcal{A}$ based on the l_1 norm metric as defined in Equation (2). Note that in general, one can use other norm as distance metric and our algorithms still apply.

—*Price metric (PM)*
In a real-world market, one common situation is that if a consumer's requirements are satisfied, then he will select the cheapest product, i.e., the one with the highest quality on the attribute of "price". In this case, we can set the weight of all attributes to zero except the "price" based on the l_1 norm metric as defined in Equation (2).

—*Richman metric (RM)*
Unlike the *PM*, some consumers may be rich and they are insensitive to the price but only want the best product. In this case, we can set the weight of "price" attribute to zero while setting the weight of other attributes to one.

Example 2.4. To illustrate, let us consider the products and consumers depicted in Figure 1. Suppose that manufacturer M decides to produce p_3, then the set of available products in the market is $\mathcal{P} = \mathcal{P}_E \cup \{p_3\} = \{p_1, p_2, p_3\}$. Let us consider the probability c_2 will adopt p_3, i.e., $\Pr(2, 3|\mathcal{P})$, when c_2 uses the above four distance metrics. From Figure 1, one can observe that c_2 is satisfied by p_1, p_2, and p_3. If c_2 uses the *discrete metric*, then $d_{2,1} = d_{2,2} = d_{2,3} = 1$, so $\Pr(2, 3|\mathcal{P}) = 1/3$. If c_2 uses the *norm metric*, then we have $d_{2,1} = 3$, $d_{2,2} = d_{2,3} = 5$. Hence, c_2 will select p_2 and p_3 with probability $\Pr(2, 2|\mathcal{P}) = Pr(2, 3|\mathcal{P}) = 1/2$. If c_2 uses the *price metric*, then we only need to consider the attribute *inverse of price*. We have $d_{2,1} = 0$, $d_{2,2} = 4$, $d_{2,3} = 5$, so $\Pr(2, 3|\mathcal{P}) = 1$, c_2 will adopt p_3. If c_2 uses the *richman metric*, we have $d_{2,1} = 3$, $d_{2,2} = 1$, $d_{2,3} = 0$. Thus, c_2 will adopt p_1 only, or $\Pr(2, 1|\mathcal{P}) = 1$ and $\Pr(2, 3|\mathcal{P}) = 0$.

2.3. Problem Formulation

To find the k-**MMP**, we first need to define the expected market share of a set of products under the distance-based adoption model. Given the market condition, i.e., the consumers $\mathcal{C}$ and the existing products $\mathcal{P}_E$, let P be the set of products we consider, then the expected market share of P is defined as

$$MS(P) = \frac{1}{l} \cdot \sum_{p_j \in P} \sum_{c_i \in \mathcal{PC}(p_j)} \Pr(i, j | \mathcal{P}_E \cup P), \tag{3}$$

where $l = |\mathcal{C}|$, $\mathcal{PC}(p_j)$ denotes the set of potential consumers of product p_j, and $\Pr(i, j|\mathcal{P}_E \cup P)$ is defined in Equation (1).

Example 2.5. Let us illustrate the expected market share of p_3, or $MS(\{p_3\})$, by considering the scenario depicted in Figure 1. There are two existing products ($P_E = \{p_1, p_2\}$) and three consumers ($\mathcal{C} = \{c_1, c_2, c_3\}$) in the market. By adding product p_3 into the market, p_3 satisfies consumers c_2 and c_3. Assume that c_2 uses the *norm metric*, then according to Example 2.4, we have $\Pr(2, 3|\mathcal{P}) = 1/2$. Now consider consumer c_3. Since we have not added p_4 into the market, p_3 is c_3's only satisfactory product, so c_3 will adopt p_3 for sure. Therefore, in this scenario, c_2 and c_3 will adopt p_3 with probability $1/2$ and 1, respectively. So the expected sales of p_3 is 1.5 units. It follows that the expected market share of p_3 is $1.5/3 = 50\%$ since there are three consumers in total.

Based on the definition of market share in Equation (3), we formulate the k-**MMP** problem as follows.

Definition 2.6 (k-**MMP**). Given a set of consumers $\mathcal{C}$, a set of existing products $\mathcal{P}_E = \mathcal{P}_C \cup \mathcal{P}_M$ in the market, and $\mathcal{P}_N$, a set of candidate products by the manufacturer M, select a set $P \subseteq \mathcal{P}_N$, where $|P| = k$ so to maximize $MS(P \cup \mathcal{P}_M)$ for manufacturer M.

To solve the k-**MMP** problem, we need to tackle the following two issues: (1) Find the proper distance metrics for the market. (2) Design an efficient algorithm to find the solution to the k-**MMP** problem. Since there are various potential distance metrics and manufacturers usually do not know which distance metrics the consumers may adopt, we present a learning approach to discover the proper set of distance metrics for a given market from historical purchase data. This is presented in Section 3. After deciding on the proper distance metrics, we present the algorithmic design in solving the k-**MMP** problem. In Section 4, we present an efficient and exact algorithm for the 1-**MMP** problem and prove that the k-**MMP** problem is *NP-hard* in general. In Section 5, we present an efficient approximation algorithm. By exploiting the monotonicity and submodularity properties of the market share function $MS(\cdot)$, we prove that our approximation algorithm can provide high performance guarantee on the quality of the solutions. Table I depicts the notations we use in this paper.

3. DISTANCE METRIC LEARNING

As discussed in Section 2, there are various distance metrics one can use and the product selection results can vary significantly depending on the distance metrics according to the results shown in Section 6. Hence, it is important to "*learn*" about the proper distance metrics (in other words, consumers' product adoption behavior) from the available data. In this work, we propose a learning method based on the market share, actual sales, or even only the ratio of actual sales of a subset of products in the market so to discover the appropriate distance metrics.

Table I. Notations

Notations	*Description*
$\mathcal{C}$	a set of consumers in the market
$\mathcal{P}_E$	a set of existing products in the market
$\mathcal{P}_M$	a set of existing products from manufacturer M
$\mathcal{P}_C$	a set of existing products from M's competitors
$\mathcal{P}_N$	a set of candidate new products for M
$\mathcal{A}$	a set of attributes
l, m, m_M, m_C, n, d	the size of $\mathcal{C}, \mathcal{P}_E, \mathcal{P}_M, \mathcal{P}_C, \mathcal{P}_N, \mathcal{A}$, respectively
k	the number of new products to produce
$\boldsymbol{r}_i$	consumer c_i's requirement vector
$\boldsymbol{q}_j$	product p_j's quality vector
$d_{i,j}$	the distance between $\boldsymbol{r}_i$ and $\boldsymbol{q}_j$
$F\mathcal{P}(c_i\|\mathcal{P})$	the set of c_i's satisfactory products which have the largest distance to c_i when a set $\mathcal{P}$ of products are available in the market
$\Pr(i, j\|\mathcal{P})$	the probability that c_i adopt p_j when a set $\mathcal{P}$ of products are available in the market
$\mathcal{PC}(p_j)$	a set of potential consumers of p_j
ms_j	the real-world market share of p_j
s_j	the real-world sales of p_j
$R_{j,j'}$	the ratio between s_j and $s_{j'}$
e_{ji}	the expected market share of p_j under the adoption model using ith potential distance metric
θ_i	the probability consumers use the ith potential distance metric
f_j	the forecast market share of p_j

3.1. Learning From Market Share

Note that in real life, some manufacturers may not release full information about their market share. Therefore, we assume that we only know the market share of a *subset* of existing products. Formally, let $\mathcal{P}'_E$ be the n' products that we know the market share data, where $\mathcal{P}'_E \subseteq \mathcal{P}_E$. Let ms_j be the market share of $p_j \in \mathcal{P}'_E$.

Assume that we have a *model set* consisting of distance-based product adoption models using m' different potential distance metrics, which are numbered from 1 to m'. Let e_{ji} be the expected market share of product p_j under the product adoption model using the ith potential distance metric, and $\boldsymbol{e}_j = (e_{j1}, e_{j2}, \ldots, e_{jm'})^T$. Let θ_i be the probability that consumers use the ith distance metric, and $\Theta = (\theta_1, \theta_2, \ldots, \theta_{m'})^T$. Then, we can forecast the market share for each product $p_j \in \mathcal{P}'_E$ as

$$\begin{aligned} f_j(\Theta) &= \Theta^T \cdot \boldsymbol{e}_j \\ &= \theta_1 e_{j1} + \theta_2 e_{j2} + \cdots + \theta_{m'} e_{jm'}, \end{aligned} \tag{4}$$

where $f_j(\Theta)$ is the forecast market share of product p_j.

We can find the best fit for Θ by minimizing the squared difference between the forecast market share f_j and the real-world market share ms_j. Let Δ_j be the difference between f_j and ms_j, or mathematically,

$$\Delta_j(\Theta) = |f_j(\Theta) - ms_j|. \tag{5}$$

Then, we can formalize the model selection problem as follows:

$$\begin{aligned} &\text{Minimize} \quad \sum_{p_j \in \mathcal{P}'_E} \Delta_j^2(\Theta), \\ &\text{subject to} \quad \Theta \geq \mathbf{0}, \quad \theta_1 + \theta_2 + \cdots + \theta_{m'} = 1, \end{aligned} \tag{6}$$

where $\Theta \geq \mathbf{0}$ means that $\theta_i \geq 0, \forall i \in \{1, \ldots, m'\}$.

Table II. An Example of Model Selection

	NM	*PM*	*RM*	*real-world*
p_1	20%	1%	50%	5%
p_2	30%	10%	10%	15%
p_3	5%	30%	0%	20%
p_4	10%	40%	5%	unknown

Thus, the problem is reduced to a linear regression problem with constrained least-squares approach, which can be solved using the technique in Gill et al. [1981]. Once we solve this linear regression problem, we can forecast the market share $f_j(\Theta)$ based on the probability vector Θ.

Example 3.1. Consider a *model set* consisting of adoption models using the NM, PM, and RM. Assume, we obtain the real-world market share of three products p_1, p_2, and p_3 and we want to forecast the market share of p_4. The real-world market share and the expected market share under three different models of these products are shown in Table II.

Let θ_1, θ_2, and θ_3 be the probability of consumers using the "NM", the "PM", and the "RM", respectively. Then, we can formalize the problem as follows:

$$\begin{aligned}&\text{Minimize} \quad \left\| \begin{matrix} \theta_1 \cdot 20\% & \theta_2 \cdot 1\% & \theta_3 \cdot 50\% & -5\% \\ \theta_1 \cdot 30\% & \theta_2 \cdot 10\% & \theta_3 \cdot 10\% & -15\% \\ \theta_1 \cdot 5\% & \theta_2 \cdot 30\% & \theta_3 \cdot 0\% & -20\% \end{matrix} \right\|_2^2 \\ &\text{subject to} \quad \Theta \geq \mathbf{0}, \quad \theta_1 + \theta_2 + \theta_3 = 1.\end{aligned}$$

We obtain $\Theta = (0.3074, 0.6926, 0)^T$ by solving the above optimization problem. Thus, we can forecast that the real-world market share of p_4 as $\Theta^T \cdot \boldsymbol{e}_j = (0.3074, 0.6926, 0) \cdot (10\%, 40\%, 5\%)^T = 30.78\%$.

3.2. Learning From Sales

Foregoing approach also works well if we have the products' actual sales. In fact, this is not as restrictive as using the market share because each manufacturer knows exact its own sales: A manufacturer M knows the actual sales of its own existing products $\mathcal{P}_M$ in the market.

Assume that we know the actual sales of products in $\mathcal{P}'_E$, where s_j denotes the sales of p_j. Let L be the number of all consumers in the market, then the market share of p_j can be expressed as $ms_j = s_j/L$. Since we want our forecast market share f_j is as close as possible to the real-world market share ms_j, the ratio between f_j and s_j should approach to a constant for any $p_j \in \mathcal{P}'_E$:

$$f_j/s_j = f_j/(ms_j \cdot L) \approx 1/L. \tag{7}$$

Thus, in this case we minimize the squared difference between f_j/s_j and $f_{j'}/s_{j'}$ for each pair of products $p_j, p_{j'} \in \mathcal{P}'_E$. Let $\Delta_{j,j'}(\Theta)$ be the difference between f_j/s_j and $f_{j'}/s_{j'}$, or mathematically,

$$\Delta_{j,j'}(\Theta) = |f_j(\Theta)/s_j - f_{j'}(\Theta)/s_{j'}|. \tag{8}$$

Then, the problem can also be transformed to a constrained minimization problem as follows:

$$\begin{aligned}&\text{Minimize} \quad \sum_{p_j, p_{j'} \in \mathcal{P}'_E} \Delta^2_{j,j'}(\Theta), \\ &\text{subject to} \quad \Theta \geq \mathbf{0}, \quad \theta_1 + \theta_2 + \cdots + \theta_{m'} = 1,\end{aligned} \tag{9}$$

where

$$\begin{aligned}\Delta_{j,j'}(\Theta) &= |f_j(\Theta)/s_j - f_{j'}(\Theta)/s_{j'}| \\ &= \left|\Theta^T \cdot \frac{\boldsymbol{e}_j}{s_j} - \Theta^T \cdot \frac{\boldsymbol{e}_{j'}}{s_{j'}}\right| \\ &= \left|\Theta^T \cdot \left(\frac{e_{j1}}{s_j}, \ldots, \frac{e_{jm'}}{s_j}\right)^T - \Theta^T \cdot \left(\frac{e_{j'1}}{s_{j'}}, \ldots, \frac{e_{j'm'}}{s_{j'}}\right)^T\right| \\ &= \left|\Theta^T \cdot \left(\frac{e_{j1}}{s_j} - \frac{e_{j'1}}{s_{j'}}, \ldots, \frac{e_{jm'}}{s_j} - \frac{e_{j'm'}}{s_{j'}}\right)^T\right|.\end{aligned} \tag{10}$$

One can observe that $\Delta_{j,j'}(\Theta)$ is a linear function of Θ as shown in Equation (10), so this problem is a linear regression problem with constrained least-squares approach. By solving this linear regression problem, we can forecast the market share $f_j(\Theta)$ based on the probability vector Θ.

3.3. Learning From Ratio of Sales

When a new manufacturer enters the market, it has no existing product in the current market, it is possible that the manufacturer does not know any product's market share or sales. However, it is not difficult to obtain the ratio of sales. For instance, if we can obtain the sales of products on an online store, then we can get an estimation of the ratio of the total sales.

Assume that we know the ratios of the actual sales of products in $\mathcal{P}'_E$, or mathematically, $R_{j,j'} = s_j/s_{j'}$. Let L be the number of all consumers in the market. With similar derivation of Equation (7), we have

$$\begin{aligned}& f_j/s_j = f_j/(s_{j'} \cdot R_{j,j'}) \approx f_{j'}/s_{j'} \approx 1/L \\ \Rightarrow\ & f_j/R_{j,j'} \approx s_{j'}/L \approx f_{j'}\end{aligned}. \tag{11}$$

Thus, in this case we minimize the squared difference between $f_j/R_{j,j'}$ and $f_{j'}$ for each pair of products $p_j, p_{j'} \in \mathcal{P}'_E$. Let $\Delta'_{j,j'}(\Theta)$ be the difference between $f_j/R_{j,j'}$ and $f_{j'}$, or mathematically,

$$\Delta'_{j,j'}(\Theta) = |f_j(\Theta)/R_{j,j'} - f_{j'}(\Theta)|. \tag{12}$$

Then, the problem can also be transformed to a constrained minimization problem as follows:

$$\begin{aligned}&\text{Minimize} \sum_{p_j, p_{j'} \in \mathcal{P}'_E} \Delta^{2'}_{j,j'}(\Theta), \\ &\text{subject to} \quad \Theta \geq \mathbf{0}, \quad \theta_1 + \theta_2 + \cdots + \theta_{m'} = 1,\end{aligned} \tag{13}$$

where

$$\begin{aligned}\Delta'_{j,j'}(\Theta) &= |f_j(\Theta)/R_{j,j'} - f_{j'}(\Theta)| \\ &= \left|\Theta^T \cdot \frac{\boldsymbol{e}_j}{R_{j,j'}} - \Theta^T \cdot \boldsymbol{e}_{j'}\right| \\ &= \left|\Theta^T \cdot \left(\frac{e_{j1}}{R_{j,j'}}, \ldots, \frac{e_{jm'}}{R_{j,j'}}\right)^T - \Theta^T \cdot \left(e_{j'1}, \ldots, e_{j'm'}\right)^T\right| \\ &= \left|\Theta^T \cdot \left(\frac{e_{j1}}{R_{j,j'}} - e_{j'1}, \ldots, \frac{e_{jm'}}{R_{j,j'}} - e_{j'm'}\right)^T\right|.\end{aligned} \tag{14}$$

Table III. Farthest Product Table Under *NM*

Consumers	$fd_1[i]$	$e_1[i]$	$m_1[i]$
c_1	2	$1\ (p_3)$	0
c_2	5	$2\ (p_2, p_3)$	$1\ (p_3)$
c_3	3	$1\ (p_4)$	$1\ (p_4)$

Note that $\Delta'_{j,j'}(\Theta)$ is also a linear function of Θ as shown in Equation (14), so this problem is also a linear regression problem with constrained least-squares approach. By solving this linear regression problem, we can forecast the market share $f_j(\Theta)$ based on the probability vector Θ.

In Section 6, we will show that we can estimate the probability vector Θ with high accuracy if we know the model set and the historical purchase data of a small number of products, based on which, we can find products with higher market share.

4. EXACT ALGORITHM AND HARDNESS

Let us first present the exact algorithm for solving a special case of the *k*-**MMP** problem when $k = 1$. This will serve as the foundation of our approximation algorithm in Section 5. Then, we prove the *NP-hardness* of the *k*-**MMP** problem.

4.1. Exact Top-1 Algorithm

One way to find the exact solution of the **1-MMP** problem is via exhaustive search: Calculate the expected market share for all candidate products in $\mathcal{P}_N$ and select the product with the largest market share. To calculate the expected market share of a product, we need to check the requirement vectors of all l consumers and the quality vectors of their satisfactory products with time complexity $O(mld)$, where m is the number of existing products and d is the dimension of the attribute vector $\mathcal{A}$. Assume that we consider a model set S consisting of m' potential product adoption models. Since there are n candidate products, the computational complexity of the exhaustive search is $O(m'mnld)$.

In the following, we present an enhanced algorithm for the **1-MMP** problem based on precomputation. This enhanced algorithm has a lower computational complexity, or $O(m'(m+n)ld)$. The main idea is as follows.

Let S be the model set consisting of m' potential product adoption models. Under each product adoption model, we build a *farthest product table* for each consumer $c_i \in \mathcal{C}$ to store the information about $F\mathcal{P}(c_i|\mathcal{P}_E)$, which represents the set of satisfactory products which are farthest from c_i when only the existing products $\mathcal{P}_E$ are considered. We store the distances between c_i and these c_i's farthest satisfactory products, the number of these farthest products, as well as the number of products from manufacturer M among these farthest products. We denote them as $fd_t[i]$, $e_t[i]$, and $m_t[i]$ under the distance metric model t, respectively. Formally, they can be expressed as

$$
\begin{aligned}
fd_t[i] &= d_{i,j}, \text{ where } p_j \in F\mathcal{P}(c_i|\mathcal{P}_E), \\
e_t[i] &= |F\mathcal{P}(c_i|\mathcal{P}_E)|, \\
m_t[i] &= |F\mathcal{P}(c_i|\mathcal{P}_E) \cap \mathcal{P}_M|. \qquad (15)
\end{aligned}
$$

The algorithm of building the farthest product table is shown in Algorithm 1.

Example 4.1. To illustrate, let us consider the products and consumers depicted in Figure 1 again. Let us say that manufacturer M has produced both p_3 and p_4, then in the current market we have $\mathcal{P}_E = \{p_1, \ldots, p_4\}$, $\mathcal{P}_M = \{p_3, p_4\}$. Suppose that consumers use adoption model NM. Assume that NM is adoption model 1, then we can build the farthest product table as shown in Table III.

ALGORITHM 1: Farthest Product Table Builder

```
Input: P_E, P_M, C, adoption model t
Output: farthest product table under adoption model t
for all c_i ∈ C(p_j) do
    fd_t[i] ← 0
    FP(c_i|P_E) ← ∅
    for all p_j ∈ P_E do
        if p_j ≿ c_i and d_{i,j} > fd_t[i] then
            fd_t[i] ← d_{i,j}, FP(c_i|P_E) ← {p_j}
        else if p_j ≿ c_i and d_{i,j} = fd_t[i] then
            FP(c_i|P_E) ← FP(c_i|P_E) ∪ {p_j}
        end
    end
    e_t[i] ← |FP(c_i|P_E)|
    m_t[i] ← |FP(c_i|P_E) ∩ P_M|
end
```

Then, for each candidate new product $p_j \in \mathcal{P}_N$, instead of calculating the market share according Equation (3), we can simply perform a table lookup to check whether each consumer will be influenced by the new product, and then calculate the increase of sales by adding p_j. Based on the increase of sales under different distance metric models and the probability of using each model, we can calculate the expected increase of sales under the given model set S. The product which has the largest expected increase on sales will be returned as the result of the algorithm. The pseudo code of this precomputation-based exact algorithm is shown in Algorithm 2.

ALGORITHM 2: Exact Top-1 Algorithm

```
Input: P_E, P_M, P_N, C, S, Θ
Output: 1-MMP
for all model t in S do
    build farthest product table
end
max_increase ← 0
for all p_j ∈ P_N do
    for all c_i ∈ PC(p_j) under each model t in S do
        Δsales_t(p_j) ← 0
        if d(i, j) > fd_t[i] then
            Δsales_t(p_j) ← Δsales_t(p_j) + (1 − m_t[i]/e_t[i])
        else if d(i, j) = fd_t[i] then
            Δsales_t(p_j) ← Δsales_t(p_j) + ((m_t[i]+1)/(e_t[i]+1) − m_t[i]/e_t[i])
        end
    end
    ΔSales(p_j) ← θ_1 Δsales_1(p_j) + ··· + θ_{m'} Δsales_{m'}(p_j)
    if ΔSales(p_j) > max_increase then
        res ← p_j max_increase ← ΔSales(p_j)
    end
end
return res
```

LEMMA 4.2. *The computational complexity of Algorithm 2 is $O(m'(m+n)ld)$, where $m' = |S|$, $m = |\mathcal{P}_E|$, $n = |\mathcal{P}_N|$, $l = |\mathcal{C}|$, $d = |\mathcal{A}|$.*

PROOF. First, we build the *farthest products table*. It takes $O(d)$ time to calculate the distance for each pair of consumer and product, while there are l consumers, m existing products, and m' product adoption models, so the complexity of building the table is $O(m'mld)$. Then, for each product $p_j \in \mathcal{P}_N$, we calculate the increase of sales caused by adding p_j, which takes $O(m'ld)$ time. Since there are n candidate new products, the complexity of these steps is $O(m'nld)$. Therefore, the total computational complexity of Algorithm 2 is $O(m'(m+n)ld)$. □

We like to point out that for some cases, Algorithm 2 could be further optimized by using skyline algorithms. The idea is as following. Consider the case that the distance metric adopted is not a *discrete metric*, then if a product p_j is dominated by another product and there does not exist an attribute $a_t \in \mathcal{A}$ that $\boldsymbol{q}_j[t] = \max_{p_i \in \mathcal{P}_E} \boldsymbol{q}_i[t]$, p_j will not have the largest distance to any consumers. Thus, by filtering out these type of products, we could reduce the number of products we need to calculate the distance. The known optimal skyline algorithm costs $O(N(\log^{d-3} N)d)$ for $d > 4$ [Chan et al. 2011], where N and d denote the number of points and the dimension of the space, respectively. In our case, $N = m+n$. So the filtering process will cost $O((m+n)(\log^{d-3}(m+n))d)$ time. Let $n' \leq m+n$ denote the number of skyline products, then the complexity of optimized algorithm will be $O(m'n'ld + (m+n)(\log^{d-3}(m+n))d)$. Note that this optimization will improve the performance only when $n' < m+n$ and $(m+n)\log^{d-3}(m+n) < m'l$, otherwise, it will have a worse complexity as compared with Algorithm 2. Thus, in the following analysis, we still adopt Algorithm 2 as the algorithm of finding the exact solution of top-1 **MMP** problem.

4.2. Top-*k* Exact Algorithm

Similarly, exhaustive search is a direct approach to find the exact solution of the k-**MMP** problem. By enumerating all possible subsets of size k from $\mathcal{P}_N$, and calculating the expected market share of each subset, one can find the set of product with size k which achieves the largest market share. However, the exhaustive approach is *not scalable* since there exist exponentially many possible subsets. In the following theorem, we formally show that finding the exact solution of the k-**MMP** problem is NP-hard, where k is considered as a variable.

THEOREM 4.3. *Finding the exact solution for the k-**MMP** selection problem is NP-hard when the number of attributes d is three or more.*

PROOF. Please refer to the appendix. □

5. APPROXIMATION ALGORITHM

In this section, we extend the top-1 algorithm for the k-**MMP** problem using a greedy-based approximation algorithm. The algorithm is not only computationally efficient, but also provide at least $(1 - 1/e)$-approximation by exploiting that the market share function is monotone and submodular. In the following, let us first present our approximation algorithm. Then, we formally prove its performance guarantee, and finally prove that the market share function we consider is indeed monotone and submodular.

5.1. Greedy-based Approximation Algorithm

Our approximation algorithm is based on the exact top-1 algorithm to solve the top-k problem. The main idea is as follows. We select k products in k steps. In each step, we select the product which is the solution of the exact top-1 algorithm. Furthermore, instead of building the farthest product tables at each step, we only build them in the first step, and then update the tables in the remaining steps. The pseudo code of this algorithm is depicted in Algorithm 3.

THEOREM 5.1 (COMPUTATIONAL COMPLEXITY). *The computational complexity of Algorithm 3 is $O(m'(m+kn)ld)$, where $m' = |S|$, $m = |\mathcal{P}_E|$, $n = |\mathcal{P}_N|$, $l = |\mathcal{C}|$, $d = |\mathcal{A}|$.*

ALGORITHM 3: Approximation Top-k Greedy Algorithm

Input: $\mathcal{P}_E, \mathcal{P}_M, \mathcal{P}_N, \mathcal{C}, S, \Theta, k$
Output: k-**MMP**
$P_{res} \leftarrow \varnothing$
while $|P_{res}| < k$ **do**
  $p_{new} \leftarrow$ solution of the exact top-1 algorithm
  for $c_i \in \mathcal{PC}(p_{new})$ *under each model t in S* **do**
    if $d(i, new) > fd_t[i]$ **then**
      $fd_t[i] \leftarrow d(i, new), e_t[i] \leftarrow 1, m_t[i] \leftarrow 1$
    else if $d(i, new) = fd_t[i]$ **then**
      $e_t[i] \leftarrow e_t[i] + 1, m_t[i] \leftarrow m_t[i] + 1$
    end
  end
  $P_{res} \leftarrow P_{res} \cap \{p_{new}\}$
  $\mathcal{P}_M \leftarrow \mathcal{P}_M \cup \{p_{new}\}$
  $\mathcal{P}_N \leftarrow \mathcal{P}_N \setminus \{p_{new}\}$
end
return P_{res}

Proof. Based on Lemma 4.2, it takes $O(m'mld)$ time to build these farthest product tables and $O(m'nld)$ time to find the exact solution of 1-**MMP**. The complexity of updating tables is only $O(ld)$. Since we only build the tables once and find the 1-**MMP** k times in Algorithm 3, the computational complexity of Algorithm 3 is $(m'(m+kn)ld)$. □

5.2. Guarantee on Solution Quality

To prove the performance guarantee of our approximation algorithm, let us first introduce the notion of "*submodular set function*" [Nemhauser et al. 1978].

Definition 5.2 (*Submodular Set Function*). Given a finite ground set U, a function f that maps subsets of U to real numbers is called *submodular* if

$$f(S \cup \{u\}) - f(S) \geq f(T \cup \{u\}) - f(T), \ \forall S \subseteq T \subseteq U, u \in U. \tag{16}$$

Next, we show one interesting property of submodular set functions [Kempe et al. 2003], based on which we design our approximation algorithm with theoretical performance guarantee.

Theorem 5.3. *For a non-negative monotone submodular function $f : 2^U \rightarrow R$, 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 $f(S) \geq (1 - 1/e) \cdot f(S^*)$. In other words, S provides a $(1 - 1/e)$-approximation, or guarantees a lower bound on the quality of solution as compared to the optimal solution.*

Applying to the k-**MMP** problem, the ground set is $\mathcal{P}_M \cup \mathcal{P}_N$, the market share function $MS(\cdot)$ defined in Section 2 maps subsets of $\mathcal{P}_M \cup \mathcal{P}_N$ to real numbers, i.e., the expected market share of products. According to Theorem 5.3, if we can prove that $MS(\cdot)$ is a non-negative monotone submodular set function, then our approximation Algorithm 3 can provide a $(1-1/e)$-approximation. We leaves the proof of these properties in the next subsection, and once we prove them, we have the following theorem.

Theorem 5.4 (Performance Guarantee). *The approximation algorithm 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 5.8, which will be proved in the following subsection, the market share function $MS(\cdot)$ in Equation (3) is non-negative, monotone submodular. So, according to Theorem 5.3, Algorithm 3 provides $(1-1/e)$-approximate solutions. □

5.3. Submodular Market Share Function

Let us consider the market share function $MS(\cdot)$ defined in Section 2. According to the definition of $MS(\cdot)$, it is obviously non-negative, so we seek to prove the monotonicity and submodularity properties. For the ease of presentation, 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 $S_j = S \cup \{p_j\}$, let $pr_i(S) = \sum_{p_j \in S} \Pr(i, j|\mathcal{P}_S)$ denote the probability of the consumer c_i adopting products in S when a set $\mathcal{P}_S$ of products is available in the market. Furthermore, when a set $\mathcal{P}$ of products is available in the market, we define $F\mathcal{C}(p_j|\mathcal{P})$ as the set of consumers that p_j is their farthest product, and recall that $F\mathcal{P}(c_i|\mathcal{P})$ is the set of farthest products from c_i.

One key fact we use in our proof is that by adding a new product, say p_u, only those consumers in $F\mathcal{C}(p_u|\mathcal{P}_u)$ will change their product adoption decisions. Therefore, to calculate the change of market share caused by adding p_u, we only need to consider the consumers in $F\mathcal{C}(p_u|\mathcal{P}_u)$. Mathematically, we have the following proposition.

PROPOSITION 5.5. *Let $\mathcal{P}_S$ be the set of products in the market, by adding a new product p_u into the market, $p_u \in \mathcal{P}_N \setminus \mathcal{P}_S$, the increase of the market share of products in S_u is*

$$MS(S_u) - MS(S) = \sum_{c_i \in F\mathcal{C}(p_u|S_u)} \frac{1}{l} \left[pr_i(S_u) - pr_i(S) \right]. \tag{17}$$

Based on Proposition 5.5, we now proceed to prove the monotonicity and submodularity of the market share function $MS(\cdot)$. First, we prove two lemmas (Lemmas 5.6 and 5.7). Based on these two lemmas, we prove the monotonicity and submodularity properties in Theorem 5.8.

LEMMA 5.6. *Let $S \subseteq \mathcal{P}_M \cup \mathcal{P}_N$ be a set of products, and p_u be another product in $\mathcal{P}_N$, $p_u \in \mathcal{P}_N \backslash S$. For a consumer $c_i \in \mathcal{C}$, if $c_i \in F\mathcal{C}(p_u|\mathcal{P}_{S_u})$, then we have*

$$pr_i(S_u) - pr_i(S) \geq 0. \tag{18}$$

PROOF. Please refer to the appendix. □

LEMMA 5.7. *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 $\mathcal{P}_N$, $p_u \in \mathcal{P}_N \backslash T$. For a consumer $c_i \in \mathcal{C}$, if $c_i \in F\mathcal{C}(p_u|\mathcal{P}_{T_u})$, then we have*

$$pr_i(S_u) - pr_i(S) \geq pr_i(T_u) - pr_i(T). \tag{19}$$

PROOF. Please refer to the appendix. □

THEOREM 5.8. *Suppose consumers adopt products following the distance-based adoption model, then the market share function $MS(\cdot)$ defined in Equation (3) is monotone submodular for the k-**MMP** problem.*

PROOF. We prove the monotonicity property first. To prove the monotonicity property, we need to show

$$MS(S_u) - MS(S) \geq 0 \quad \forall S \subseteq \mathcal{P}_N \cup \mathcal{P}_M, p_u \in \mathcal{P}_N \tag{20}$$

holds, which can be proved by combining the results of Proposition 5.5 and Lemma 5.6.

To prove the submodularity property, according to Definition 5.2, we need to show

$$MS(S_u) - MS(S) \geq MS(T_u) - MS(T) \tag{21}$$

holds $\forall S \subseteq T \subseteq \mathcal{P}_N \cup \mathcal{P}_M$ and $p_u \in \mathcal{P}_N$.

Table IV. Parameters of Synthetic Data

Parameters	*Range*	*Default*
Distance metrics	*DM, NM, PM, RM*	*NM*
m ($\|\mathcal{P}_E\|$)	$1K, 2K, 3K, 4K$	$1K$
l ($\|\mathcal{C}\|$)	$10K, 20K, 30K, 40K$	$10K$
d ($\|\mathcal{A}\|$)	$10, 15, 20, 25$	10
n ($\|\mathcal{P}_N\|$)	$20, 40, 60, 80$	20
k	$2, 3, 4, 5$	2

In the case of $p_u \in S$, Inequality (21) holds since both sides are equal to 0. In the case of $p_u \in T \setminus S$, the right side of the inequality equals 0, while according to the monotonicity, which has been proved, the left side is non-negative. Hence, Inequality (21) also holds. In the case of $p_u \in \mathcal{P}_N \setminus T$, Inequality (21) can be easily proved by combining the results of Proposition 5.5 and Lemma 5.7. Thus, Inequality (21) holds $\forall S \subseteq T \subseteq \mathcal{P}_N \cup \mathcal{P}_M$ and $p_u \in \mathcal{P}_N$. □

6. EXPERIMENTS

We perform experiments on both synthetic datasets and real-world web datasets. We implement our approximation algorithm and the exhaustive search algorithm in C++ and perform experiments on a PC with a 16-core 2.4GHz CPU, 30GB of main memory under the 64-bit Debian 6.0. First, we use synthetic datasets to evaluate the computational efficiency and accuracy of our approximation algorithm. Then, we apply our algorithm on the real-world web datasets to show the impact of different distance metrics, and how to learn distance metrics from some historical sales data and to perform product selection.

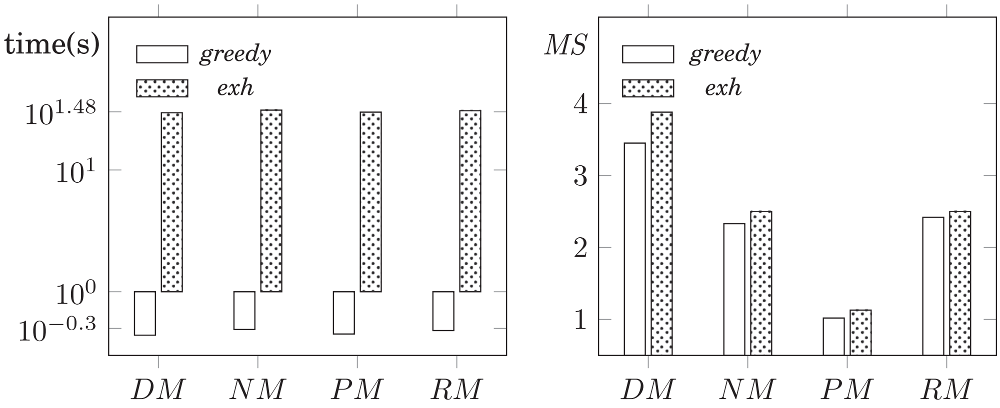
6.1. Speedup and Accuracy

We generate the synthetic datasets using the generator provide by Borzsony et al. [2001]. In a real-world market, products usually do not have high quality on *all* attributes. Instead, they have high quality on some subset of attributes only. For example, a smart phone with a large screen will have high quality on display but low quality on portability. Furthermore, if a product has high quality on most attributes, then the price of this product will be high in general, which indicates low quality on the price attribute. We generate the datasets of products with negative correlation on attributes: Products which have high quality in one attribute tends to have low quality on at least one other attribute. On the other hand, we generate the consumers' requirement of each attribute independently using a uniform distribution.

We compare the running time and the market share between our approximation algorithm (or *greedy*) and the exhaustive search algorithm (or *exh*). We examine the impact of various factors, including the size of datasets (n, m, l), the number of new products we need to select (k), and models using different distance metrics (four distance metrics as introduced in Section 2).

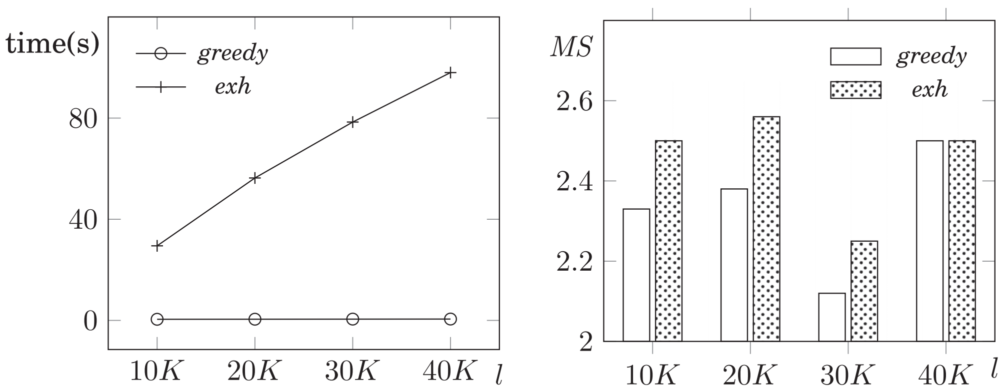
The values of parameters are shown in Table IV. In the following, we examine the impact of each factor by considering the corresponding parameter as a variable and setting other parameters as default value as shown in Table IV. Figures 2–7 depict the results. In each figure, the horizontal axis shows the corresponding parameter which is considered as the variable, while the vertical axis of (a) shows the running time of the two algorithms and the vertical axis of (b) shows the expected market share of the selected products.

—*Impact of distance metrics*
We firstly explore the impact of different distance metrics on the efficiency and accuracy of our greedy algorithm. Figure 2 shows the running time and expected market



(a) running time (s) (b) market share (%)

Fig. 2. Varying distance metrics: *greedy* vs. *exh*.



(a) running time (s) (b) market share (%)

Fig. 3. Varying l: *greedy* vs. *exh*.

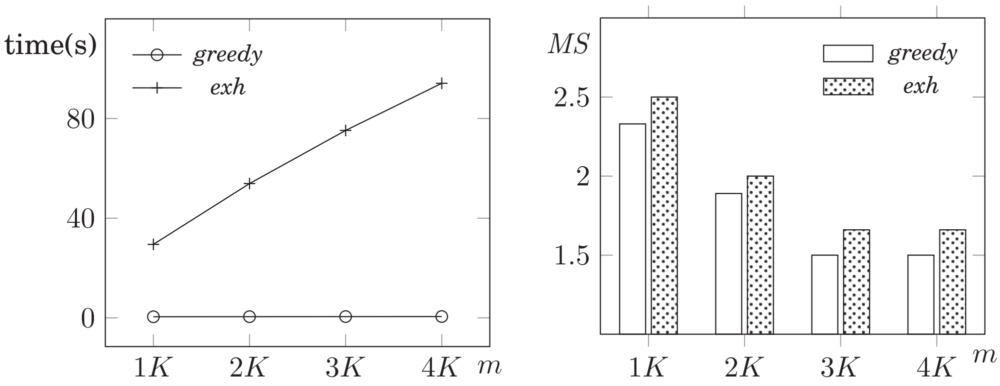
share under adoption models using different distance metrics. One can observe that our greedy algorithm is much faster (about 70 times faster) than the exhaustive search algorithms for all different distance metrics. The expected market share of the products selected by our greedy algorithm is nearly the same with that by exhaustive search and it is insensitive to the distance metrics. The optimal market share, however, are sensitive to the distance metrics. The expected market share varies significantly, which shows that the distance metrics have a high impact on the expected market share.

Since the computational efficiency and accuracy our experiments are similar under all distance models, we only show the results for the *norm distance metric* in the following experiments. Note that if we want to use m' different distance models to describe the potential behavior of consumers, then the running time of the algorithm will be about m' times longer.

—*Impact of the market size*

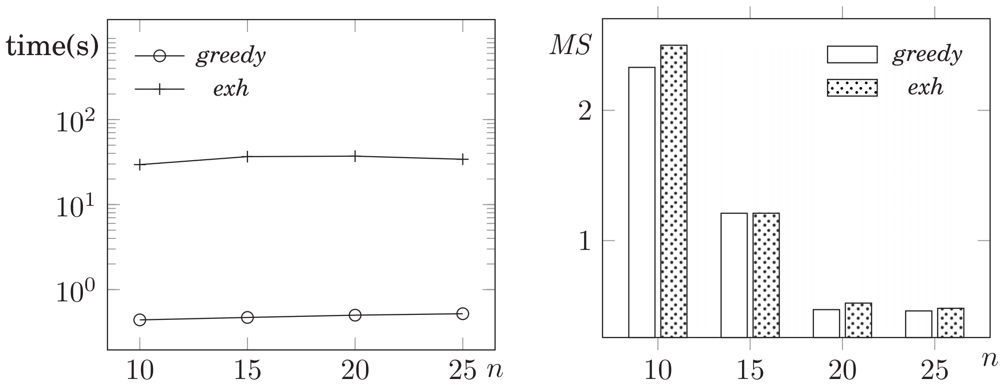
Recall that the size of the market can be measured by the number of consumers l and the number of existing products m.

First, we vary the number of consumers l from 10^4 to 4×10^4. We show the running time and the market share in Figure 3. Then, we set the number of consumers to



(a) running time (s) (b) market share (%)

Fig. 4. Varying m: *greedy* vs. *exh*.



(a) running time (s) (b) market share (%)

Fig. 5. Varying d: *greedy* vs. *exh*.

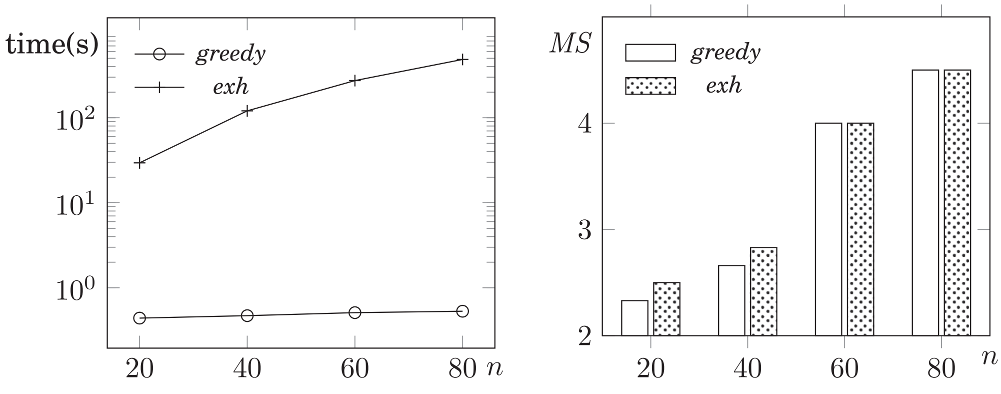
the default number and vary the number of existing products m from 10^3 to 4×10^3. The running time and market share are shown in Figure 4. From the figure, one can observe that the running time of both algorithms increase linearly as the increase of the market size, and our greedy algorithm maintains a high level of quality guarantee regardless of the change of the market size.

—*Impact of d*
We examine the impact of the number of attributes, or d, on the efficiency and accuracy of our greedy algorithm. Figure 5 shows the running time and expected market share when we vary d from 10 to 25. In Figure 5, one can observe that the efficiency and accuracy are insensitive to the number of attributes. It is interesting to observe that increasing the number of attributes decreases the expected market share. Because larger number of attributes indicates stronger consumers' requirement, thus less products will satisfy the consumers.

—*Impact of n and k*
Table V shows the speedup of our approximation algorithm over the exhaustive algorithm. Figures 6(a) and 7(a) show the running time of these two algorithms, where the horizontal axis depicts the variation on parameters n (number of candidate products

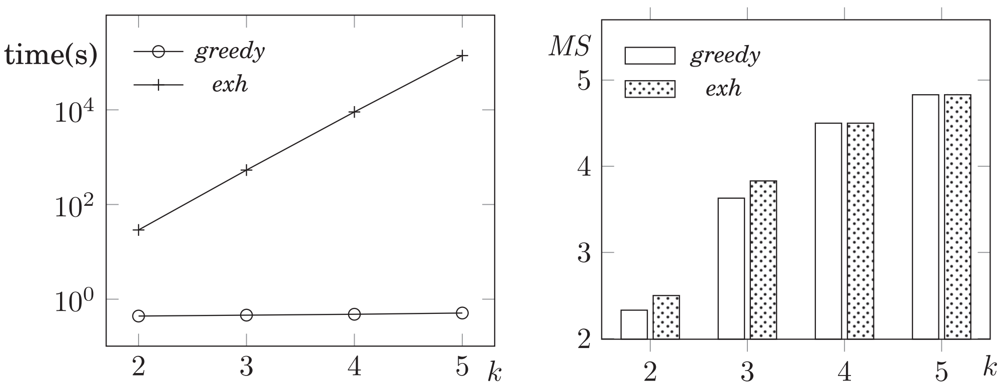
Table V. Speedup: Varying k and n

	$k=2$	$k=3$	$k=4$	$k=5$
n=20	65.89	1,160.37	18,799.17	285,804.08
n=40	256.62	8,111.76	287,626.53	$\approx 1 \times 10^7$
n=60	535.29	26,511.67	$\approx 1 \times 10^6$	$\approx 5 \times 10^7$
n=80	915.38	57,812.33	$\approx 4 \times 10^6$	$\approx 2 \times 10^8$



(a) running time (s) (b) market share (%)

Fig. 6. Varying n: *greedy* vs. *exh*.



(a) running time (s) (b) market share (%)

Fig. 7. Varying k: *greedy* vs. *exh*.

we need to consider) and k (number of products we need to select), while the vertical axis depicts the log scale of the running time, in seconds.

From the table and the figure, one can observe that our approximation algorithm is significantly faster than the exhaustive algorithm: $O(n^k)$ times faster when selecting k products from n candidate products. The speedup is around 285,000 even for a small dataset (i.e., select $k = 5$ products from $n = 20$ candidates). In this case, the running time of exhaustive algorithm is around 40 hours. In the case of selecting five products from $n = 80$ candidates, our conservative estimate on the running time of the exhaustive algorithm is about 10 years. In contrast, the running time of our approximation algorithm for all cases remain in less than 1 second. We also test our approximation

Table VI. Parameters of Web Dataset

# of products	# of consumers	# of attributes
1,605	186,249	8

Table VII. Results of the 2-**MMP** Problem

Distance metrics	*ID of selected products*	*Market share of selected products*
DM	214,566	32.33%
NM	284,214	11.20%
PM	566,350	35.43%
RM	284,214	11.20%
$\hat{\Theta}$	566,284	38.09%

algorithm on a larger dataset, where $m = 1{,}000$, $n = 100$, $l = 1{,}000{,}000$. We select $k = 8$ new products from the 100 candidates. Our approximation algorithm still only takes about 7 minutes.

Figures 6(b) and 7(b) depict the expected market share of the two algorithms. One can observe that our approximation algorithm provides high accuracy: about 0.96 approximation on average as compared with the optimal solution obtained using the exhaustive algorithm. This shows that our algorithm generates results which is much better than the theoretical lower bound guarantee. In fact, the results of the two algorithms are exactly the same for over 80% of all experiments we performed and our approximation algorithm still provides a 0.82 approximation even under the worst case scenario among all experiments.

6.2. Impact of Distance Metrics

In this subsection, we perform experiments on a real-world web dataset, and we aim to show the influence of using different distance metrics.

We extract the *TripAdvisor* dataset from Wang et al. [2010]. Hotels and reviewers of these hotels are considered as products and consumers, respectively, in this dataset. The reviewers rated hotels on seven attributes: value, room, location, cleanliness, front desk, service, and business service. We use the average rating of an attribute as the quality of that attribute for each hotel. We also add the inverse of the average price of the hotel as the eighth attribute, which is normalized in the range of (1, 5). For each consumer, we extract requirement vector as follows. Let $\bar{r}$ be the average rating of a hotel's attribute and r_i be the rating from the consumer c_i. If r_i is lower than $\bar{r}$, it means that c_i has a higher requirement than average, and if r_i is higher than $\bar{r}$, c_i may have a lower requirement than the average. Thus, we set the requirement of c_i as $\bar{r} + (\bar{r} - r_i)$. For example, if $\bar{r} = 3.5$ and $r_i = 4$, then the requirement of c_i will be $3.5 + (3.5 - 4) = 3$. Table VI shows the overall statistics of the dataset.

We select the first 605 hotels as the candidate products and set the remaining 1,000 hotels as the existing products. We apply our approximation algorithm to solve the 2-**MMP** problem using the four distance metrics introduced in Section 2: *DM*, *NM*, *PM*, and *RM*. The results are shown in the first four rows of the second column in Table VII. One can observe that the results vary greatly when we use different distance metrics. This implies the importance of inferring and understanding consumers' adoption behavior.

6.3. Learning Distance Metrics

In the following, we first evaluate the accuracy of our learning method when we have a perfect knowledge on the distance metric of each model in the model set, i.e., we know

Table VIII. Market Share (%)

ID	*DM*	*NM*	*PM*	*RM*	*real-world*
91	0.21	0.13	4.81	0.13	2.93
500	0.40	0.14	0.33	0.07	0.27
517	1.30	49.83	2.10	25.38	13.75
746	1.07	0.79	0.81	11.40	1.87
350	0.68	1.09	3.80	1.09	2.64

exactly what kind of distance metrics the consumers may adopt. Afterwards, we also evaluate our method in case that we don't have a perfect knowledge.

—*Learning with perfect knowledge*
In the following, we assume that the consumers use the four distance metrics we introduced in Section 2, i.e., *DM, NM, PM,* and *RM*. Since we do not have the information about products' real-world market share and consumers' adoption models, we manually set the probability $\hat{\Theta}$ that consumers use the above four distance metrics. Then, we randomly set the distance metric for each consumer according to $\hat{\Theta}$ and estimate the "real-world market share" by enumerating each consumer's choice.

We estimate the probability as Θ using the learning method in Section 3 and compare the *normalized root-mean-square error (NRMSE)* between Θ and $\hat{\Theta}$ to evaluate the accuracy of our learning method. Note that *NRMSE* ranges in $(0, 1)$ and lower value implies higher accuracy.

We present the experimental results in the case that $\hat{\Theta} = (0.1, 0.2, 0.6, 0.1)^T$ and the "real-world market share" of a set $\mathcal{P}'_E$ of five products are known. First, we calculate the expected market share of these products under all the four potential models. The results are shown in Table VIII along with the "real-world market share". Then, by solving the following optimization problem, we can estimate $\Theta = (0.1084, 0.1979, 0.5953, 0.0984)^T$. One can observe that Θ is very close to $\hat{\Theta}$ (*NRMSE* ≈ 0.0099), which indicates a high accuracy of the estimation.

$$\begin{aligned} \text{Minimize} \quad & \left\| \begin{matrix} \theta_1 \cdot 0.21\% & \theta_2 \cdot 0.13\% & \theta_3 \cdot 4.81\% & \theta_4 \cdot 0.13\% & -2.93\% \\ \theta_1 \cdot 0.40\% & \theta_2 \cdot 0.14\% & \theta_3 \cdot 0.33\% & \theta_4 \cdot 0.130.07\% & -0.27\% \\ \theta_1 \cdot 1.30\% & \theta_2 \cdot 49.83\% & \theta_3 \cdot 2.10\% & \theta_4 \cdot 0.1325.38\% & -13.75\% \\ \theta_1 \cdot 1.07\% & \theta_2 \cdot 0.79\% & \theta_3 \cdot 0.81\% & \theta_4 \cdot 0.1311.40\% & -1.87\% \\ \theta_1 \cdot 0.70\% & \theta_2 \cdot 0.15\% & \theta_3 \cdot 1.68\% & \theta_4 \cdot 0.130.15\% & -2.64\% \end{matrix} \right\|_2^2 \\ \text{subject to} \quad & \Theta \geq \mathbf{0}, \quad \theta_1 + \theta_2 + \theta_3 + \theta_4 = 1. \end{aligned}$$

Based on the derived probability Θ, one can forecast the market share of products and make a better product selection decision. The result of the 2-**MMP** problem in this scenario is shown in the last row of Table VIII. For the selected products under each adoption model in Table VIII, we estimate the "real-world market share" and list the result in the last column. One can observe that, the product selection result based on learning the proper weighting of distance metrics achieves a better market share than other distance metrics.

We also select different sets $\mathcal{P}'_E$ of products that we know the market share and examine the *NRMSE*. The results are shown in Figure 8, where the vertical axis is the *NRMSE* of the estimation, the horizontal axis of (a) is n' which is the size of $\mathcal{P}'_E$, and the horizontal axis of (b) is the average variance σ^2 of the expected market share of products in $\mathcal{P}'_E$ under different models when $n' = 5$ and $n' = 20$.

One can observe that our estimation maintains a high accuracy in general. The average accuracy is about 0.035 even in the case that we only know the market share of five products. Furthermore, the accuracy increases exponentially fast when the size

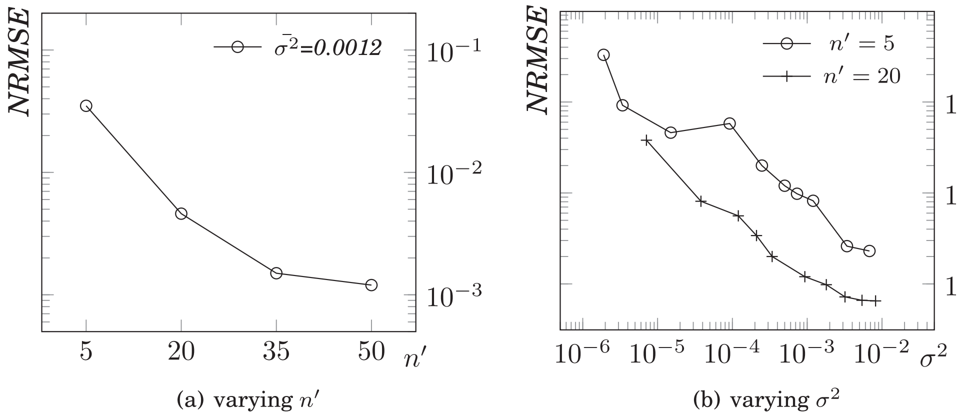


Fig. 8. Accuracy of learning from perfect knowledge.

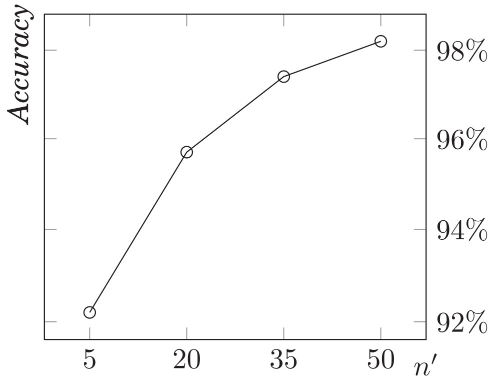


Fig. 9. Accuracy of learning from arbitrary guess.

of $\mathcal{P}'_E$ increases. On the other hand, product sets with larger σ^2 have higher accuracy, which is realistic since if the market share varies slightly under different models, it may be difficult to estimate. We only present one example here, however, we like to note that our results and conclusions are consistent when we vary $\hat{\Theta}$, model set, or any other parameters.

—*Learning with arbitrary guess*
In reality, it is difficult to know all the consumers' behavior. Hence, we probably do not have a perfect knowledge on the distance metrics in the model set. In the following, we estimate the accuracy of our learning method when we do not know the exact consumer behavior. We randomly assign 20 different distance metrics, *excluding* the four distance metrics introduced in this paper, to the consumers. These 20 different metrics include Euclidean norm, maximum norm, norms with different weight on different attributes, and the like. Similarly with learning from perfect knowledge, we estimate the "real-world market share" by enumerating each consumer's choice.

Other than calculating the probability of the 20 different distance metrics we assigned to consumers, we still use *DM, NM, PM,* and *RM* to estimate the probability Θ. We then calculate the market share based on the estimation of Θ and compare the results with "real-world market share". Figure 9 shows the accuracy of our learning method where the horizontal axis n' is the number of products which we know the market share, and the vertical axis shows the average ratio between the estimated market share and the "real-world mark share".

One can observe that the estimation of market share has a high accuracy as compared with the "real-world market share" even though the distance metrics we use are very different from the reality. This shows that even there is no guarantee on the accuracy when using the "wrong" model set, but if the distance metrics in the model set are representative enough for the consumer behavior, the differences between the selected distance metrics and real-world consumer behavior will have an acceptable small influence on the estimation results. Thus, when applying our learning algorithm to the real-world problems, the selected distance metrics need not be exactly the same as the real consumer behavior, and this implies high robustness and practicability of our learning method.

7. RELATED WORK

Product selection

Let us provide some related work on product selection. In Kleinberg et al. [1998], authors formulated a number of microeconomic applications as optimization problems via data mining perspective. Inspired by Kleinberg et al. [1998] and Li et al. [2006] extended the concept of dominance, which is used as skyline operators [Borzsony et al. 2001] to analyze various forms of relationships between products and consumers. A manufacturer can position popular products effectively while remaining profitable by analyzing the dominance relationships. The works in Zhang et al. [2009]; Wan et al. [2009, 2011]; Peng et al. [2012] considered the situation that there exist multiple manufacturers. The authors of Zhang et al. [2009] derived the Nash Equilibrium when each manufacturer modifies its product in a round robin manner to maximize the market share. Wan et al. [2009] aimed to find the most competitive products which are not dominated by any competitors without taking into account the consumers. They extended their work in Wan et al. [2011]; Peng et al. [2012] by considering the consumers' preferences. However, the above papers all aimed to maximize the *number of potential consumers*, which is not equivalent to the *market share* derived in this paper. In fact, potential consumers may not lead to higher market share because different consumers have different probability to adopt new products. Authors in Lin et al. [2013] aimed to find the products with the maximum expected number of total adopters, which is similar with the market share in our paper. But their algorithm could not provide any theoretical performance guarantee. Furthermore, none of the previous works consider the complicated product adoption behavior of consumers. Instead, they assumed that consumers will make random product adoption decisions, which corresponds to a special case of our product adoption model using the *discrete norm*.

Maximization of submodular functions

Submodular functions have properties which are very similar to the convex and concave functions. The authors of Cornuejols et al. [1977] and Nemhauser et al. [1978] showed that a natural greedy hill-climbing strategy can achieve a provable performance guarantee for a problem of maximizing a non-negative monotone submodular function: at least 63% of optimal. Due to the generality of this performance guarantee, this results has found applications in a number of areas, e.g., discrete optimization [Nemhauser and Wolsey 1988], materialized view [Harinarayan et al. 1996], and influence maximization [Kempe et al. 2003].

8. CONCLUSIONS

In this work, we present the problem of finding the k-**MMP** under a distance-based adoption model. Our adoption model is general in that we can use different distance metrics to describe various consumers' adoption behaviors. Given some historical datasets on market share, we propose a learning method to select the appropriate distance metrics to describe consumers' production adoption behavior. We prove that the k-**MMP** problem is *NP-hard* in general. We propose a polynomial time approximation algorithm to solve the k-**MMP** problem. Using the submodularity analysis, we formally prove that our approximation algorithm can guarantee a $(1 - 1/e)$-approximation as compared to the optimal solution. We compared our approximation algorithm with the exhaustive search algorithm on the synthetic datasets. The results showed that our approximation algorithm can achieve $O(n^k)$ times speedup when selecting k products from n candidates. Furthermore, the solution quality of our algorithm is about 96% on average, which is much higher than the theoretical lower bound. We also perform experiments on the real-world web datasets to show the crucial impact of different distance metrics and how we can improve the accuracy of product selection using our distance metric selection method.

APPENDIX

—*Proof of Theorem 4.3*:

PROOF. The *NP-hardness* proof can be achieved by transforming an *NP-hard* problem, called the *top-k Representative Skyline Product* (top-k RSP) [Lin et al. 2007], to a special case of the k-**MMP** problem.

Let us state the top-k RSP [Lin et al. 2007]. Given a set U of points and a positive integer k, compute a set S of k skyline points such that the number of points dominated by these k points is maximized. A point $p = (p[1], p[2], \ldots, p[d])$ *dominates* another point $q = (q[1], q[2], \ldots, q[d])$ iff $p[i] \geq q[i]$, $\forall 1 \leq i \leq d$ and there exists at least one dimension d such that $p[d] > q[d]$, and we denote this as $p \succ q$. Consequently, the *skyline* point is defined as follows. Given a set U of points, the skyline points of U are the set of $S \subseteq U$ points which are not dominated by any points in U.

Given an instance of top-k RSP problem, we construct an instance of k-**MMP** problem, which can be carried out as follows. Set $\mathcal{P}_E = \varnothing$, i.e., $m = 0$. Let $\mathcal{P}_N$ be the set of skyline points in U, and $\mathcal{C}$ be the rest, i.e., $\mathcal{C} = U \backslash \mathcal{P}_N$. Note that in general, the concept of *dominance* is different from *product satisfiability* as stated in Definition 2.2. Formally, we have $p_j \succ c_i \Rightarrow p_j \succsim c_i$, but $p_j \succsim c_i \nRightarrow p_j \succ c_i$. However, if $p_j \succsim c_i$ but $p_j \not\succ c_i$, then the quality vector of p_j is exactly the same with the requirement vector of c_i, i.e., p_j and c_i have the same location in the d-dimensional space. But in our construction, the product points are skyline points while the consumer points are not, so there *does not exist* such kind of c_i and p_j pairing in our construct. Therefore, we can treat *dominance* and *product satisfiability* to be the same in this instance.

Let P be the set of k products we select from $\mathcal{P}_N$. In this case, since there is no existing product, so if a consumer has any satisfactory product in P, the consumer will adopt one unit of products in P, otherwise, 0. As a result, the expected number of adopters is equal to the number of consumers who have satisfactory products in P. In another word, $MS(P)$ is equal to the total number of points dominated by the skyline points in P divide by the total number of non-skyline points. Since the total number of non-skyline points is fixed, the result of the corresponding top-k RSP problem is also the result of this instance of the k-**MMP** problem.

Therefore, any instance of the top-k RSP problem can be transformed to an instance of the k-**MMP** problem. Since the top-k RSP problem has been proved to be an *NP-hard* problem, the k-**MMP** problem is also *NP-hard*. □

—*Proof of Lemma 5.6*:

PROOF. To simplify the proof, we define the following notations. Let $\sigma = |F\mathcal{P}(c_i|\mathcal{P}_{S_u})|$ and $s = |F\mathcal{P}(c_i|\mathcal{P}_{S_u}) \cap S_u|$, where $\sigma \geq s \geq 1$. Let $d = d_{i,j}$, where $p_j \in F\mathcal{C}(c_i|\mathcal{P}_S)$. Since $c_i \in F\mathcal{C}(p_u|\mathcal{P}_{S_u})$, we have $d_{i,u} \geq d$. If $d_{i,u} > d$, then c_i will adopt p_u with probability 1, i.e., $pr_i(S_u) = 1$. While $pr_i(S) \leq 1$, so Inequality (18) holds. If $d_{i,u} = d$, then $F\mathcal{P}(c_i|\mathcal{P}_{S_u}) = F\mathcal{P}(c_i|\mathcal{P}_S) \cup \{p_u\}$, so $|F\mathcal{P}(c_i|\mathcal{P}_S)| = \sigma - 1$. Similarly, we have $|F\mathcal{P}(c_i|\mathcal{P}_S) \cap S| = s - 1$. When $\sigma = 1$, $F\mathcal{P}(c_i|\mathcal{P}_S)$ is an empty set, which means p_u is c_i's only choice, the situation is the same with the case of $d_{i,u} = d$. So we only need to consider the case when $\sigma > 1$. Bring the notations into Inequality (18), we have

$$pr_i(S_u) - pr_i(S) = \frac{s}{\sigma} - \frac{s-1}{\sigma - 1} = \frac{\sigma - s}{\sigma(\sigma - 1)} \geq 0, \tag{A1}$$

which can be proved by observing that $\sigma \geq s$ and $\sigma > 1$. □

—*Proof of Lemma 5.7*:

PROOF. Because $S_u \subseteq T_u$, p_u has more competitors when a set T_u of products is available in the market. As a result, $F\mathcal{C}(p_u|T_u) \subseteq F\mathcal{C}(p_u|S_u)$. Since $c_i \in F\mathcal{C}(p_u|T_u)$, we have $c_i \in F\mathcal{C}(p_u|S_u)$. Follow the same notations in the proof of Lemma 5.6, let us consider the case of $\sigma = 1$ first. In this case, $pr_i(S_u) = 1$, $pr_i(S) = 0$, so the left side of Inequality (19) equals to 1. While the right side of the inequality is obviously no larger than 1, so the Inequality (19) holds. Now, let us consider the case when $\sigma > 1$. According to the proof of Lemma 5.6, we have

$$pr_i(S_u) - pr_i(S) = \frac{s}{\sigma} - \frac{s-1}{\sigma - 1}. \tag{A2}$$

Let d_T and d_S denote the distance between c_i and the products in $F\mathcal{P}(c_i|\mathcal{P}_T)$ and $F\mathcal{P}(c_i|\mathcal{P}_S)$, respectively, where $d_{i,u} \geq d_T \geq d_S$. In the case of $d_{i,u} > d_S$, then the left side of Inequality (19) equals to 1, so the inequality holds. Thus, the remaining thing is to prove the inequality holds when $d_{i,u} = d_T = d_S$. In this case, $F\mathcal{P}(c_i|\mathcal{P}_{S_u}) = F\mathcal{P}(c_i|\mathcal{P}_S) \cup \{p_u\}$, $F\mathcal{P}(c_i|\mathcal{P}_{T_u}) = F\mathcal{P}(c_i|\mathcal{P}_T) \cup \{p_u\}$, and $F\mathcal{P}(c_i|\mathcal{P}_S) \subseteq F\mathcal{P}(c_i|\mathcal{P}_T)$. Let $\delta = |F\mathcal{P}(c_i|\mathcal{P}_{T_u})| - |F\mathcal{P}(c_i|\mathcal{P}_{S_u})|$, then it follows that $\delta \geq 0$, $\sigma + \delta = |F\mathcal{P}(c_i|\mathcal{P}_{T_u})|$, $s + \delta = |F\mathcal{P}(c_i|\mathcal{P}_{T_u}) \cup T_u|$. Thus, we have

$$pr_i(T_u) - pr_i(T) = \frac{s+\delta}{\sigma+\delta} - \frac{s+\delta+1}{\sigma+\delta+1}. \tag{A3}$$

According to Equations (A2) and (A3), we can derive Inequality (19) as follows.

$$\begin{aligned} &\text{Inequality (19) holds} \\ &\Leftrightarrow \frac{s}{\sigma} - \frac{s-1}{\sigma-1} \geq \frac{s+\delta}{\sigma+\delta} - \frac{s+\delta-1}{\sigma+\delta-1} \\ &\Leftrightarrow \frac{\sigma - s}{\sigma(\sigma-1)} \geq \frac{\sigma - s}{(\sigma+\delta)(\sigma+\delta-1)}. \end{aligned} \tag{A4}$$

Hence, we only need to show that Inequality (A4) holds, which can be proved by observing that $\sigma > 1$, $\delta \geq 0$ and $\sigma \geq s$. □

REFERENCES

S. Borzsony, Donald Kossmann, and Konrad Stocker. 2001. The skyline operator. In *Proceedings of the 17th International Conference on Data Engineering*. IEEE, 421–430.

Timothy M. Chan, Kasper Green Larsen, and Mihai Pătraşcu. 2011. Orthogonal range searching on the RAM, revisited. In *Proceedings of the 27th Annual Symposium on Computational Geometry*. ACM, 1–10.

Gerard Cornuejols, Marshall L. Fisher, and George L. Nemhauser. 1977. Location of bank accounts to optimize float. *Manag. Sci.* 23, 8 (1977), 789–810.

Philip E. Gill, Walter Murray, and Margaret H. Wright. 1981. Practical optimization. Academic Press, New York.

Venky Harinarayan, Anand Rajaraman, and Jeffrey D. Ullman. 1996. Implementing data cubes efficiently. In *ACM SIGMOD Record*, Vol. 25. ACM, 205–216.

David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 137–146.

Jon Kleinberg, Christos Papadimitriou, and Prabhakar Raghavan. 1998. A microeconomic view of data mining. *Data Min. Knowl. Discovery* 2, 4 (1998), 311–324.

Cuiping Li, Beng Chin Ooi, Anthony K. H. Tung, and Shan Wang. 2006. Dada: A data cube for dominant relationship analysis. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*. ACM, 659–670.

Chen-Yi Lin, Jia-Ling Koh, and Arbee L. P. Chen. 2013. Determining k-most demanding products with maximum expected number of total customers. *IEEE Trans.Knowl. Data Eng.* 25, 8 (2013), 1732–1747.

Xuemin Lin, Yidong Yuan, Qing Zhang, and Ying Zhang. 2007. Selecting stars: The k most representative skyline operator. In *IEEE 23rd International Conference onData Engineering (ICDE 2007).* IEEE, 86–95.

George L. Nemhauser and Laurence A. Wolsey. 1988. *Integer and Combinatorial Optimization*. Vol. 18. Wiley, New York, NY.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions - I. *Math. Program.* 14, 1 (1978), 265–294.

Yu Peng, Raymond Chi-Wing Wong, and Qian Wan. 2012. Finding top-k preferable products. *IEEE Trans. Knowl. Data Eng.* 24, 10 (2012), 1774–1788.

Qian Wan, Raymond Chi-Wing Wong, Ihab F. Ilyas, M Tamer Özsu, and Yu Peng. 2009. Creating competitive products. In *Proceedings of the VLDB Endowment*, Vol. 2. VLDB Endowment, 898–909.

Qian Wan, R. C.-W. Wong, and Yu Peng. 2011. Finding top-k profitable products. In *IEEE 27th International Conference on Data Engineering (ICDE)*. IEEE, 1055–1066.

Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. Latent aspect rating analysis on review text data: A rating regression approach. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 783–792.

Zhenjie Zhang, Laks V. S. Lakshmanan, and Anthony K. H. Tung. 2009. On domination game analysis for microeconomic data mining. *ACM Trans. Knowl. Discovery Data (TKDD)* 2, 4 (2009), 18.

Received October 2014; revised March 2015; accepted March 2015