5-2019

Towards personalized data-driven bundle design with QoS constraint

Mustafa MISIR
Singapore Management University, mustafamisir@smu.edu.sg

Hoong Chuin LAU
Singapore Management University, hclau@smu.edu.sg

Citation

MISIR, Mustafa and LAU, Hoong Chuin. Towards personalized data-driven bundle design with QoS constraint. (2019). *Business and consumer analytics: New ideas*. 865-909.

Available at: https://ink.library.smu.edu.sg/sis_research/4680

Towards Personalized Data-Driven Bundle Design with QoS Constraint

MISIR, Mustafa*

Nanjing University of Aeronautics and Astronautics, College of Computer Science and Technology, Nanjing, Jiangsu, China

LAU, Hoong Chuin

Singapore Management University, School of Information Systems, Singapore, Singapore

*This work was done while the author was a post-doc at the Singapore Management University.

Published in Moscato, Pablo, de Vries, Natalie J. (Eds.), *Business and Consumer Analytics: New Ideas*. Cham: Springer, 2019. Chapter 23.

DOI: 10.1007/978-3-030-06222-4_23

Accepted version

Abstract

In this paper, we study the bundle design problem for offering personalized bundles of services using historical consumer redemption data. The problem studied here is for an operator managing multiple service providers, each responsible for an attraction, in a leisure park. Given the specific structure of interactions between service providers, consumers and the operator, a bundle of services is beneficial for the operator when the bundle is underutilized by service consumers. Such revenue structure is commonly seen in the cable television and leisure industries, creating strong incentives for the operator to design bundles containing lots of not-so-popular services. However, as customers might choose to bypass a bundle completely if it is not sufficiently attractive, we need to impose a quality of service (QoS) constraint on the lower bound of the perceived attractiveness. In this paper, we make two major contributions (1) recognizing the inherent differences in customer preferences, we propose an approach for detecting different user classes, and for each user class, make an appropriate bundle recommendation; and (2) in order to make the bundling scheme even more adaptive to unknown customer preferences, we propose a dynamic bundling strategy, which allows customers to “trade in” any number of undesirable services dynamically so that they can be replaced by an alternative set of services. A step to generate fixed or static bundles is also studied. The pros and cons of different bundling strategies are illustrated using a real-world dataset collected from a large leisure park operator in Asia that manages a large collection of attraction providers.

Keywords

Bundling, Dynamic recommendation, Static recommendation, Customer segmentation, Recommender systems, Matrix factorization

23.1. Introduction

Product development [27] is an important part of service innovation and marketing today. While product development could refer to designing new products from scratch or releasing an upgraded/modified version of an existing product, it could also refer to combining existing products as packages that we see in travel, financial, healthcare, information and telecommunications services. The latter process is commonly known as product bundling, which may entail both design and pricing aspects. For example, bundling of services are commonly found in vacation packages: bundles of airline tickets with hotels and car rental, as well as TV channel subscriptions.

This work focuses on the bundle design [19] problem, which is a well-studied problem (see related work in Sect. 23.2.1). The bundle design problem in this chapter is concerned with designing bundles of attractions for a large theme park. Typically in a theme park, there are many attractions not all of which are interesting to a given visitor—much like not all TV channels are appealing to a particular viewer. Hence, it is an interesting problem to be able to offer a given visitor an appropriate bundle of attractions that fits his/her profile. This bundle is then sold as a ticket comprising the subset of attractions that can be redeemed by the ticket holder.

Our work falls under a larger category of consumer analytics for the hospitality industry. It thus offers a methodology for the travel industry in terms of generating good bundle options for tourists in general. With the increasing number of both domestic and international travellers every year, there is a need to better understand demands of customers in order to better serve and provide a more enjoyable experience. In the past, many business decisions were made based on the "gut feeling" of decision makers, which might not always be the best as important information is not usually readily available. With the advancement of technology, especially with the emergence of data analytics, these critical business insights may be obtained from data, which allows these decision makers to make better business decisions in a timely manner.

Our goal is to develop strategies for designing bundles for profit maximization with a quality of service (QoS) constraint for the park operator. The target bundle design problem has unique characteristics differentiating from the existing bundling problems in terms of profit. The profit gained from a customer/visitor depends on his/her visit preferences, thus per-visitor profit varies among the visitors who bought the same bundle with the same price. While designing bundles, we explicitly consider the existence of multiple user classes/segments, and disparate bundling strategies typically exist for different user classes. The identification of user classes is achieved through analysing historical consumption patterns when customers were free to choose among all available services. These consumption patterns turn out to be the major difficulty in designing profitable bundles, as the perceived attractiveness of a particular service usually depends on the other included services in the bundle. The QoS constraint is used to maintain a certain level of this interactive attractiveness while designing profitable bundles.

We propose a data-driven approach benefiting from recommender systems (RSs) to offer personalized bundles of services. RSs refer to a broad range of software systems or tools that provide suggestions for items that a user might be interested in consuming [40]. The underlying techniques of RSs aim to produce high-quality prediction on ratings users might have for potential candidate items. To achieve such an objective, it is essential to have ways to reliably predict users' own preferences, which usually implies that a RS must have access to either a user's past choices or consumptions. Considering a user's own preferences as basis, a RS can be designed to generate its predictions using either other user's choices (the collaborative filtering approach [50] which is discussed in Sect. 23.2.2) or just candidate item's description/content (the content-based filtering approach [53]). Given the huge online footprint of users, service (e.g. retail and entertainment) providers have access to large amounts of user service consumption records. With such information, highly effective personalized RSs can be created for a wide variety of items including but not limited to movies, books, music, research papers and even friendships.

The target bundle design problem relates to the three-tier business model shown in Fig. 23.1. Examples of such business models include cable networks and leisure parks. The first is the tier of customers, who consume services from the given bundle based on their own preferences and limitations, like leisure park visitors; second, the tier of service providers who rent resources from the operator and provide services to customers, like the service provider of a roller coaster; and third, the operator, who owns the resources required for running services (e.g. land for physical services, or bandwidth for digital services) like the owner of the whole leisure park. To receive services from providers, customers can either pay the full rate directly to providers or they may purchase a pre-constructed bundle containing that service from the operator. When purchasing a bundle, a customer needs to pay the bundle price upfront (which is usually discounted from the total bundle value), but does not need to pay again for any service included in the bundle. As customers do not pay providers directly, providers will seek reimbursement at a pre-determined rate from the operator using verifiable service records. Table 23.1 gives a numerical example of the interaction between operator, services providers and visitors. Consider that a $p = 40$\$ bundle of four attractions are bought by four visitors. Whenever one of these visitors visits a particular attraction, a predetermined cost occurs to the operator and assumes that these four attractions cost (c_j) 17\$, 13\$, 8\$ and 27\$, respectively. Thus, although each visitor bought the same bundle with the same price, the operator's profit ($p - c_j$) from each visitor varies, i.e. 23\$, 27\$, 32\$ and 13\$.

Fig. 23.1 A three-tier business model including customers, the service providers and the operator

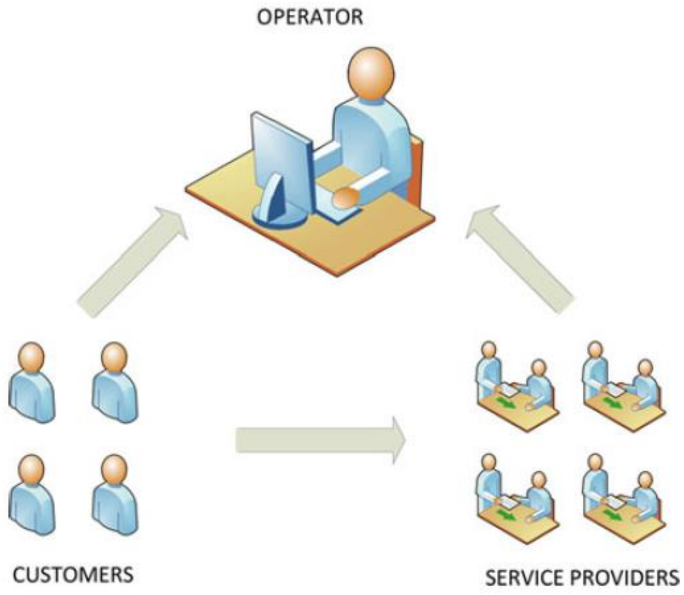


Table 23.1 An example of four visitors who bought a bundle of four attractions where each attraction is managed by a service provider (+: visited, −: unvisited)

	a_1	a_2	a_3	a_4	Price (p)	Cost (c_j)	Profit ($p - c_j$)
v_1	+	−	−	+	40$	17$	23$
v_2	+	+	−	−	40$	13$	27$
v_3	−	+	−	−	40$	8$	32$
v_4	−	+	+	+	40$	27$	13$

We are interested in addressing this problem of designing service bundles for the operator. The operator receives income from two sources: (1) the fixed income from renting resources to service providers, and (2) the variable income from selling bundles. As the operator receives full payment for the bundle upfront, and only needs to reimburse a service provider if a customer uses that service, the operator actually enjoys additional income if a customer buys the bundle but utilizes relatively few services. Thus, solely from the perspective of the operator, having bundles of unpopular services seems like a profitable strategy. However, from the service providers' perspective, each attraction should be visited as much as possible so that each service provider who is responsible from each attraction can make a high profit. Otherwise, the service providers will not be willing to rent the attractions considering that they will not be able to reach their expected profit. Consequently, the operator will face the similar problem of making limited profit or losing money due to the issues of the service providers. Yet, both the operator's and the service providers' financial situation mainly depend on the park visitors. If the park visitors enjoy their time in the park through trying various attractions, each attraction can have a profitable business. This results in an increasing interest of renting the attractions by the service providers, which help the operator to make more income from the attractions' rents. This means that when constructing a service bundle, the operator may not simply include all the unpopular services, as the customers will simply refuse to buy the bundle if it is not attractive enough. We view this requirement as the Quality-of-Service (QoS) constraint the bundle needs to satisfy.

After formally formulating the bundle design problem that provides the mathematical foundation for quantifying the trade-off between operator profit and QoS constraint, we make two major contributions in designing the RS for the bundle design problem. First, we design a *static* segment-specific bundling strategy, in which a fixed bundle is constructed for a given customer (or a customer group) according to the segment s/he belongs. The static bundling falls into the category of the traditional bundle design. Having a certain level of flexibility of the bundles can make the bundling idea even more attractive for the customers. Thus, secondly, in order to grant even more flexibility to the consumer, we propose a *dynamic* bundling strategy where a customer has the option to choose to *trade in* any services s/he does not intend to utilize in the current bundle, and receive an alternative replacement which s/he may choose to keep or skip.

The remainder of the chapter is organized as follows. Section 23.2 provides reviews both bundle design and collaborative filtering. The bundle design problem studied here is explained in Sect. 23.3. The design details of the data-driven bundle design approach are provided in Sect. 23.4. Section 23.5 presents computational results using a real-world leisure park data. The conclusions and suggestions for future research are discussed in Sect. 23.6.

23.2 Background

The main goal of *"bundling"* is to deliver more profitable products in bundles than selling each of the products in a separate way. *Collaborative filtering* is a field of recommender systems. It is motivated to help group preferences. If two users have common preferences on certain items, their preferences are expected to be similar on other items and thus inform predictions.

These two subjects are now related in the literature. Recommender systems have also been used to address the bundling problems. In [59], a recommender system was devised to offer *k* best packages by extending the item-based recommendation idea using approximation algorithms. A graphical model was proposed to detect the consumers' unknown preferences that are used to predict user-specific best possible TV channel bundles in [17]. A bundle recommender system that reduces the item set first for discovering an optimal bundle for e-commerce was designed in [66]. Crowdsourced data was incorporated to deliver personalized travel packages in [61, 62]. In [9], a collaborative filtering based recommender system that utilizes personalized demand functions and price modelling was developed, focusing on both suppliers and consumers. Additionally, various studies [30, 51] were performed to recommend travel packages from a set of existing ones.

This section covers some fundamentals on bundling and collaborative filtering as related to our present work.

23.2.1 Bundling

The purposed goal of "bundling" comes with some advantages [44] including price discrimination, cost saving and potential entry deterrence. The goal is aimed to be achieved by attracting and satisfying consumers in terms of experiencing bundles and their costs. These objectives have been investigated under three perspectives in the literature [55]. These perspectives relate to *suppliers*, *consumers* and *competition*.

Suppliers seek ways to decrease all kinds of costs such as inventory costs [18] and marginal costs [54] while increasing profits and sales. In [20], the profitability of bundling was examined for a supplier that leads the complete market, considering the negotiations between the intermediaries and a competing firm. Another profitability work [16] was carried out under different conditions including the consumers' preferences, product diversity and rival existence. In [58], the supplier selection problem was studied from a bundling perspective for notebook manufacturers in Taiwan. The goal is to determine the best possible bundle of suppliers for notebook production depending on various cost and quality measures. The effects of heterogeneity of the bundled products with a risk analysis were investigated in [44]. In [37], a constrained-based adaptive bundling strategy is proposed to offer product bundles by taking into account the newly introduced constraints/rules and the changes on products' availability. Bundles of information products were analysed from the suppliers' perspectives with the products' complementarities and substitutabilities in [36]. In [34], bundles of sensor data, referring to applications of the "Internet of Things", coming from multiple suppliers were offered with a pricing strategy. Pricing strategies on mobile tariffs as bundles were discussed in [12].

Consumers look for bundles to pay less when the bundled products are separately bought and to consider bundles that are interesting in terms of the correlation/complementarity between the bundled items. The way of discounts offered by bundles of automobiles with optional extras was studied considering the consumers' feedback in [25]. In [1], the factors affecting the consumers' decisions on buying a particular bundle were studied under the light of the bundle size, the bundles' uniqueness and similarities. The relation between the decrease on the consumers valuations for the information goods and the number of goods was examined in [22]. A consumer preference analysis on choosing between bundles and single products was performed in terms of search and assembly costs in [24]. The consumers' evaluations were analysed on the individual products of a bundle considering price discount and product complementarity in [46]. The bundles in the telecom industry of Turkey were studied to understand the actions of consumers on buying bundles and their future bundle choices in [52]. In [13], the consumers' behaviour was reviewed on their food bundle preferences focusing on fruits and vegetables, relating to their health issues.

Competition [8] is another factor that needs to be taken into account if required. From this perspective, bundling studies are studied for the *monopolist*, *duopolist*, *oligopolist* and *perfect competition* environments. Monopoly refers to the markets where there is only one supplier for a particular product. For the duopolist markets, there are two competing suppliers for a single product. In the case of oligopoly, more than two strong yet limited number of suppliers are available. The perfect competition occurs when many suppliers and consumers exist. To give a number research works concentrating on the competition aspect, in [54], bundle design and pricing were studied for the monopolist settings. In [6], the effects of bundling on social welfare were studied for monopolists. In [15], an equilibrium theory is studied for the profitability of the bundles considering the duopoly and perfect competition markets. For the oligopoly markets, the effects of discounts on the bundled products considering the products' interrelations were analysed in [21]. An analysis for two-good bundle pricing is performed under oligopoly in [45].

Referring to the form of bundling [49], the studies could be classified as pure or mixed bundling [35]. Pure bundling is for the firms that sell only bundled products while mixed bundling [35] is valid when a firm sells both bundled and separate products. For the case when there is no bundling, "unbundling" is the term used in the literature.

In this chapter, we study the bundle design problem of a multi-product monopolist (no competition) for pure bundling, yet also applicable to mixed bundling, without bundle-related supplier costs, where product interactiveness are taken into account.

23.2.1.1 Item Interactivity

One of the critical aspects for bundle design is interactiveness of items. Interactiveness is considered in three ways, namely complements, substitutes and independent. Among them, complementarity and substitutability are studied mostly as interactiveness indicators. The traditional approach to specify whether two products are complementary or substitutable is to check the sales of products regarding their price changes. For two complementing products, a decrease in the price of one product is expected to cause an increase in the sales of the other product. For two substitutable products, however, increase in the price of one product is expected to cause an increase in the sales of the other product. In other words, the complementary products are interesting together for the consumers while the substitutable products are expected to be sold as alternatives [64]. For instance, a bike and a bike tyre can be considered complementary while two different bikes are substitutable since either one of them could be bought. Yet, changes on the prices and sales are unable to provide an ultimate criterion to talk about complementarity and substitutability for a given pair of products [47]. It is possible to see some products that are complementary for some consumers while they are substitutes for some others [28]. Following the same bike example, a road bike and a mountain bike can be complementary if a consumer has a particular interest of using them for different purposes, i.e. driving on a regular road or off-road driving. These bikes can be substitutable for another consumer if the only purpose is riding a bike.

The products that are complementary for some consumers, while substitutes for some others, require more complex bundling approaches than in the case where it exists a strict interactiveness difference. The bundling problem we are working on has an additional characteristic than when complementarity and substitutability can be favourable or unfavourable, is unknown. This makes our problem even more challenging. However, the bundles of complementing products (a detailed discussion is given in Sect. 23.5.1) are usually assumed more attractive to the customers than the bundles of substitutable products in the existing studies.

23.2.1.2 Size of the Bundles

Although the effects of bundles' sizes are a crucial subject to research, most of the bundling studies focus on the cases where bundle size is only two [14, 32]. In spite of the strong conclusions derived in these papers, working on the smallest bundles limits the applicability of these studies to the real-world scenarios. Unlike these studies, thousands of information goods are considered to form bundles in [7]. They investigated various topics including market segmentation, interactiveness and profit analysis regarding the bundle size.

Related to this chapter, in [33], the bundling operation is handled by generating a Markov Random Field (MRF) based on the visit transition frequencies of a given dataset. MRF is an undirected graphical model with Markov property. For our target problem domain, each node refers to an attraction and edges between nodes indicate the relation between attractions. Attractions that are disconnected or without a directly connecting edge are considered independent or unrelated. In other words, unconnected attractions are the ones which are not visited together or visited together a few times only. A MRF generated based on a relatively large historical visit dataset is able to reflect popular attractions and their level of popularity to be visited together with other attractions. Weights calculated for attraction subsets indicate how strong this popularity level is. These weights are also used to evaluate the popularity of each attraction when a set of certain attractions available in the form of conditional probabilities. In the aforementioned study, this information is utilized as the indicator of attractions' attractiveness. While attractiveness level shows the probability of an attraction being visited, it is used to calculate the cost that incurs to the main leisure park operator. Here, suggesting highly attractive bundles means that visitors will be highly satisfied while the operator will pay a large amount to the service providers and vice versa. Due to this inverse relationship between attractiveness and cost, the problem is designed as the knapsack problem [31]. The only constraint is set as the attractiveness level so that a resulting bundle should have a certain attractiveness level at least. Since the attractiveness of attractions is not constant or fixed but varies depending on the other attractions included in the bundle, the knapsack problem is considered with interactiveness.

23.2.2 Collaborative Filtering

Collaborative filtering (CF) [50] became a popular field of research mainly after the Netflix challenge [10]. The aim of this competition was to predict a set of users' preferences on a group of movies in the form of ratings/scores when limited rating data is available. Similarly, the Amazon.com [29] like datasets have been used for evaluating various CF methods on customer-item matrices involving the customers' item preferences. In those datasets, there is no dependent relation among both the users and items. Differently, in our case, visitors can be dependent when they move in groups and attractions are highly correlated which affects both revenue and QoS. In addition, one of the tested data forms here consists of both ordinal and availability information together while the current CF literature focuses on single-aspect data like movie ratings as in Netflix.

The motivation behind CF is related to the similarity levels between different users with respect to their preferences. If two users have common preferences on certain items, their preferences are expected to be similar on other items. Thus, user-based similarity is the straightforward choice for CF. Determining similarities between items [41] have also been used to perform predictions. A prediction process is concerned with either *matrix completion* or *cold start* [42]. The matrix completion task is targeted when all the users and items are known at some level. In other words, each user should have his/her score at least on one item and each item should be evaluated at least by one user. In the cold start case, the goal is to make predictions for unknown or new users and new items.

23.2.2.1 Memory-Based and Model-Based Collaborative Filtering

Existing CF algorithms are studied under two categories including *memory-based* CF and *model-based* CF. The memory-based methods predict unknown elements directly using available sparse data. The nearest-neighbour algorithm is the primarily studied memory-based approach. The idea is to determine missing matrix values by using a predetermined number of similar users or items.

Model-based algorithms build a model which approximates to a given matrix. Matrix factorization (MF) is the primary technique used for this purpose, mostly via optimization. The memory-based CF techniques are very effective in spite of their simple designs. However, they are likely to deliver worse performance than the model-based CF approaches when the data provided is highly sparse [50].

Matrix factorization is basically a mathematical process to derive two matrices from a given matrix where the multiplication of these two matrices relates to the original matrix. Singular value decomposition (SVD) [11] is one of the well known matrix factorization methods used in various CF studies. The main issue with this kind of pure mathematical approaches is the requirement of having a full matrix. Thus, a pre-processing step is needed to complete a sparse matrix first. The other family of MF algorithms define the factorization process from an optimization perspective. The goal is to find two matrices that gradually approximates to a given matrix. While earlier methods like SVD deliver a perfect factorization, the second group of methods usually provide near-perfect or near-optimal factorization. Thus, they are also referred to as *"matrix approximation"*.

23.2.2.2 Matrix Approximation and Factorization Methods

The matrix approximation methods have been used on two different types of matrices. The first matrix type involves both negative and positive real values. The basic strategy to address matrix completion is minimizing the sum of squared error between the original matrix and the predicted matrix. Gradient descent is used to perform this optimization task by updating two factorized-matrices in an alternating manner. Besides this basic implementation, Maximum-Margin Matrix Factorization (MMMF) [48] emulates SVM by considering multiple classification tasks and accommodating the hinge loss. Semidefinite programming is used to realize the whole optimization process. Due to the scalability issues of semidefinite programming, a gradient-based optimization technique was proposed to speed up MMMF in [38]. Unlike these methods, probabilistic MF [43] optimizes the posterior distribution on the factorized-matrices. In order to deliver better factorization performance, *tensor factorization* [56] has been studied to add content-based user or item specific data as additional dimensions to the traditional two-dimensional matrices. The second matrix type only allows non-negative values. The idea is to further analyse factorization results by applying the non-negative MF (NMF) algorithms [65]. The analysis part is related to the hidden/latent factors characterizing users and items, revealed by MF. In the non-negative case, these factors, which are also non-negative, are directly interpretable due to their additive nature. Although the factors from the first matrix type are still useful, there is no a systematic way to specify what each factor refers to. In terms of solution strategies, we evaluate a number of memory-based and model-based CF algorithms on two different matrix forms. Besides that, we suggest a way to combine the strengths of the tested methods when they are used on one of the matrices to deliver overall better performance.

23.3 Problem Statement

In this section we will define the problem we are dealing with, in which we have multiple customer classes. However, to understand the concept we rely on first explaining the single-cluster model which can then be generalized to a multiple-cluster one so we start with this one first.

23.3.1 Single-Cluster Model

Let $\mathbf{N} = \{1, 2, \ldots, N\}$ be the set of available services. Let $K < N$ be the desired bundle size. The objective of our bundle design problem is to maximize the expected profit for a typical bundle customer:

$$\max \sum_{i \in \mathbf{N}} r_i(\mathbf{x}) x_i, \quad (23.1)$$

with the QoS constraint that the overall bundle attractiveness must be at least c:

$$\sum_{i \in \mathbf{N}} a_i(\mathbf{x}) x_i \geq c. \quad (23.2)$$

Finally, to ensure that the bundle size is correct, we have:

$$\sum_{i \in \mathbf{N}} x_i = K. \quad (23.3)$$

In the above formulation, x_i is a binary decision variable: set to 1 if service i is included and 0 otherwise. $\mathbf{x} = \{x_1, \ldots, x_N\}$ is the vector of all decision variables x_i, $i \in \mathbf{N}$. The function $r_i(\mathbf{x})$ gives the expected profit earned by the operator when the bundle is $\{i | x_i = 1, i \in \mathbf{N}\}$. Following earlier definitions, this expected profit function can be computed as:

$$r_i(\mathbf{x}) = P(S_i = 0 | S_1 = x_1, \ldots, S_N = x_N)\, u_i, \quad (23.4)$$

where $P(S_i = 0 | S_1 = x_1, \ldots, S_N = x_N)$ is simply the probability that a typical bundle customer would choose NOT to utilize service i, and u_i is the fee the operator needs to pay service provider i if the customer uses the service.

The attractiveness of service i can be similarly defined as:

$$a_i(\mathbf{x}) = P(S_i = 1 | S_1 = x_1, \ldots, S_N = x_N)\, v_i, \quad (23.5)$$

where the first component of Eq. (23.5) is the probability that a customer would choose service i, and v_i is the associated *value* it would bring to the customer. Note that for both cases, $P(\cdot)$ needs to be computed from the historical data.

Both Eqs. (23.4) and (23.5) are nonlinear; to linearize these two sets of equations, we can enumerate all possible bundles of size K as set $\mathbf{Y}_K$, and define new decision variables y_j to take on the value of 1 if bundle $Y^j \in \mathbf{Y}_K$ is chosen, and 0 otherwise. Let YjiYij be 1 if service i is included in bundle j; Eqs. (23.4) and (23.5) can then be rewritten as:

$$r_i(Y^j) = \begin{cases} 0, & \text{if } Y_i^j = 0 \\ P(S_i = 0 | S_1 = Y_1^j, \ldots, S_N = Y_N^j)\, u_i, & \text{o/w}, \end{cases} \quad (23.6)$$

$$a_i(Y^j) = \begin{cases} 0, & \text{if } Y_i^j = 0 \\ P(S_i = 1 | S_1 = Y_1^j, \ldots, S_N = Y_N^j)\, v_i, & \text{o/w}. \end{cases} \quad (23.7)$$

With (23.6) and (23.7), the bundle design problem (23.1)–(23.3) can then be linearized as:

$$\max \sum_{Y^j \in \mathbf{Y}_K} y_j \sum_{i \in \mathbf{N}} r_i(Y^j). \quad (23.8)$$

s.t.

$$\sum_{Y^j \in \mathbf{Y}_K} y_j \sum_{i \in \mathbf{N}} a_i(Y^j) \geq c, \quad (23.9)$$

$$\sum_{Y^j \in \mathbf{Y}_K} y_j = 1, \quad (23.10)$$

The final constraint ensures that exactly one bundle is selected.

The above optimization model looks deceivingly simple, yet it is non-trivial to solve, as the set $\mathbf{Y}_K$ is combinatorial, and the number of decision variable y_j will grow exponentially in K.

23.3.2 Multi-Cluster Model

As stated earlier, one major contribution we make in this chapter is the incorporation of multiple customer classes. To accommodate this, we only need to modify Eqs. (23.6) and (23.7). Instead of having only a single set of joint probability distribution on service selection (P(·))(P(·)), we will now have multiple sets of probability distributions.

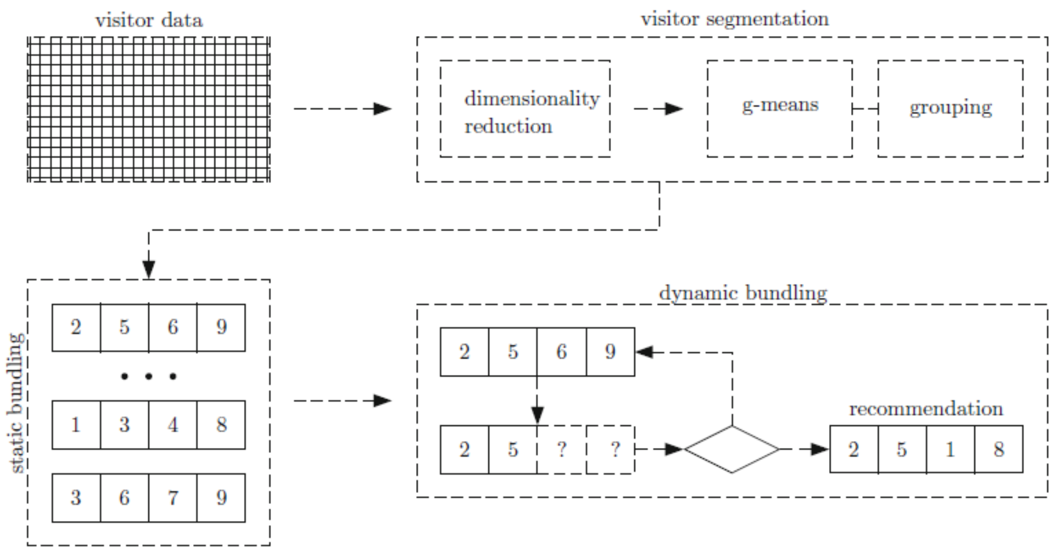
23.4 A Data-Driven Approach

The primary contribution of our chapter is the design and implementation of a data-driven approach for an operator managing a collection of services. More specifically, recommending bundles of services using consumer redemption data should maximize expected profits, while obeying QoS requirements (characterized by a lower bound on attractiveness or customer valuation). The mathematical formulation of the proposed method is already presented in Sect. 23.3.

In this section, we introduce and compare approaches for extracting multiple user classes from a real-world dataset. As the size of the mathematical formulation grows exponentially, we use an efficient and effective greedy heuristic for solving the bundle design problem. To provide a concrete context to the methodological discussion, we develop our approaches based on a real-world dataset (details are presented in Sect. 23.5) collected from a leisure park operator who leases out its land parcels to multiple attraction operators (i.e. service providers).

The proposed approach involves two important stages: (1) customer segmentation: based on historical data, infer multiple classes of customers who have inherently different preferences over provided services, and (2) segment-specific bundle recommendation: based on segment-specific parameters (in the form of probability function $P(.)$), solve individual bundle design problem using a greedy heuristic (to be explained). For the second stage, we propose two variants. The first variant recommends *static* bundles, referring that customers have to commit to the content of the purchased bundle. The second variant allows bundles to be *dynamic*, which means that a customer can request part of the bundle to be replaced by new recommendation. The greedy heuristic is based on an earlier work [33] on similar problem. For the dynamic variant, a memory-based CF method, i.e. nearest-neighbour and a model-based CF algorithm (CofiRank) are applied to two matrices consisting binary visit information and time information. Figure 23.2 visualizes the complete workflow of the proposed strategy.

Fig. 23.2 Workflow of the proposed data-driven approach with three steps including visitor segmentation: to determine different types of customers, static bundling: to offer attractive and profitable attraction bundles and dynamic bundling: to suggest on-the-fly changes on the bundles



23.4.1 Visitor (Customer) Segmentation

Visitor segmentation is a critical aspect of a bundle design process. For instance, consider a tourism agent who would like to sell travel packages for visiting multiple cities. The first step is how to determine these packages. In principle, each package should be designed such that they can meet different customers' needs. There can be a group of customers who are interested in nature while another group of customers are interested in history. Designing a single package to satisfy these two groups of customers is likely to be an unsuccessful strategy considering the strict differentiation between customers' preferences. Instead, providing a travel package with visits relating to nature and a travel package involving visits of historical sites is expected to more satisfactory for these customers.

In order to detect such variations on customers, some prior information regarding consumers is required. A questionnaire performed on a large group of people shows that customers' explicit preference information, historical order data and online browsing history are some examples that can be used to specify customer groups or can be used to segment customers. In this study, we use visit data of earlier visitors in the form of visitor × attraction matrices such that each row refers to a visitor and each column represents an attraction. This data indicates which attractions are visited by each visitor with day/time information. Although directly using such data would be useful, it might be hard to process depending on its size. Besides that, it is likely that data of a large number of visit information has some noise.

Singular Value Decomposition (SVD) [11] is applied to address both data size and noise issues. SVD is a widely used MF [26] technique particularly in the CF field both to approximate a given matrix to its smaller-rank version and to provide representative latent (hidden) factors. Matrix rank here refers to the column rank. SVD is capable of decreasing the number of columns of a corresponding matrix allowing to perform various matrix manipulations easier due to having relatively smaller-sized data. Thus, SVD is also stated as a dimensionality reduction approach. The other capability of SVD, being able to provide latent factors, is helpful to eliminate noise in the data. In particular, visitors can be characterized more efficiently by using these latent factors rather than directly using their attraction visit information.

As shown in Eq. (23.11), applying SVD to a matrix MM results in three matrices, i.e. U, V and Σ. U and V matrices are defined as left and right singular vectors while the diagonal Σ matrix provides sorted singular values. These U and V matrices represent rows and columns of $\mathcal{M}$, respectively. The Σ matrix informs about the importance of resulting latent factors that decreases for each subsequent dimension, e.g. the first latent factor is more significant than the second one.

$$\mathcal{M} = U \Sigma V^T \quad (23.11)$$

SVD is practical to achieve an efficient similarity analysis between the elements of a matrix. However, the primary reason using it here is to take advantage of dimensionality reduction for the sake of better and faster clustering for segmentation. After applying SVD to a customer's visit data, the produced U matrix is used as the matrix representing customers. The clustering on U is performed by the Gaussian-means (G-means) algorithm [23]. It is a clustering algorithm using k-means where k is automatically determined. G-means incrementally applies k-means for larger k values if a resulting cluster doesn't belong to Gaussian distribution. The process starts with a single cluster, $k = 1$ or $k > 1$ if there is prior knowledge, and k is iteratively increased until all the clusters are from Gaussian distribution. The distribution check is operated by applying the Anderson-Darling statistic [4]. If this test determines non-Gaussian clusters , the corresponding clusters are divided into 2 using principal component analysis and k-means is re-applied. The overall customer segmentation process is explained in Algorithm 1.

Algorithm 1: Two-step clustering strategy for customer segmentation, where $\mathcal{M}_{v,a}$ is a customer-attraction matrix

---

Data extraction: $\phi \rightarrow \mathcal{M}_{v,a}$;
Dimensionality reduction: SVD$(\mathcal{M}_{v,a}, r) \rightarrow U_{v,r}\, \Sigma_{r,r}\, V^T_{r,a}$ where $r <= min(v, a)$;
Customer-based clustering: $C = $ G-Means$(U_{v,r})$;
Calculate transition frequency matrices: $T_{c_i,aa}$ for each c_i from $C = \{c_1, \ldots, c_n\}$;
Measure cluster similarities based on these T matrices: $sim(T_{c_i,aa}, T_{c_j,aa})$;
Combine the clusters with a certain similarity level: $sim(T_{c_i,aa}, T_{c_j,aa}) \geq 0.5$;

The same MRF approach studied in [33] is pursued, now for each customer segment. Thus, it is expected that each segment differs in terms of their visit transition frequencies. For this purpose, the clusters determined by G-means are processed to generate transition matrices that are converted to vectors afterwards, where the frequency values are normalized as percentages. These vectors are compared to a similarity metric for detecting similar clusters. The similarity check is realized either using the Cosine similarity (Eq. 23.13) or Pearson Correlation Coefficient (Eq. 23.13). The Cosine similarity compares two vectors based on the angle between them while the Pearson Correlation coefficient considers average values in addition to Cosine. For the first metric, the similarity values change between 0 and 1 where 1 refers to the most similar cases. For the second metric, the similarity level varies between − 1 and 1 where 1 refers to the strong positive correlation and − 1 indicates strong negative correlation. In the equations, v_i refers to the customer transition vector, $a \in A$ is an attraction and rvi,arvi,a is the transition frequency value for a customer-attraction pair.

$$\begin{aligned} Cos.Sim(v_1, v_2) &= \frac{\vec{v_1}.\vec{v_2}}{\|v_1\| \times \|v_2\|} \\ &= \frac{\sum_{a \in A} r_{v_1,a} r_{v_2,a}}{\sqrt{\sum_{a \in A} r^2_{v_1,a}}\sqrt{\sum_{a \in A} r^2_{v_2,a}}} \end{aligned} \quad (23.12)$$

$$Cos.Sim(v_1, v_2) = \frac{\vec{v_1}.\vec{v_2}}{\|v_1\| \times \|v_2\|} = \frac{\sum_{a\in A} r_{v_1,a} r_{v_2,a}}{\sqrt{\sum_{a\in A} r_{v_1,a}^2}\sqrt{\sum_{a\in A} r_{v_2,a}^2}} \quad (23.13)$$

23.4.2 Static Recommendation

After all the customers are segmented, a static bundle is recommended for each visitor segment. The motivation studying static recommendation is to deliver generally acceptable bundles. When a new customer arrives, s/he should be able to find a reasonably satisfactory static bundle to start visiting the corresponding leisure park. The per segment static bundles with varying sizes are generated by applying a simple yet effective greedy construction heuristic [33]. The greedy heuristic iteratively adds an attraction with at least c/K attractiveness level and the lowest cost to generate a bundle of a size K, i.e. involving exactly K attractions. The greedy heuristic initially adds a single attraction via Eq. (23.14) respecting Eq. (23.15). The remaining $K-1$ attractions are selected using Eq. (23.16) while satisfying Eq. (23.17). It was shown that this heuristic is capable of delivering near-optimal solutions. The bundle tickets at this point are recommended as either fixed bundles or flexible bundles. The fixed ones belong to the customer segments which are too large considering all the segments. These segments are detected by using the interquartile range (IQR). IQR is calculated as $IQR = Q3 - Q1$, the difference between the third, $Q3$, and first quartiles, $Q1$. The segments larger than $Q3 + (1.5 \times IQR)$ are considered upper outliers.

$$Y = \arg\max_{Y_i \in X} \sum_{y_i \in Val(Y_i)} R_{y_j} P(y_i | X \backslash Y_i = 0) \quad (23.14)$$

$$\frac{\Gamma_{min}}{K} \leq P(Y_i = 1 | X \backslash Y_i = 0) \quad (23.15)$$

$$\arg\max_{Y_i \in X \backslash Y} \sum_{Y_j \in \hat{Y}_i}^{y_j \in Val(Y_j)} R_{y_j} P(y_j | X \backslash \hat{Y}_i = 0) \quad (23.16)$$

$$\frac{(|Y|+1)c}{K} \leq \sum_{Y_j \in \hat{Y}_i} P(Y_j = 1 | X \backslash \hat{Y}_i = 0) \quad (23.17)$$

23.4.3 Dynamic Recommendation

The idea of dynamic recommendation is to make online suggestions by offering changes on an existing ticket. Figure 23.2 provides a simple example. In the given scenario, a visitor who bought a bundle ticket (2, 5, 6, 9) decides to make changes on the ticket. This visitor either already visited attractions 2 and 5 or planning to visit these two attractions. Dynamic bundling aims at suggesting other attractions in exchange of attractions 6 and 9 which are not preferred to visit by the visitor. In the given example, (6,9) is swapped with (1,8). It should be noted that in this operation there is no obligation regarding 1-to-1 exchange meaning that one or more than two attractions can also be offered to change with two attractions.

As shown in the given example, dynamic recommendation is the next step of the static recommendation on the bundles. However, it could be argued that the dynamic recommendation might not be a critical component since the bundles are already personalized with the prior operation of customer segmentation. Although the argument sounds right, it is not completely correct since the primary idea of dynamic recommendation is to increase the level of personalization on the bundles. The level of personalization could be increased due to the generalization effect of the customer segmentation. From the customer segmentation perspective, the optimal segmentation would be considering each customer as a separate segment while the worst segmentation would be considering all the customers in a single segment. The customer segmentation approach can provide a balance between the optimal segmentation and the worst segmentation. The optimal segmentation could have been, of course, used to resolve this issue yet it can obviously make the system impractical. For instance, if we consider a leisure park of just 10 attractions, the possible number of bundles is 1023 ($2^N - 1$ where $N = 10$) that is the optimal case which can easily confuse the customers. The customer segmentation handles this confusion at a large extent yet it is still a generalization due to the conflict between the personalization and practicality. The dynamic recommendation can give the opportunity of personalizing the bundles further.

Dynamic recommendation is useful not only for a high level personalization but also for addressing the issues occurring in real-time. To exemplify, a customer walks towards to an attraction but encounters with a long queue and decides that they do not want to wait, or a customer needs to leave the leisure park earlier than her/his planned schedule so to find an attraction which requires shorter time or a customer sees an attraction from outside and dislikes it. For such cases, a customer can easily change her/his decision on-the-fly with dynamic recommendation.

For addressing dynamic bundling, the idea of collaborative filtering (CF) is incorporated. CF is popular in making predictions or recommendations of items that are expected to be interesting for a particular customer. When partial customers' preference information on items is available, a CF method can efficiently predict a customer's preferences on the items s/he haven't seen or tried yet. For this purpose, a number of memory-based and model-based CF methods are utilized. The *k*-nearest neighbour (kNN) based prediction method with varying similarity metrics is employed as a memory-based CF algorithm. A MF algorithm, i.e. CofiRank, is applied as a model-based approach. The customer-attraction matrix either with binary visit information or time period data is used to test dynamic recommendation. The results from the first matrix provide dynamic ticket changes while the second matrix is useful to also suggest tickets with order-based information.

For *k*NN, cosine similarity is used to determine the most similar entries. The similarity measurement can be done either by comparing customer entries/rows or attraction entries/columns. Due to the data size we are dealing with, similarities are measured based on matrix columns. After calculating all the similarities, a prediction matrix is created using the weighted prediction measure [39] as shown in Eq. (23.18).

$$pred(v, a_i) = \overline{r_{a_i}} + \frac{\sum_{a_j \in N} Cos.sim(a_i, a_j) \times (r_{i,a} - \overline{r_{a_j}})}{\sum_{a_j \in N} Cos.sim(a_i, a_j)} \qquad (23.18)$$

CofiRank [57] is inspired from MMMF [48]. It primarily aims at optimizing a rank-based loss function, i.e. Normalized Discounted Cumulative Gain (NDCG), Eq. (23.19).

NDCG@k is a metric that shows how well the top *k* items are predicted, besides that predicting an earlier item is more valuable for predicting an item that comes later, in the form of ranking. In other words, NDCG imitates the motivation behind search engines about finding the most relevant sites about a subject, first. The given equation provides a value between 0 and 1 where 0 refers to the case where all the predictions are wrong while 1 shows the case of a perfectly correct prediction. DCG@k is the actual measure indicating the quality of the prediction. It is divided by IDCG@k which is the perfect prediction case to normalize. DCG@k is calculated as shown in Eq.

(23.20). $r_{i,j}$ is the score or quality of item j where i indicates the prediction order. $1/log(i+2)$ is the discount factor. If the initial items are correctly predicted, their contributions to the DCG will be higher. Thus, the objective is to maximize DCG@k so NDCG@k. Besides NDCG, the squared regression loss (Eq. 23.21) and ordinal loss (Eq. 23.22) functions are provided. The regression loss is the generally used loss function considering the differences between the predicted values, p, and the actual values, r. The last function focuses more on predicting the right order of items regarding their scores or qualities. CofiRank targets to minimize one of these loss functions (L(UV,M)L(UV,M)) together with a trace norm based regularization element, as illustrated in Eq. (23.23). The optimization process accommodates a method manipulating U and V in an alternating manner for approximating to a given sparse or incomplete matrix. Due to the superior performance of the regression loss, only its results are reported.

$$NDCG@k = \frac{DCG@k}{IDCG@k} \qquad (23.19)$$

$$DCG@k = \sum_{i=1}^{k} \frac{2^{r_{i,j}} - 1}{log(i+2)} \qquad (23.20)$$

$$l(p,r) = \frac{1}{2}(p-r)^2 \qquad (23.21)$$

Table 23.2 details all the experiments carried out to address dynamic bundling. Besides the experiments using the aforementioned CF algorithms for completing sparse binary and time matrices, a combined approach is additionally suggested. kNN-BT, CofiRank-BT and kNN-B+CofiRank-T fall into this category. The idea here is to apply a CF algorithm separately to binary and time matrices, then validating the time matrix by the predictions from binary matrix. Each element of a time matrix changes between 0 to a relatively large value representing the daily end visit time. A 0 means unvisited or “not-be-included” and the rest of the values show that corresponding attractions are included. Considering that space of possible values to be assigned, it is very likely to set values other than zero to an attraction which is not actually visited. In the case of binary matrix, a possible value is either 0 or 1 so the chance of false prediction is very limited in comparison to the case where time matrix is used. In order to alleviate this issue, the unvisited attractions detected from the binary matrix are used to remove the non-zero elements for those attractions in the predicted time matrix.

Table 23.2 Tested collaborative filtering methods

Method	Details	Target matrix	Output
kNN-B	Column-based NN with Cosine similarity	Binary matrix	Dynamic bundling
kNN-T	Column-based NN with Cosine similarity	Time matrix	Dynamic bundling with ordering
kNN-BT	Column-based NN with Cosine similarity	Binary matrix + time matrix	Dynamic bundling with ordering
CofiRank-B	Regression loss	Binary matrix	Dynamic bundling
CofiRank-T	Regression loss	Time matrix	Dynamic bundling with ordering
CofiRank-BT	Regression loss	Binary matrix + time matrix	Dynamic bundling with ordering
kNN-B+CofiRank-T	Combined	Binary matrix + time matrix	Dynamic bundling with ordering

$$l(p,r) = \frac{1}{M}\sum_{r_i<r_j} C(r_i,r_j)max(0, 1+p_i-p_j) \tag{23.22}$$

$$\min_{U,V} L(UV, \mathcal{M}) + \frac{\lambda}{2}(trVV^T + trUU^T) \tag{23.23}$$

For comparison purposes, Association Rule Mining (ARM) [63] (a.k.a. Frequent Itemset Mining[2]) is used. ARM is a popular approach particularly in market basket analysis. The idea is to determine the items which are expected to be bought by a customer when s/he already bought a set of items. In other words, ARM looks for rules such that $A \rightarrow B$ where A is the set of existing items and B is the items expected to be included or interesting (A∪B=∅A∪B=∅). ARM operates based on frequencies like MRFs but in a simpler way. Support and confidence are the two main measures to evaluate the quality of ARM rules. Support of a rule $A \rightarrow B$ is calculated as shown in Eq. (23.24) where $n_{A \cup B}$ shows the number of transactions when items A and B are bought together and N refers to the total number of transactions. Confidence is calculated using support as presented in Eq. (23.25). While support is a basic frequency calculation of a rule, confidence is a conditional probability, $P(B|A)$, indicating how strong a rule is. For dynamic bundling, attractions (to be) visited are the ones included in A while B are the ones to be offered as dynamic choices ignoring the attractions not to be visited. However, it should be noted that using ARM can be costly for large datasets. Suppose that there are m items. The total number of itemsets can be derived is $2^m - 1$. Exhaustive search requires $N \times (2^m - 1)$ checks to evaluate and detect all the rules.
Increasing m values makes this process computationally expensive. Apriori [3] was introduced to efficiently tackle this issue. Apriori is able to quickly derive all the rules where each rule has a support level which is equal or higher than a predetermined value. The speed up provided by Apriori comes from this support level bound. Whenever it detects a rule that has a lower support than this bound, there is no need to check other rules containing the items from this particular rule. Thus, many infeasible solutions (the ones with low support) can be eliminated without even checking them and delivers a faster way of rule detection. For dynamic bundling, preferences on to-

be visited attractions and not-to-be visited attractions are provided as prior information to ARM. ARM returns the item set or bundle that includes these prior preferences with the highest confidence.

$$support(A \to B) = \frac{n_{A \cup B}}{N} \tag{23.24}$$

$$confidence(A \to B) = \frac{support(A \cup B)}{support(A)} \tag{23.25}$$

As the final comparison algorithm, the aforementioned Greedy construction heuristic is used. In this case, dynamic bundling is thought as static bundling with prior information. This prior information consists of the attractions to be initially included and the attractions to be ignored from visitors' dynamic bundling requests.

23.5 Computational Analysis

We work with a large leisure park operator in Asia which manages 17 attraction providers. The attendance (or redemption) records of 22,287 visitors under an all-you-can-visit model (i.e. visitors would pay for a bundle that consists of ALL 17 attractions) were collected. Our focus in this section is to show how we can improve the bottom line of the operator through better bundle design.

Each attendance record is a tuple containing 〈*Timestamp*, *CardID*, *AttrID*〉, where *AttrID* is the attraction ID, *CardID* is the card id representing each visitor and *Timestamp* shows when a particular attraction is visited. Sample attendance records are illustrated in Table 23.3.

Table 23.3 Sample attendance records of two visitors

Timestamp	CardID	AttrID
9/1/2012 09:33:11	308702168	4
9/1/2012 09:35:17	308599949	11
9/1/2012 10:12:34	308702168	6
9/1/2012 10:25:19	308599949	15

The data is then used to generate two matrices as exemplified in Table 23.4. The first is a binary matrix indicating whether an attraction is visited by each visitor. The second matrix (called the time matrix) keeps track of which attraction is visited at what time by which visitor. For the latter, time is sliced into 15-min intervals, where each time interval is denoted by an integer. In the given example matrices, visitor v_1 is shown as visited attractions a_2, a_3 and a_4 during time periods 1, 4 and 6, respectively.

For the dynamic bundling experiments, both binary and time matrices are used. In particular, one binary matrix and one time matrix are extracted from each visitor segment that is determined earlier. Initially, these matrices are processed to generate test instances in the form of *k*-fold cross validation. The cross-validation is done for the visitors/matrix rows. The visitors are first divided into *k* partitions. For instance, using the matrices from Table 23.4, twofold cross validation can be realized by dividing the visitors into two groups as (v_1, v_2, v_3) and (v_4, v_5, v_6).

Table 23.4 Redemption data representation example with six visitors and six attractions in two matrix forms: 0–1 matrix and time matrix from left to right (*v*: visitor, *a*: attraction)

	a_1	a_2	a_3	a_4	a_5	a_6
v_1	0	1	1	1	0	0
v_2	1	0	1	0	0	1
v_3	1	0	1	0	0	1
v_4	0	0	0	1	1	0
v_5	0	1	1	0	1	1
v_6	1	1	1	0	0	1

	a_1	a_2	a_3	a_4	a_5	a_6
v_1	0	1	4	6	0	0
v_2	6	0	3	0	0	4
v_3	11	0	7	0	0	5
v_4	0	0	0	4	9	0
v_5	0	10	12	0	18	20
v_6	3	9	5	0	0	13

For each partition, x entries are provided while the rest are removed in each row. The resulting test matrix has the full matrix entries for $k-1$ partitions and only x entries for the visitors in the other partition. In total, k different test matrices are created by considering each partition with partial data. Again if we look at Table 23.4 with $k=2$ and $x=3$, we keep the visit information of the first visitor group while keeping only $x=3$ entries of each visitor for the second group.
Table 23.5 shows the resulting matrices after performing the explained process. The same process is done for the second visitor group. As a result, two versions of binary and time matrices are generated for twofold. The goal here is to apply a CF method to complete these two matrices. In the matrices to be completed, available x values for each visitor refer to their partial preferences of the attractions visited and attractions not being visited. The missing values, denoted as "–", are the ones to be predicted to determine visitors' preferences on the related attractions. The attractions with the prediction of being visited are offered to swap with the attractions excluded as dynamic bundling. In a dynamic bundling scenario, the values different than zero refer to the attractions which are already visited and entries with zeros refer to the attractions requested to be swapped.

Table 23.5 Redemption data representation example for dynamic bundling with six visitors and six attractions in two matrix forms: 0–1 matrix and time matrix from left to right (*v*: visitor, *a*: attraction)

	a_1	a_2	a_3	a_4	a_5	a_6
v_1	0	1	1	1	0	0
v_2	1	0	1	0	0	1
v_3	1	0	1	0	0	1
v_4	–	0	–	1	1	–
v_5	0	–	1	–	1	–
v_6	1	1	–	–	0	–

	a_1	a_2	a_3	a_4	a_5	a_6
v_1	0	1	4	6	0	0
v_2	6	0	3	0	0	4
v_3	11	0	7	0	0	5
v_4	–	0	–	4	9	–
v_5	0	–	12	–	18	–
v_6	3	9	–	–	0	–

In the experiments, k is set to 5 (fivefold cross validation) which is small enough to test on the small-sized visitor segments and x is set as 3 to evaluate the dynamic bundling performance when the data is sparse enough.

For the clarity of the computational analysis, the notations used are listed in Table 23.6.

Table 23.6 Notations used onwards

Notation	Definition
$\mathbf{N} = \{a_1, a_2, \ldots, a_N\}$	Set of available services or attractions to be visited
$\mathbf{B} = \{b_1, b_2, \ldots, b_{2^N-1}\}$	Set of all possible bundles, involving services from **N**, i.e. powerset of **N** excluding $\emptyset$: $\mathcal{P}(\mathbf{N})\backslash\emptyset$
$\mathbf{B}' \subseteq \mathbf{B}$	Set of unique bundles consumed before
$M = \|B'\| \leq 2^N - 1$	Number of unique bundles consumed before
$K = \|b_i\| \leq N$	Size of a bundle or number of services included in a bundle
V_s	Set of segments
$V_{s,i}$	Set of visitors in segment i
$f(b_i)$	Bundle frequency, i.e. number of times a set of services, composing b_i, appears together in $\mathbf{B}'$
$f(b_i)$ where $\|b_i\| = 1$	Single service consumption frequency, i.e. number of times a service is consumed before
$f_r(b_i) = f(b_i)/M$	Bundle frequency ratio
$f_r(b_i) = f(b_i)/N$ where $\|b_i\| = 1$	Single service consumption frequency ratio
$f_r(b_i) \times 100$ where $\|b_i\| = 1$	Single service consumption frequency in percentage, i.e. % visit frequency
c_{a_j}	Expected cost of service a_j, to the park operator
q_{a_j}	Attractiveness of service a_j
$\mathcal{C}(b_i) = \sum_j^{\|b_i\|} c_{a_j}$ where $a_j \in b_i$	Expected cost of bundle b_i, to the park operator
$\mathcal{Q}(b_i) = \sum_j^{\|b_i\|} q_{a_j}$ where $a_j \in b_i$	Attractiveness of bundle b_i
$\mathcal{P}_l(b_i) = \{b_{i1}, b_{i2}, \ldots, b_{iz}\}\backslash\{b_i\}$ where $f(b_{ij}) \geq l$	Subsets of services from b_i excluding b_i, with $f_r(b_i) \geq l$
$\mathcal{D}_l(b_i) = f_r(b_i)/(\sum_j^{\|\mathcal{P}_l(b_i)\|} b_{ij}/\|\mathcal{P}_l(b_i)\|)$ where $b_{ij} \in \mathcal{P}_l(b_i)$	Degree of complementarity

23.5.1 Complementarity Analysis

One of the most commonly studied aspects of bundles is the relation between bundled items. The level of relation between items can affect both the attractiveness and the price of a bundle. This relation can be evaluated from the perspectives of *substitutability* and *complementarity*. Substitutability refers to the products or services which have similar nature, e.g. two different roller coasters in the same leisure park. Complementarity is used for the goods with different characteristics such as a roller coaster and a 3D movie theatre, to be tried in complement to the overall experience. In principle, the bundles of complementing products are expected to be more attractive than the ones with the substitutable products, to the customers. However, it is not always clear to say whether two products are substitutable, complementary or totally independent so to talk about the bundles' attractiveness, as discussed in Sect. 23.2.1. For instance, a customer who likes roller coasters a lot would be more interested in a bundle of allowing to experience two roller coasters than the bundle of a roller coaster and a 3D movie theatre.

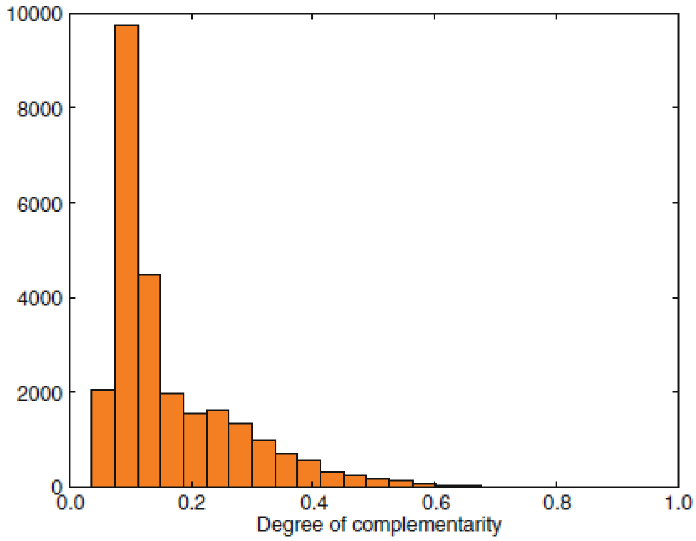
In [60], the degree of complementarity concept evaluating these relations is introduced from a pricing perspective. We consider the same concept using frequency ratios of service bundles ($f_r(b_i)$) derived from the given historical redemption data. A frequency ratio simply indicates the popularity of a bundle. For instance, a bundle of two products, i.e. $b = \{\text{p1}, \text{p2}\}$, appears together in the redemption data for 30% of the visitors, means that $f_r(b) = 0.3$. As given in Table 23.6, the degree of complementarity (DD) is calculated by using $f_r(b)$ of a given bundle divided by average $f_r(.)$ of all the subsets of b excluding itself. DD changes between 0 and 1 where 1 shows the case when a bundle occurs as much as the average of its subsets. Thus, the values close to 1 indicate that the bundles with a high degree of complementarity while 0 indicates that some of the items decrease the degree of complementarity. Going back the two product bundle examples, $b = \{p_1, p_2\}$, consider that product p_1 appears in the 80% of the visits, i.e. $f_r(\{p_1\}) = 0.8$ and product p_2 is chosen for 70% of the time, i.e. $f_r(\{p_2\}) = 0.7$. In this case. the degree of complementarity of the bundle b is D=fr(b)/((fr({p1})+fr({p2}))/2)=0.3/((0.8+0.7)/2)=0.4D=fr(b)/((fr({p1})+fr({p2}))/2)=0.3/((0.8+0.7)/2)=0.4. Although both products are relatively popular when considered independently, the resulting bundle is unable to provide a similar level popularity as only appearing in 40% of the redemption transactions. It should also be noted that the highest possible value of D(bi)D(bi) is the product subset with the lowest frequency ratio, $f_r(.)$, as shown in Eq. (23.26). Thus, including a highly unpopular product set in a bundle substantially degrades the degree of complementarity of a bundle additionally with very popular products.

$$max(\mathcal{D}_l(b_i)) \leq min(f_r(\{b_{ij}\})) \text{ for } \forall\, b_{ij} \in \mathcal{P}_l(b_i) \qquad (23.26)$$

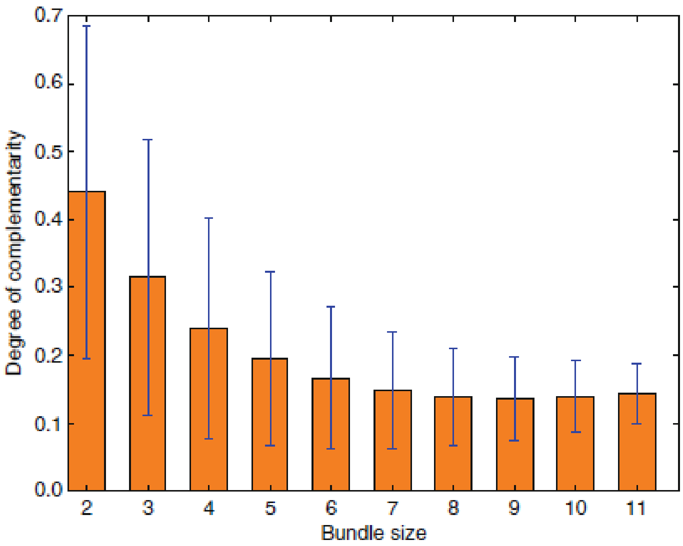
Figure 23.3 indicates D0.01D0.01 of each bundle with respect to its subsets using the target redemption data. Without considering bundle size, the figure shows that majority of the attraction sets or bundles have a degree of complementarity around 0.1. However, there are still bundles with high degree of complementarity, e.g. 0.93, even though they are very rare. When we look at the bundles of the same size, the smaller sized bundles show a higher degree of complementarity. The underlying reason is about the number of subsets of a bundle, i.e. adding more attractions to a bundle increases the number of attraction subsets. Thus, DD has tendency to become smaller yet more stable. Besides that, adding an attraction set with very low visit occurrence automatically decreases the visit frequency of the whole bundle, as just discussed.

In addition, having many attractions in a bundle causes a more complex complementarity structure, requiring evaluating the relations between all the subsets of a bundle. That’s why complementarity has been studied and analysed mostly for the bundles of two products only [5].

Fig. 23.3 Degree of complementarity of the bundles in the given historical redemption data (only attraction sets with at least 1% of occurrence ($f_r(b_i) \geq 0.01$) are considered). (**a**) All bundles. (**b**) Per bundle size



(a)



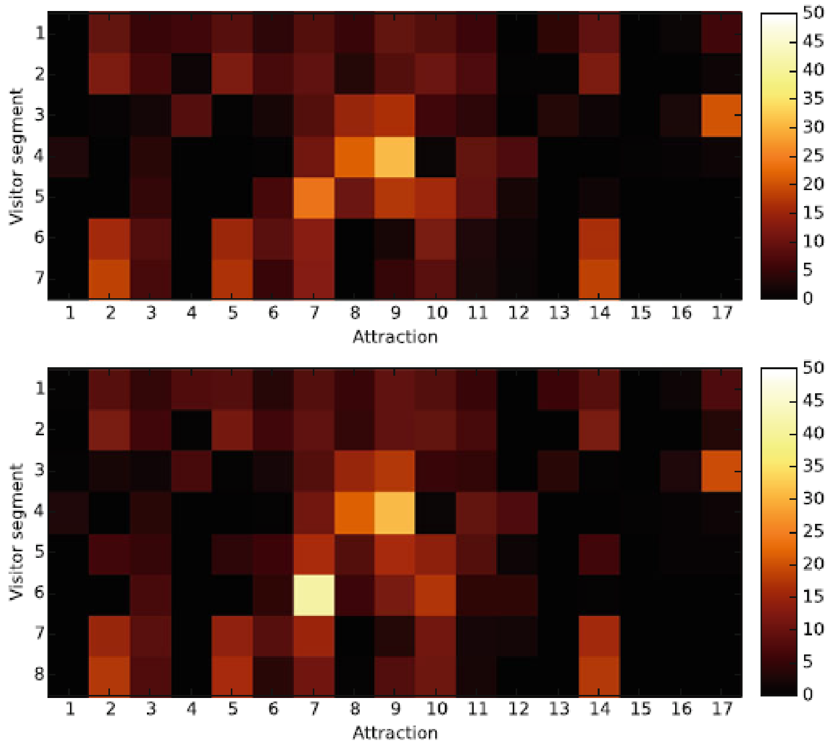
(b)

23.5.2 Visitor (Customer) Segmentation

The time matrix is extracted from the redemption data (Algorithm 1, line 1) for segmenting visitors. Each matrix row represents a visitor's attendance record. SVD is applied to the time matrix first for dimensionality reduction (Algorithm 1, line 2). One of the resulting matrices, i.e. $U_{v,r}$, is used in the next clustering step since it represents visitors. While v shows the total number of visitors, r can be set to a value between 1 and $min(v, a)$, where a refers to the number of attractions. $min(v, a)$ here is the mathematical dimension limit for SVD. In our case, r can be 17 (min(22287, 17)) at most. Since SVD provides a singular matrix with sorted values, the initial features are more critical to approximate than the subsequent ones. In particular, the top five singular values returned by SVD are 8256, 2794, 2234, 2148, 2091. Due to the large difference of the first singular value to the rest, single dimension already provides a very good approximation to the time matrix alone. The clusters derived by G-means also showed that for $r > 2$, there is no significant change on the clusters, thus r is set to 2. In the next step (Algorithm 1, line 3), G-means revealed 207 clusters derived from the U matrix with $r = 2$. The 207 clusters are further analysed to detect the similar clusters in terms of their normalized visit frequencies using two similarity metrics, namely Cosine and Pearson (Algorithm 1, line 4–5). After combining similar clusters, the number of clusters is decreased to 7 and 8 for Cosine and Pearson (Algorithm 1, line 6).

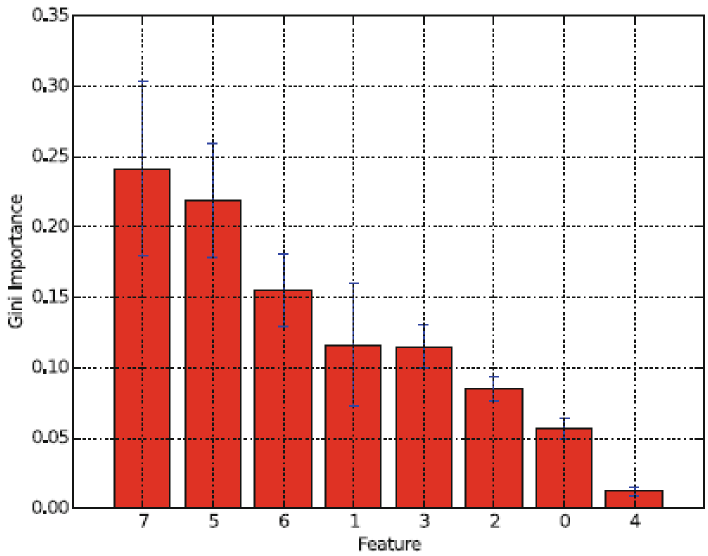
Figure 23.4 visualizes differences between clusters/visitor segments regarding percentage visit frequencies of each attraction in each segment. In other words, this figure gives information like 30% of the time attraction a_i is visited while attraction a_j is visited only 5% of the time. In the redemption data, the reported visit frequencies indicate that there is no clear similarity between visitor segments meaning that visitors are efficiently segmented based on this most basic visit frequencies. This is valid for both the Cosine and Pearson similarity metrics. When the visitor segments derived using these two metrics are compared, per attraction visit behaviours of the customers look similar. This shows that there is relatively high overlap between segments of both metrics. If we just consider the most frequently visited attractions in each segment, attractions 2, 7, 9, 14 and 17 are the most popular ones across different segments for both similarity metrics. If we look at the attractions which are either not visited at all or visited least frequently, attractions 1, 4, 8, 13, 15, 16, 17 and attractions 1, 2, 4, 5, 13, 15, 17 come out for Cosine and Pearson, respectively.

Fig. 23.4 Percentage visit frequencies of each attraction in each visitor segment for Cosine (top) and Pearson (bottom)

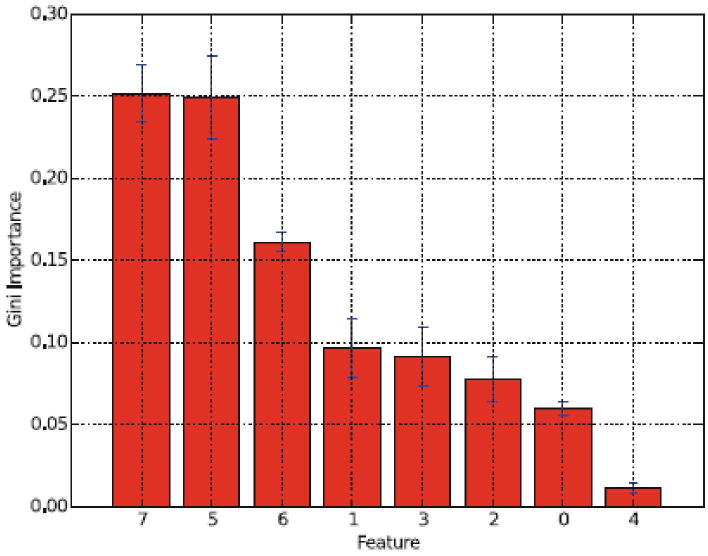


In order to further determine the elements affecting the formation of these clusters, eight basic visitor features are defined, as given in Table 23.7. The first feature represents the ticket type bought by visitors and the second feature shows the number of visits performed by each visitor. Unlike these two features, the remaining ones are all date/time-related. Random forests are applied to analyse the effects of these features on these clusters by generating a classification model between the features and clusters (as classes). Figure 23.5 details the Gini importance of each feature while building the corresponding classification models. The results indicate that the standard deviation of spent time on attractions and total spent time during a visit are the most critical features in determining the clusters obtained. Average spent time per attraction also contributed as another relatively useful feature. Following these major features, the number of visited attractions, the visit start time for the first attraction and weekly visit day respectively. Among the least important two features, the ticket type has very limited effect since the tickets provided do not really differentiate the behaviour of the visitors. The worst performing feature, in terms of separating the clusters, is that of whether a particular visit is performed on a weekday or weekend. Considering that the number of resulting clusters and this feature is binary, its effect on clusters is negligible as expected.

Fig. 23.5 Feature/variable importance determined by random forests using eight features based on clusters determined by the two-stage clustering (features are sorted w.r.t. their importance levels). (**a**) Cosine—seven segments. (**b**) Pearson—eight segments



(a)



(b)

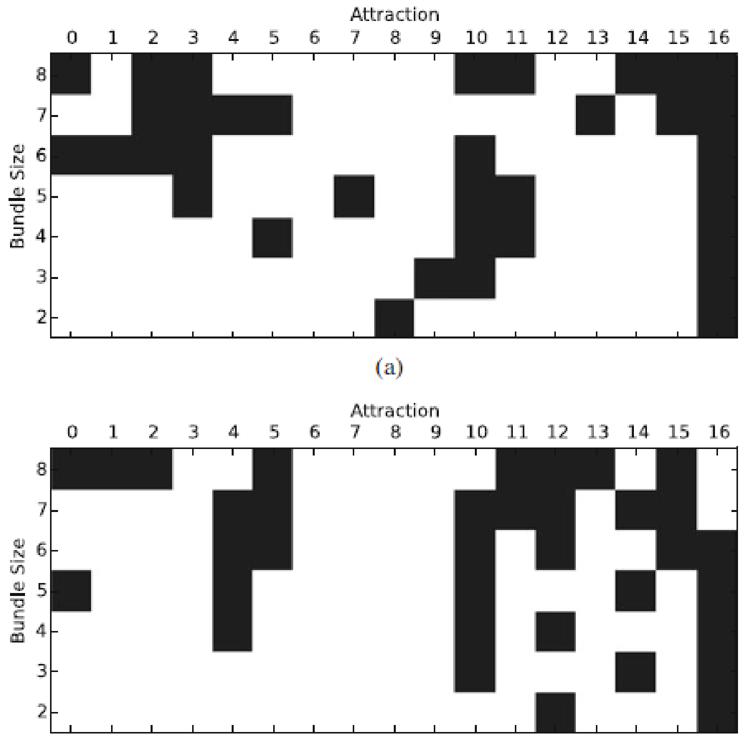
Table 23.7 Eight features representing visitors

Feature	Definition
f0	Ticket type
f1	Number of visited attractions
f2	Visit day
f3	First visit start time in hours
f4	Day type: weekday or weekend
f5	Total time spent except for the last visited attraction
f6	Average spent time except for the last visited attraction
f7	Standard deviation of the time spent, ignoring the last visited attraction

23.5.3 Static Recommendation

Figure 23.6 illustrates the bundle tickets recommended for different bundle sizes. When Cosine is used as the similarity metric to combine clusters determined by G-means, attractions 6 and 12 are the ones not included in any of the bundles. Attraction 16 is included in each bundle. For the Pearson Correlation, attractions 3, 6, 7, 8 and 9 are not considered in the suggested bundles. Although there is no attraction that is always available in the bundles in this case, attraction 10 and attraction 16 are the most commonly bundled attractions.

Fig. 23.6 Outlier bundles delivered by the Cosine and Pearson similarity metrics (black refers to fixed attractions, white shows that attractions are not recommended in bundles). (**a**) Cosine. (**b**) Pearson

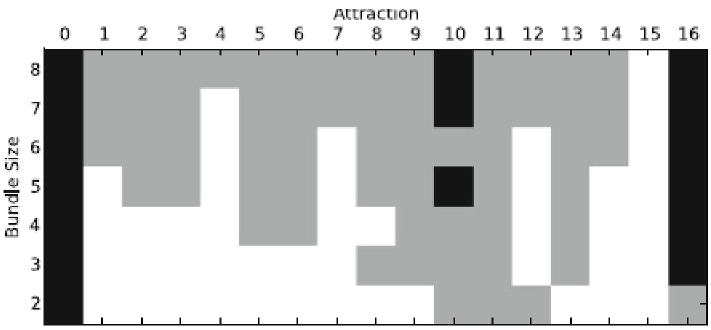


(a)

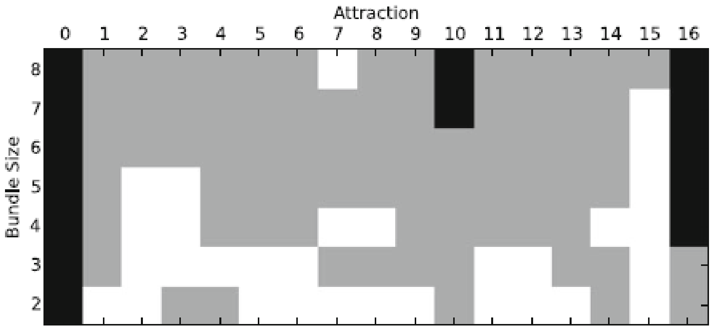
(b)

Considering that the segment sizes are relatively small, flexible bundle tickets are proposed for the remaining segments. Figure 23.7 details these bundles. For both similarity metrics, 0 is the common attraction for all the reported bundle sizes. Attraction 16 is also bundled in the majority of the bundles. Attraction 10 is the last fixed bundled attraction, but only for relatively large-sized bundles. When Cosine is used in segmentation, many attractions are excluded from the bundles, especially for the bundles with a few attractions. The Pearson similarity provides more flexible choices for different bundle sizes.

Fig. 23.7 Non-outlier flexible bundles delivered by the Cosine and Pearson similarity metrics (black refers to fixed attractions, grey shows that attractions that can be added to the bundles, white shows that attractions are not recommended in bundles). (**a**) Cosine. (**b**) Pearson



(a)



(b)

In order to perform an additional analysis on the quality of the visitor segments or to determine whether visitor segmentation is required for our data, we evaluated both the expected cost and attractiveness (QoS) of each static bundle generated for each segment. For evaluation, we generated a separate MRF for each visitor segment and one MRF using whole redemption data without segmentation. Using the resulting MRFs, we re-evaluated each static bundle.
Figures 23.8 and 23.9 illustrate the results for the static bundles generated after visitor segmentation with Cosine and Pearson, respectively. In both figures, one chart focuses on only the largest segments which have significantly more visitors than the remaining segments. For these charts, expected bundle costs and attractiveness are directly reported. For the smaller-sized visitor segments, the costs and attractiveness values are illustrated in the form of weighted average. Equations (23.27) and (23.28) show how average cost and average attractiveness are calculated. When the largest single segment in Cosine is considered, similar cost evaluation is seen for both MRF of the whole data and segment specific MRFs for the bundles sized until $K = 5$. For the larger sized bundles ($K > 5$), in particular the ones with 6 and 7 attractions, full redemption data based MRF expects significantly more costly bundles while segment specific MRFs tell that they are actually expected to incur less cost to the park operator. The bundles' attractiveness levels are smaller for the full data based MRF compared to the segment specific MRFs except the bundles with size 6 and 7 which are the costly ones. In the Pearson-based largest segment, the changes on cost and attractiveness are smoother. The expected bundle costs consistently increase in relation to the bundles' sizes for the MRF generated without segmentation. Until a bundle size of 7, segment specific MRF based expected costs and attractiveness levels are higher. For a bundle size of 8, the full data based MRF evaluates the corresponding bundle as significantly costly and very attractive for the visitors. A similar trend can be seen for the remaining segments of both Cosine and Pearson. In these cases, cost and attractiveness of segment based MRFs are very consistent.

$$\sum_{i}^{k} V_{s,i} \times \mathcal{C}(b_i)/|V_s| \qquad (23.27)$$

$$\sum_{i}^{k} V_{s,i} \times \mathcal{Q}(b_i)/|V_s| \qquad (23.28)$$

Fig. 23.8 The effect of Cosine-based visitor segmentation in terms of bundles' expected costs and attractiveness (QoS constraint set to 1.0; bar plots show cost, line plots indicate attractiveness). (**a**) Outlier/largest segment. (**b**) Remaining segments

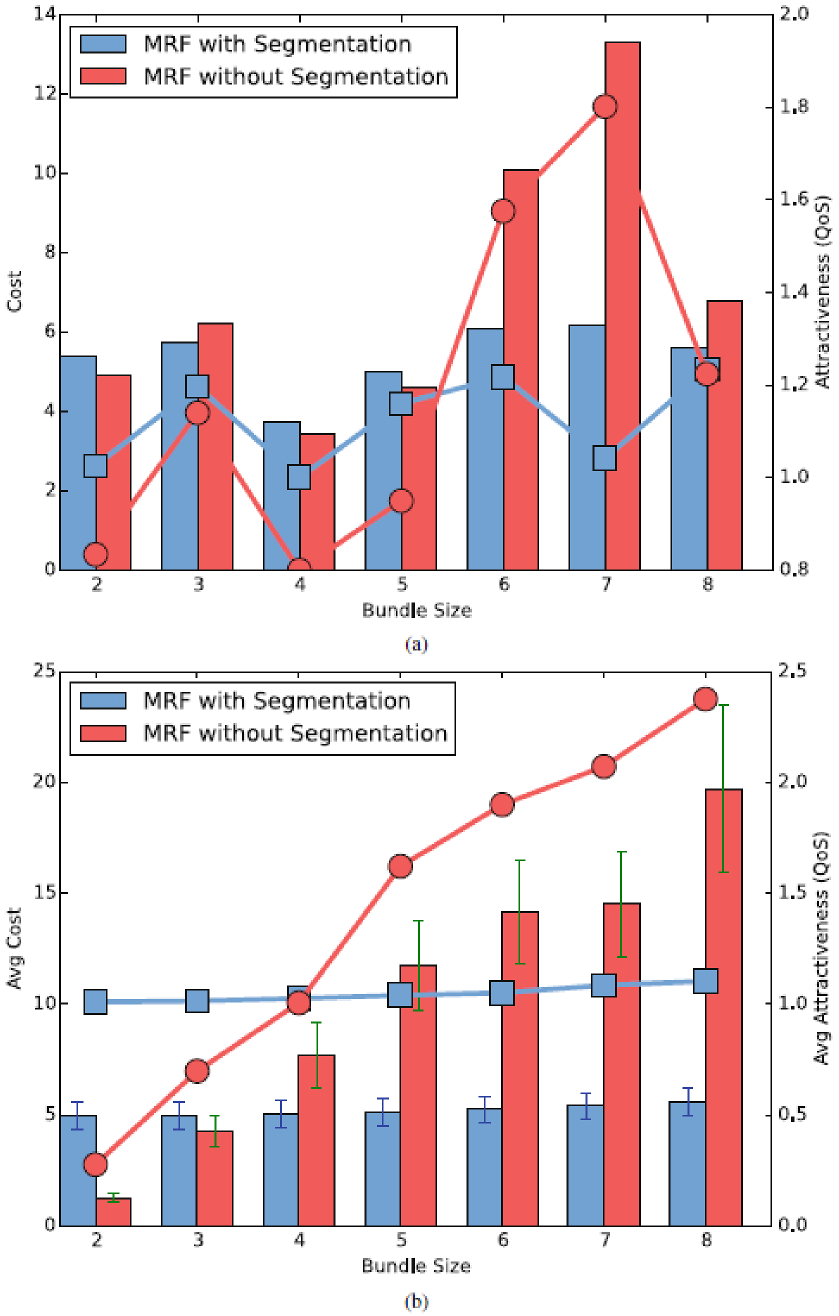
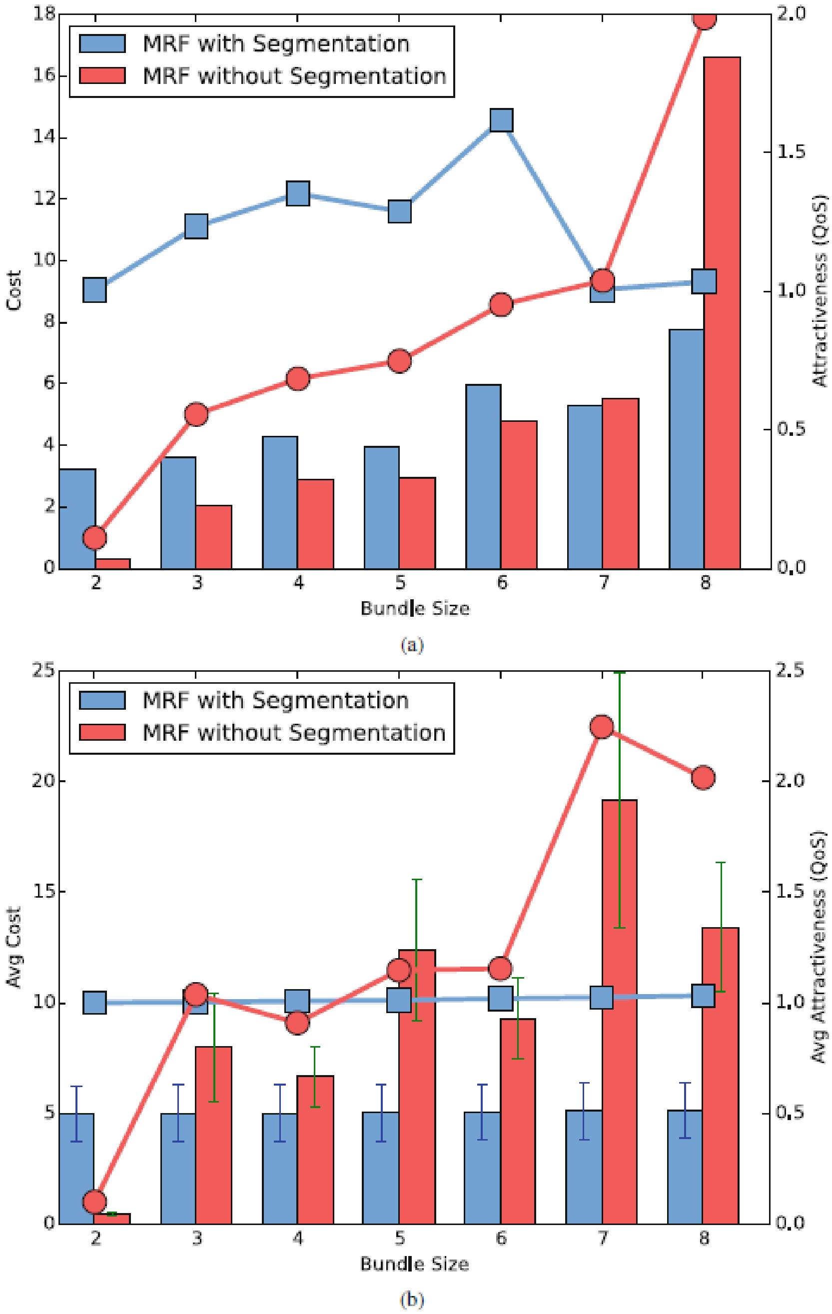


Fig. 23.9 The effect of Pearson-based visitor segmentation in terms of bundles' expected costs and attractiveness (QoS constraint set to 1.0); bar plots show cost, line plots indicate attractiveness). (**a**) Outlier/largest segment. (**b**) Remaining segments

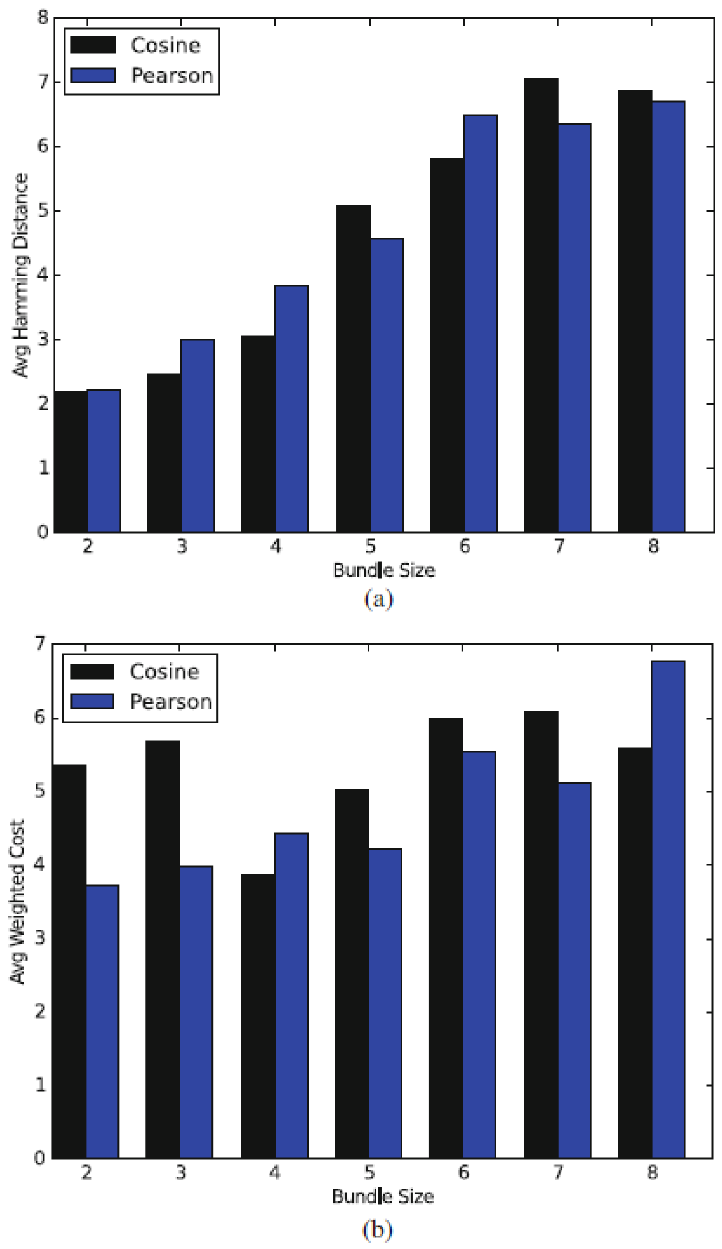


In brief, these figures indicate that generating a single MRF using whole redemption data can be misleading due to not taking visitors with distinct attraction preferences into account. The differences on the expected costs and attractiveness can be explained by two reasons. If expected cost and attractiveness are higher when full data based single MRF is used, it means that the corresponding bundles occur more frequently in the complete data while these bundles are rarely preferred in some of the visitor segments. For the opposite case, there can be a small group of visitors with similar visit behaviour. Since the number of these visitors is small, they are considered as not-interesting and the single full data based MRF is unable to appreciate their specific attractiveness to these small set of visitors. Thus, the expected costs and attractiveness are measured as lower by the single MRF.

Figure 23.10 shows how diverse the static bundles are and the average weighted cost incurs when either Cosine or Pearson is used as a similarity metric besides the bundle size. Both diversity and cost are measured across all the static bundles generated for all the visitors segments. Diversity is particularly useful to check how similar the bundles proposed for each visitor segment. High diversity indicates that different visitors have distinct preferences while low diversity means that visitor segmentation is unnecessary for bundling (indicating a homogeneous customer base). Diversity is measured using the mean hamming distance, H(B)H(B), where *B* refers all the bundles. As shown in Eq. (23.29), it is calculated as the average of the pairwise distances.

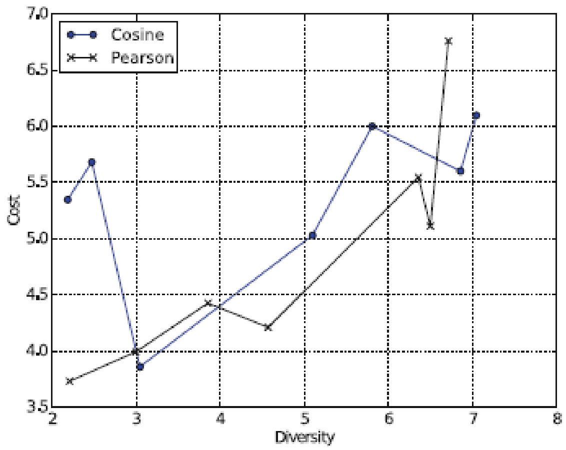
$$\mathcal{H}(B) = \frac{2}{N(N-1)} \sum_{i_1 \neq i_2}^{N} \sum_{j=1}^{l} |r_{i_1,j} - r_{i_2,j}| \qquad (23.29)$$

Fig. 23.10 Diversity and cost of the bundles with varying sizes generated by the greedy heuristic for the visitor segments. (**a**) Diversity. (**b**) Avg weighted cost



In terms of diversity, the best similarity metric changes for different bundle sizes. For instance, if the bundle size is set to 4, Pearson provides higher diversity but Cosine is better when bundle size is set to 7. Besides that, the diversity values in general and their increase in relation to bundle size indicate that visitor segmentation *is* in fact required and the proposed approach is able to detect the differentiate visitors segments. The cost results indicate that Cosine delivers more costly bundles for 5 out of 7 bundle size options. As detailed in Fig. 23.11, if the relation between Diversity and Cost are explicitly analysed, Pearson usually offers bundles with lower costs for similar diversity levels than Cosine.

Fig. 23.11 Diversity vs Cost relation for the Cosine and Pearson similarity metrics



23.5.4 Dynamic Recommendation

The performance of different CF recommendation approaches is analysed on the binary and time matrices. The first prediction task is to determine which attractions should be included in a bundle in the form of dynamic bundling. The other prediction task is about accurately detecting visiting order of the bundled attractions.

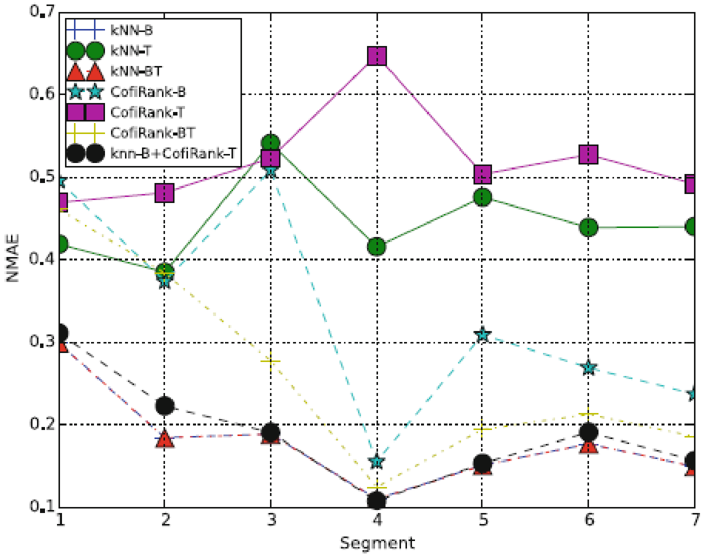
Figure 23.12 reports Normalized Mean Absolute Error (NMAE) (Eq. 23.30) for predicting the right dynamic bundle suggestions in terms of the attractions included in the bundles. MAE indicates the prediction error as the average of the absolute difference between the predicted and actual values. NMAE is the normalized version of MAE. For example, NMAE = 0.2 means that 20% of the predictions are wrong, so lower NMAEs are better. The results indicate that kNN is able to deliver more accurate bundle predictions than CofiRank when the Binary matrix is used even though kNN is faster and simpler.

$$MAE = \frac{\sum_{i,j} |p_{ij} - r_{ij}|}{n}$$

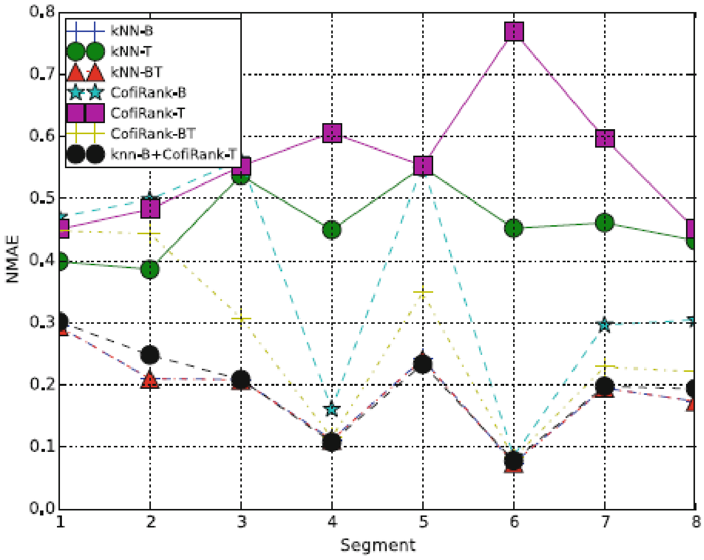
$$NMAE = \frac{MAE}{r_{max} - r_{min}} \qquad (23.30)$$

The kNN performance changes between ~10% and ~30% for different visitor segments both for Cosine and Pearson Correlation, which is very successful from a collaborative filtering perspective. The underlying reason behind the superior performance of kNN is the given incomplete matrix. Considering that we have a large amount of visitor redemption data, the matrix incompleteness level/sparsity is very low in comparison to the existing target collaborative filtering data. Thus, memory-based approaches like kNN are expected to perform well. When the incompleteness level is very low or there is not enough redemption data, model-based approaches like CofiRank are likely to perform better. If the time matrix is used to predict which attractions should be included in a bundle for each visitor, the results are significantly worse compared to the binary matrix case. As mentioned earlier, the reason is that the time matrix has two types of information including attractions available in each bundle and the visiting time of each attraction with higher range of values than just 2 as in a binary matrix. Applying a CF algorithm directly to a time matrix assigns time to many of the unvisited attractions. Thus, these attractions are also assumed to be visited in the predictions which result in high NMAEs.

Fig. 23.12 Normalized Mean Absolute Error (NMAE) for evaluating the quality of the dynamic bundle suggestions (the graphs belong Cosine and Pearson respectively). (**a**) Cosine. (**b**) Pearson



(a)

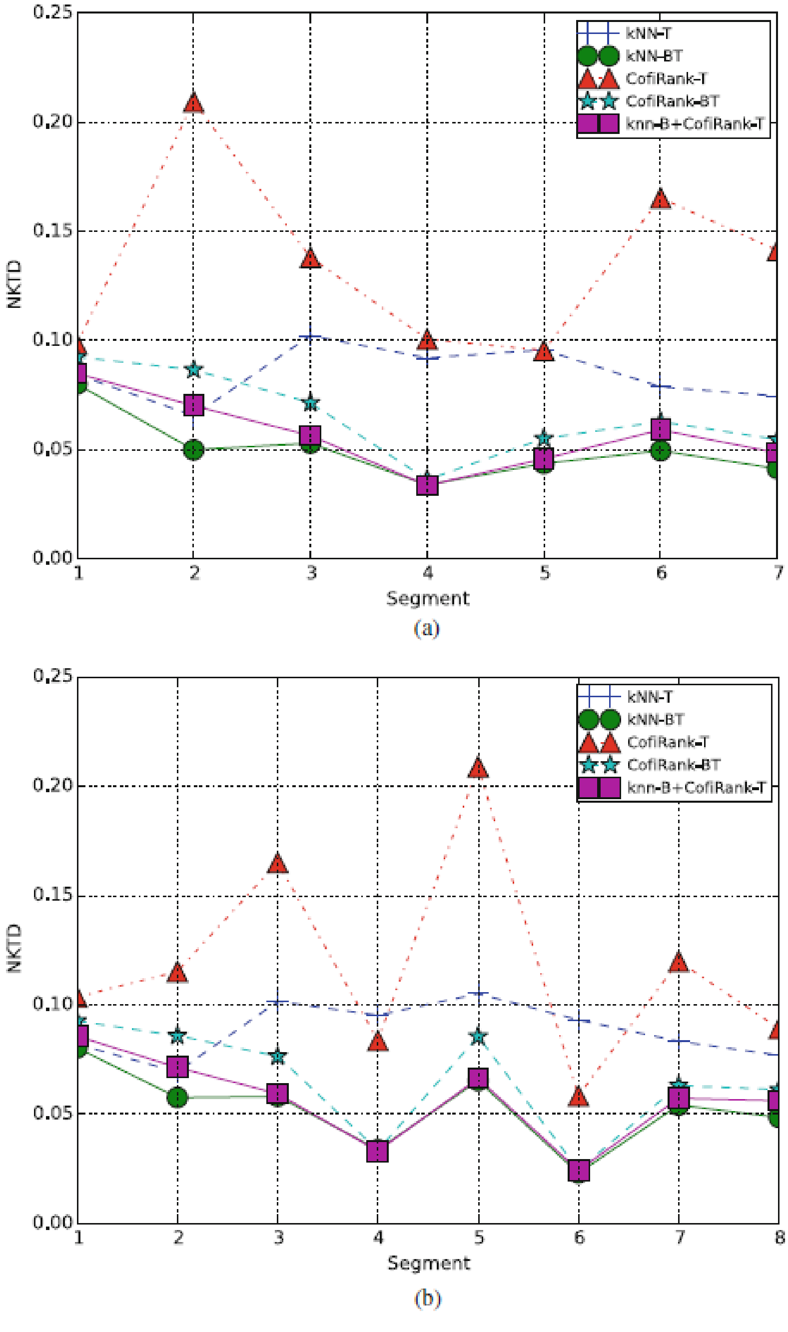


(b)

In order to resolve this issue of using the time matrix for prediction, the predictions on the binary matrix are utilized to correct the results on the time matrix. As a consequence, a combined approach, i.e. kNN-BT, which applies kNN to the binary matrix first to correct the predictions of kNN on the time matrix, delivered the best performance. The Wilcoxon rank sum test within a 99% confidence level indicated that kNN-BT is statistically better than all tested approaches except kNN-T and kNN-B+CofiRank-T.

Besides recommending dynamic bundles efficiently, predicting a good visiting order for bundles would help the visitors to enjoy the theme park more or to make better choices to decide on which attraction to visit next. Figure 23.13 evaluates the performance of the tested CF methods for the ordinal predictions in terms of the normalized Kendall's tau distance (NKTD) ∈ [0, 1] (Eq. 23.32).

Fig. 23.13 Normalized Kendall's tau distance (NKTD) for assessing the recommended visiting order of the dynamic bundle suggestions (the graphs belong Cosine and Pearson respectively). (**a**) Cosine. (**b**) Pearson



KTD basically counts the number of times when the pairwise visiting order of attractions is incorrect. For the ordinal predictions, kNN-BT comes as the statistically best method together with kNN-B+CofiRank-T according to the Wilcoxon rank sum test within a 99% confidence level. kNN-BT delivers results with NKTD varying between 0.04 and 0.07. This indicates that CF is successful in determining good visit orders or trip plans while suggesting dynamic bundle recommendations. CF is able to address a small-sized optimization problem without needing an optimization algorithm, solely as a recommender system.

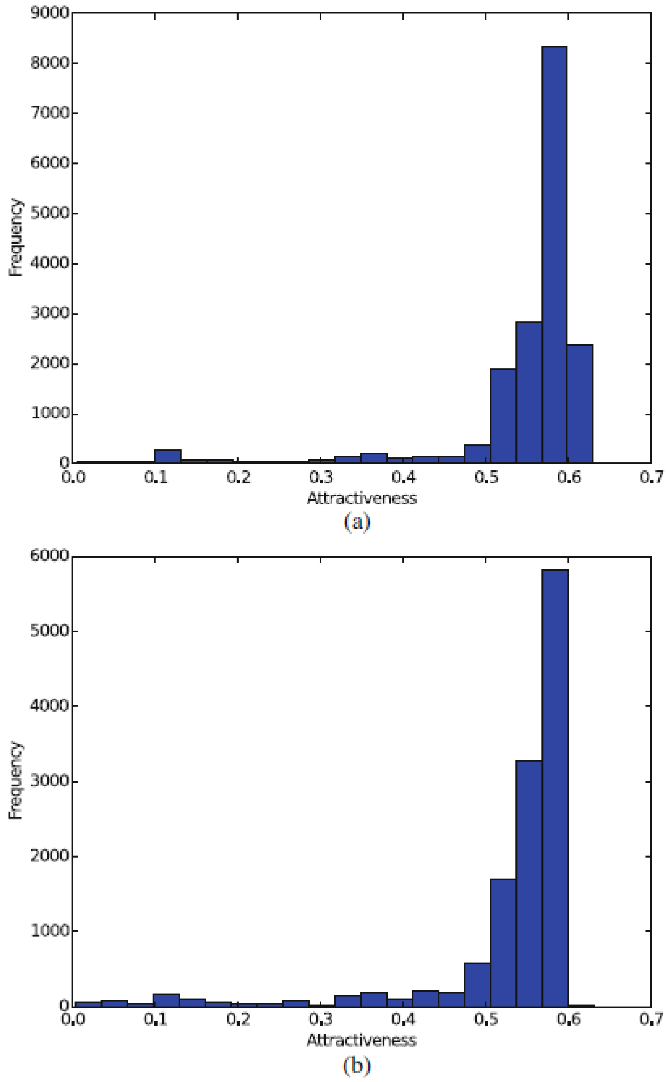
$$KTD = \sum_{i}^{v} \sum_{j}^{a} \sum_{k}^{a} z_{i,j,k} \text{ where} \tag{23.31}$$

$$z_{i,j,k} = \begin{cases} 1, & \mathcal{M}(i, j) < \mathcal{M}(i, k) \text{ and } \mathcal{P}(i, j) > \mathcal{P}(i, k) \\ 0, & \text{otw.} \end{cases}$$

$$NKTD = \frac{2KTD}{v(a^2 - a)} \tag{23.32}$$

Figure 23.14 shows the average attractiveness of each attraction included in dynamic bundles for the first and largest visitor segments of both Cosine and Pearson. Attractiveness is measured again using MRFs.

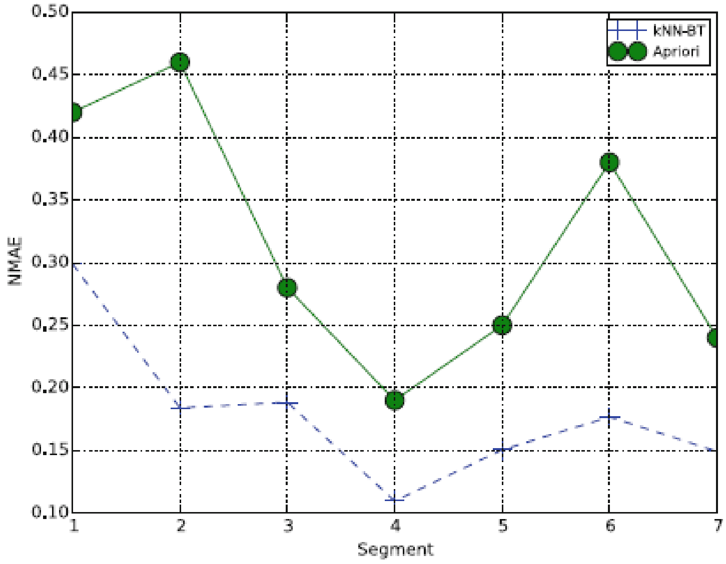
Fig. 23.14 Average attractiveness of the dynamic bundles offered w.r.t. MRF's attractiveness measure. (**a**) Cosine—Segment #1. (**b**) Pearson—Segment #1



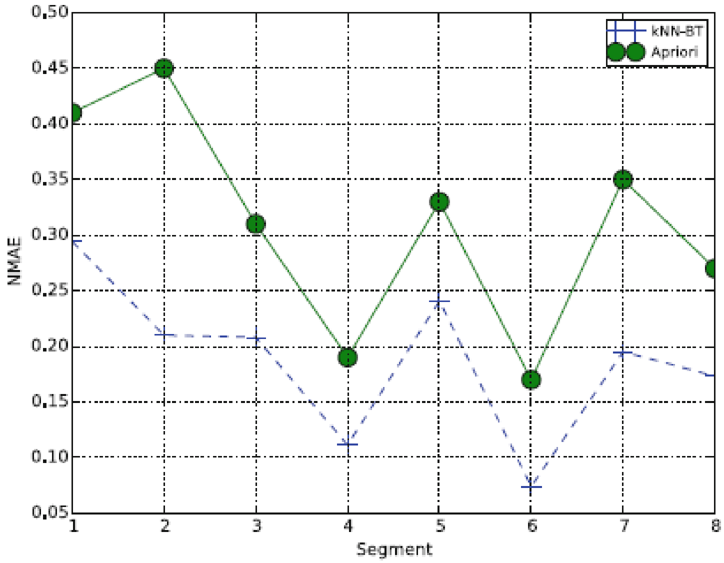
The majority of the bundles' attractiveness levels are around 0.5–0.6. Although MRFs are not suitable for detecting attractive bundles for relatively small group of visitors, it is still able to indicate that the tested CF-based dynamic bundling methods are able to deliver attractive bundles. This is important since the idea of CF is all about providing attractive suggestions. Thus, this analysis also supports the earlier results about the success of using CF for dynamic bundling.

The best performing CF-based dynamic bundling method, i.e. kNN-BT, is compared to Apriori based ARM approach. Figure 23.15 presents the NMAEs both for Cosine and Pearson. kNN-BT clearly outperforms ARM based dynamic bundling and its performance is statistically better within a 99% confidence level for the Wilcoxon rank sum test.

Fig. 23.15 Normalized Mean Absolute Error (NMAE) for evaluating the quality of the dynamic bundle suggestions of the best tested CF method and ARM based recommendation (the graphs are from Cosine and Pearson results respectively). (**a**) Cosine. (**b**) Pearson



(a)



(b)

23.5.4.1 Dynamic Recommendation as Static Recommendation

The reported static bundles are found by applying the Greedy heuristic. In order to make a direct comparison between static bundling and dynamic bundling, the same heuristic is used to offer dynamic bundles. Thus, dynamic bundling is thought as static bundling with prior preferences. However, the Greedy heuristic failed to generate comparable bundles. The one reason is that it adds attractions to the additional attractiveness of c/K. If there is no attraction that meets this criterion, it is unable to add new attractions to the bundle. Considering that the dynamic bundles offered by CF are highly attractive, this criterion fails to generate comparable bundles at most cases. Besides that, CF is able to offer large sized bundles mostly at size of 13 attractions. Since the total number of attractions 17, there is not much of attraction choices to vary. The aforementioned per attraction attractiveness constraint, c/K, becomes very challenging to meet. Thus, in order to suggest good dynamic bundles via static bundling, the corresponding static bundling algorithm should be more flexible, e.g. having a backtracking method.

23.6 Conclusion

This chapter introduces a data-driven approach to recommend personalized bundles for leisure parks using historical visitor trajectory data. Our idea is to first perform visitor segmentation using clustering. We then deliver personalized recommendations on the fly to make changes on existing ticket bundle. This is akin to the general idea in marketing of targeted upselling and cross-selling on the fly based on the specific visitor consumption pattern, backed by insights derived from a wealth of data collected from past visitor sale and consumption records. What set our work apart from many others is that we consider bundling and provide recommendation to a visitor *without* the need to take specific visitor information. While this idea has been studied under collaborative filtering that uses any preference or rating data to derive recommendations, it is mostly in the online world. We have adapted it for the physical world in a specific leisure park setting, which we believe to be first of its kind.

Our experimental results showed the correlation between bundle sizes, their diversity levels and costs for the static bundles. The low error rates achieved by the tested CF methods revealed the appropriateness of considering the bundling problem as a recommendation problem.

For future research, the performance of hybrid recommendation systems combining content-based and collaborative filtering should be investigated. The temporal dynamics based matrix factorization approaches will be additionally applied to offer dynamic bundles considering long-term visitor behavioural changes.

Acknowledgements

The authors would like to thank Shih-Fen Cheng, Pradeep Varankantham, and a number of un-named team members who contributed in various ways to this project. This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office, Media Development Authority (MDA).

References

Manoj K Agarwal and Subimal Chatterjee. Complexity, uniqueness, and similarity in between-bundle choice. *Journal of Product & Brand Management*, 12(6):358–376, 2003.

Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *ACM SIGMOD Record*, volume 22, pages 207–216. ACM, 1993.

Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499, 1994.

T. W. Anderson and D.A. Darling. Asymptotic theory of certain “goodness of fit” criteria based on stochastic processes. *The Annals of Mathematical Statistics*, 23(2):193–212, 1952.

Mark Armstrong. A more general theory of commodity bundling. *Journal of Economic Theory*, 148(2):448–472, 2013.

Alessandro Avenali, Anna DÁnnunzio, and Pierfrancesco Reverberi. Bundling, competition and quality investment: a welfare analysis. *Review of Industrial Organization*, 43(3):221–241, 2013.

Yannis Bakos and Erik Brynjolfsson. Bundling information goods: Pricing, profits, and efficiency. *Management Science*, 45(12):1613–1630, 1999.

Yannis Bakos and Erik Brynjolfsson. Bundling and competition on the internet. *Marketing Science*, 19(1):63–82, 2000.

Moran Beladev, Bracha Shapira, and Lior Rokach. Recommender systems for product bundling. *Knowledge-Based Systems*, 111:193–206, 2016.

James Bennett and Stan Lanning. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35, 2007.

D. Billsus and M.J. Pazzani. Learning collaborative information filters. In *Proceedings of the 15th International Conference on Machine Learning (ICML'98)*, pages 46–54, 1998.

Enrico Calandro and Chenai Chair. Policy and regulatory challenges posed by emerging pricing strategies. *Information Technologies & International Development*, 12(2), 2016.

Kathryn A Carroll, Anya Savikhin Samek, Lydia Zepeda, et al. Product bundling as a behavioral nudge: Investigating consumer fruit and vegetable selection using dual-self theory. In *2016 Annual Meeting, July 31-August 2, 2016, Boston, Massachusetts*, number 236130. Agricultural and Applied Economics Association, 2016.

Suchan Chae. Bundling subscription TV channels: A case of natural bundling. *International Journal of Industrial Organization*, 10(2):213–230, 1992.

Yongmin Chen. Equilibrium product bundling. *Journal of Business*, 70(1):85–103, 1997.

Yongmin Chen and Michael H Riordan. Profitability of product bundling. *International Economic Review*, 54(1):35–57, 2013.

Bing Tian Dai and Hady W Lauw. Modeling preferences with availability constraints. In *2013 IEEE 13th International Conference on Data Mining*, pages 101–110. IEEE, 2013.

Gary D Eppen, Ward A Hanson, and R Kipp Martin. *Bundling-new products, new markets, low risk*. Purdue University, Krannert Graduate School of Management, 1991.

David S Evans and Karen L Webster. Designing the right product offerings. *MIT Sloan Management Review*, 49(1):44, 2007.

Esther Gal-Or. Evaluating the profitability of product bundling in the context of negotiations. *The Journal of Business*, 77(4):639–674, 2004.

Joshua S Gans and Stephen P King. Paying for loyalty: Product bundling in oligopoly. *The Journal of Industrial Economics*, 54(1):43–62, 2006.

Xianjun Geng, Maxwell B Stinchcombe, and Andrew B Whinston. Bundling information goods of decreasing value. *Management science*, 51(4):662–667, 2005.

G. Hamerly and C. Elkan. Learning the k in k-means. In *Proceedings of the 17th Annual Conference on Neural Information Processing Systems (NIPS'03)*, pages 281–288, 2003.

Judy Harris and Edward A Blair. Consumer preference for product bundles: The role of reduced search costs. *Journal of the Academy of Marketing Science*, 34(4):506–513, 2006.

Michael D Johnson, Andreas Herrmann, and Hans H Bauer. The effects of price bundling on consumer evaluations of product offerings. *International Journal of Research in Marketing*, 16(2):129–142, 1999.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

V. Krishnan and Karl T. Ulrich. Product development decisions: A review of the literature. *Management Science*, 47(1):1–21, 2001.

Arthur Lewbel. Bundling of substitutes or complements. *International Journal of Industrial Organization*, 3(1):101–107, 1985.

Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.

Qi Liu, Enhong Chen, Hui Xiong, Yong Ge, Zhongmou Li, and Xiang Wu. A cocktail approach for travel package recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 26(2):278–293, 2014.

Silvano Martello, David Pisinger, and Paolo Toth. New trends in exact algorithms for the 0–1 knapsack problem. *European Journal of Operational Research*, 123(2):325–332, 2000.

R Preston McAfee, John McMillan, and Michael D Whinston. Multiproduct monopoly, commodity bundling, and correlation of values. *The Quarterly Journal of Economics*, 104:pages 371–383, 1989.

T.H. Nguyen, P.R. Varakantham, S-F. Cheng, and H.C. Lau. Mining resource bundles by balancing profitability. LARC-TR-01-14, Singapore Management University, 2014.

Dusit Niyato, Dinh Thai Hoang, Nguyen Cong Luong, Ping Wang, Dong In Kim, and Zhu Han. Smart data pricing models for the internet of things: a bundling strategy approach. *IEEE Network*, 30(2):18–25, 2016.

Brooks Pierce and Harold Winter. Pure vs. mixed commodity bundling. *Review of Industrial Organization*, 11(6):811–821, 1996.

Srinivasan Raghunathan and Sumit Sarkar. Competitive bundling in information markets: A seller-side analysis. *MIS Quarterly*, 40(1):111–131, 2016.

Elli Rapti, Anthony Karageorgos, and Georgios Ntalos. Adaptive constraint and rule-based product bundling in enterprise networks. In *IEEE 23rd International WETICE Conference*, pages 15–20. IEEE, 2014.

Jasson D.M. Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd International Conference on Machine learning*, pages 713–719. ACM, 2005.

P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM, 1994.

Francesco Ricci, Lior Rokach, and Bracha Shapira. *Introduction to recommender systems handbook*. Springer, 2011.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, pages 285–295. ACM, 2001.

Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 253–260. ACM, 2002.

Hanhuai Shan and Arindam Banerjee. Generalized probabilistic matrix factorizations for collaborative filtering. In *Proceedings of the IEEE 10th International Conference on Data Mining (ICDM'10)*, pages 1025–1030. IEEE, 2010.

Mehdi Sheikhzadeh and Ehsan Elahi. Product bundling: Impacts of product heterogeneity and risk considerations. *International Journal of Production Economics*, 144(1):209–222, 2013.

Sandro Shelegia. Multiproduct pricing in oligopoly. *International Journal of Industrial Organization*, 30(2):231– 242, 2012.

Shibin Sheng, Andrew M Parker, and Kent Nakamoto. The effects of price discount and product complementarity on consumer evaluations of bundle components. *Journal of Marketing Theory and Practice*, 15(1):53–64, 2007.

Allan D Shocker, Barry L Bayus, and Namwoon Kim. Product complements and substitutes in the real world: The relevance of "other products". *Journal of Marketing*, 68(1):28–40, 2004.

Nathan Srebro, Jason D.M. Rennie, and Tommi Jaakkola. Maximum-margin matrix factorization. In *NIPS*, volume 17, pages 1329–1336, 2004.

Stefan Stremersch and Gerard J Tellis. Strategic bundling of products and prices: A new synthesis for marketing. *Journal of Marketing*, 66(1):55–72, 2002.

Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009:4, 2009.

Chang Tan, Qi Liu, Enhong Chen, Hui Xiong, and Xiang Wu. Object-oriented travel package recommendation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(3):43, 2014.

M Mithat Üner, Faruk Güven, and S Tamer Cavusgil. Bundling of telecom offerings: An empirical investigation in the turkish market. *Telecommunications Policy*, 39(1):53–64, 2015.

Robin Van Meteren and Maarten Van Someren. Using content-based filtering for recommendation. In *Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop*, 2000.

R Venkatesh and Wagner Kamakura. Optimal bundling and pricing under a monopoly: Contrasting complements and substitutes from independently valued products*. *The Journal of Business*, 76(2):211–231, 2003.

R Venkatesh and Vijay Mahajan. The design and pricing of bundles: a review of normative guidelines and practical approaches. *Handbook of Pricing Research in Marketing*, page 232, 2009.

Yuanhong Wang, Yang Liu, and Xiaohui Yu. Collaborative filtering with aspect-based opinion mining: A tensor factorization approach. In *Proceedings of the IEEE 12th International Conference on Data Mining (ICDM'12)*, pages 1152–1157. IEEE, 2012.

Markus Weimer, Alexandros Karatzoglou, Quoc Viet Le, and Alex Smola. CofiRank: Maximum margin matrix factorization for collaborative ranking. In *Proceedings of the 21th Annual Conference on Neural Information Processing Systems (NIPS'07)*, 2007.

Wann-Yih Wu, Badri Munir Sukoco, Chia-Ying Li, and Shu Hui Chen. An integrated multi-objective decision-making process for supplier selection with bundling problem. *Expert Systems with Applications*, 36(2):2327–2337, 2009.

Min Xie, Laks VS Lakshmanan, and Peter T Wood. Breaking out of the box of recommendations: from items to packages. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 151–158. ACM, 2010.

Ruiliang Yan and Subir Bandyopadhyay. The profit benefits of bundle pricing of complementary products. *Journal of Retailing and Consumer Services*, 18(4):355–361, 2011.

Zhiwen Yu, Yun Feng, Huang Xu, and Xingshe Zhou. Recommending travel packages based on mobile crowdsourced data. *IEEE Communications Magazine*, 52(8):56–62, 2014.

Zhiwen Yu, Huang Xu, Zhe Yang, and Bin Guo. Personalized travel package with multi-point-of-interest recommendation based on crowdsourced user footprints. *IEEE Transactions on Human-Machine Systems*, 46(1):151–158, 2016.

Chengqi Zhang and Shichao Zhang. *Association rule mining: models and algorithms*. Springer-Verlag, 2002.

Mingyue Zhang and Jesse Bockstedt. Complements and substitutes in product recommendations: The differential effects on consumers' willingness-to-pay. In *Proceedings of the Joint Workshop on Interfaces and Human Decision Making for Recommender Systems co-located with ACM Conference on Recommender Systems (RecSys 2016), Boston, MA, USA, September 16, 2016.*, pages 36–43, 2016.

Sheng Zhang, Weihong Wang, James Ford, and Fillia Makedon. Learning from incomplete ratings using non-negative matrix factorization. In *SDM*, pages 549–553. SIAM, 2006.

Tao Zhu, Patrick Harrington, Junjun Li, and Lei Tang. Bundle recommendation in ecommerce. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 657–666. ACM, 2014.