Maximal Marginal Relevance-Based Recommendation for
Product Customization

Abstract:
Customized product design is attracting increasing attention because it allows companies to produce products that better satisfy customer requirements. However, consumers can be overwhelmed by the variety of products. To confront this challenge, this paper presents a two-step recommendation approach for customized products based on the idea of maximal marginal relevance. First, an adaptive specification process captures customer requirements in an accelerated manner by presenting the most informative attribute for a customer to specify. Then, a maximal marginal relevance-based recommendation set is presented, based on the customer’s partial specifications. This process ensures broad coverage of customers’ needs by considering not only the relevance of each product to their requirements but also redundancy in the recommendation set. Thus, products can be recommended to customers with limited input from them. A case study is presented to validate the specification and recommendation process.

Keywords: customization, product recommendation, probability relevance model

1. INTRODUCTION
The efficient design and manufacturing of custom products is an important area of research in the increasingly diversified and competitive global market (Chen et al., 2009; Wang and Li, 2021; Li et al., 2021). Product development teams must capture customers’ preferences and transform them into tangible product specifications effectively and efficiently (Pallant et al., 2020). However, customers may face an astronomical number of choices. For example, BMW has
claimed that the number of possible variations in the BMW 7 Series alone may be as high as $10^{17}$ (Zhu et al., 2008). Customers can be overwhelmed by the vast assortment of products and options.

Product recommendation has been accepted as a powerful tool to meet that challenge. It enables customers and the product development team to communicate directly with minimal input from middlemen such as sales and marketing personnel. This can shorten the product development process. Moreover, a well-established product recommendation process can help customers purchase satisfactory products more efficiently, shielding them from the tedious process of screening a vast number of products and selecting among them. Allowing customers to make informed decisions about purchases expeditiously is essential to the ultimate goal of customer satisfaction, increases the probability of cross-selling, enhances customer loyalty and reduces back-end costs associated with customer returns and warranty-related demands. With the rapid development of the Internet, recommendation systems have become an important tool for companies to better match their offerings to customers. Online product recommendation systems have been implemented on Amazon, eBay, Alibaba, jd.com and other online merchandising platforms.

However, the established recommendation approaches are designed primarily for off-the-shelf products such as books, movies and clothes. Adapting recommendation systems to customized products has been difficult due to the intrinsic differences in the elicitation of needs and presentation of results for customized versus off-the-shelf products. Customers’ choice behaviour in the context of online customization environments has been studied previously (Lu et al., 2020). Several issues must be addressed to facilitate online customization and help customers to meet their needs quickly.
• **Efficiency**: Online customers may not have the patience to configure a long list of items or choose among numerous product recommendations (Banister 2003). The proposed solution should be efficient and require minimal input from the customer.

• **New customers**: There may be insufficient information about new customers’ preferences to provide them with reliable recommendations, a phenomenon often referred to as the cold-start problem (Schein et al, 2002). We want to provide an approach that does not depend on information about a customer’s preferences.

• **Inconsistent specification**: Customers’ requirements and specifications can be incomplete or even contradictory if they lack expertise and domain knowledge. Therefore, it is necessary to develop a robust and reliable recommendation method for customized products based on incomplete or even distorted information about customer preferences.

To overcome these challenges, a marginal relevance model-based product recommendation method is presented in this paper. First, an adaptive specification approach is developed to elicit a customer’s preferences through a sequence of questions. It requires minimal prior knowledge about the customer. Information gain is adopted as the metric to measure the usefulness of each attribute in eliciting customer preferences. This guarantees that most uncertainty about a customer’s needs can be eliminated. Based on incomplete or even inconsistent specifications, the probability that each product will meet the customer’s specifications will be calculated. The recommendations are presented based on the probability of relevance. We try to maximize the probability of finding a satisfactory product for a customer among the top \( n \) recommendations. The optimization of this problem is NP-hard (Chen and Karger, 2006). Thus, we derive a maximal marginal relevance-based heuristic approach to approximate the optimal solution. The
algorithm not only considers the probability of relevance but also eliminates redundancy in the recommendation set by incorporating the novelty factor in the recommendation criteria.

The paper combines the adaptive specification approach with an effective recommendation set based on maximal marginal relevance. Numerical studies are also conducted to verify the proposed approach. This paper is organized as follows: Section 2 reviews the literature on the topic. Section 3 lays out the framework of the recommendation approach for customized products. Section 4 introduces the adaptive specification method. Section 5 presents the maximal marginal relevance-based recommendation process. Session 6 describes the case study and presents the conclusions.

2. Literature review

Recommendation methods have been widely studied in academia and applied in industry to help customers find their desired products. A product recommendation system tries to recommend items that are likely to meet a customer’s requirements. It serves as an information filter to present items that customers are interested in and acts as a bridge between customer attributes and design parameters (Tsang et al. 2021). The history of recommendation can be traced back to “word of mouth” since the emergence of human beings. People seek recommendations or comments from domain experts before they purchase goods. Product recommendations are a staple in almost all types of commercial transactions, especially e-commerce businesses.

Recommendation algorithms can be divided into memory-based and model-based approaches (Breese et al, 1998). Memory-based algorithms store comprehensive data and operate on them to make predictions. Typically, memory-based approaches have three phases: neighbourhood formation, pairwise prediction and prediction aggregation. The weighted average
of votes given to a product by other customers is computed as the vote of the active customer. The memory-based method has been extensively studied in recommendation systems. GroupLens (Resnick et al, 1994) and Ringo (Shardanand and Maes, 1995) were among the first to use memory-based recommendation approaches.

Model-based approaches use data on customer preferences to learn a general model that is deployed to predict a new customer’s preferences and provide recommendations. The model is usually trained offline over a long period. The recommendation is computed online as quickly and accurately as recommendations generated by memory-based methods (Breese et al, 1998). Examples of models used in recommendation systems include Bayesian networks, clustering techniques, neural networks, induction rule learning and linear classifiers (Breese et al, 1998; Billsus, 1998; Goldberg et al. 2001; Basu, 1998; Zhang, 2002). Breese et al. compared a number of model-based methods, including Bayesian clustering and decision-tree models. In their study, Bayesian network and correlation models outperformed other model types. Goldberg et al. (2001) proposed employing the algorithm Eigentaste, which uses universal queries to elicit real value ratings on a set of items. Principal component analysis (PCA) was exploited to facilitate dimension reduction for user classification and provide recommendations. Billsus and Pazzani (1998) constructed some feed-forward neural networks to make recommendations. Each user in the database had a corresponding network that was trained by backward propagation. The network could map an unknown item to a rating that was used to provide recommendations. The approach had fairly good accuracy but the cost of training the set of neural networks was too high. Zhang et al. (2002) explored the use of various linear classifiers in generating recommendations, treating the process as a classification task. A comparison with a decision tree model and other memory-based models showed that the linear model was well suited for
providing recommendations. It outperformed memory-based models in accuracy and offered a better balance between offline training and online computation.

Product recommendation systems have also been studied by marketing researchers. Bruyn et al. (2008) leveraged preference models by using conjoint analysis to design a questionnaire-based decision tree to help capture customers’ preferences based on information such as demographics, product usage, and self-reported preferences. They compared cluster classification, Bayesian tree regression and stepwise componential regression and developed an optimal sequence of questions to predict customer preferences. Their approach provided recommendations with minimal customer input. Ghose et al (2012) applied the idea that products that have a higher surplus should be more relevant to customer queries and proposed a ranking system that provides the best value for the money. They applied the system to a hotel ranking case and provided customers with the “best-value” hotels beforehand. Backhaus et al (2010) not only considered customer preferences using conjoint analysis when building a recommendation system but also incorporated the customer’s willingness to pay, allowing for dynamic pricing of the product. They applied conjoint analysis and a stepwise componential segmentation approach to collect data and based the recommendation system on collaborative filtering. Similarly, Zhang et al (2012) considered price as an attribute in a recommendation system for books. Dummy variable regression was deployed to analyse data from Amazon, making it possible to prioritize a variety of attributes in the recommendation process.

Most recommendation approaches try to find products for customers. Scholz et al (2012) attempted to identify the optimal number of recommendations that customers prefer to see. They used a signal detection theory-based model to estimate the number of recommendations for customers and applied multi-attribute value theory to address the cold-start problem and
changing preferences (Scholz, 2016). To enhance prediction accuracy, Shahbazi et al. (2020) combined extreme gradient boosting machine learning architecture to explore purchased products based on users’ click patterns.

3. The framework of the proposed recommendation approach.

A two-step approach is proposed to tackle the issue of product recommendation for custom products, containing an adaptive specification and recommendation process. This sequential, two-phase process involves an iterative procedure of preferences elicitation and recommendation. A schematic drawing of the product customization process is shown in Figure 1. Initially, the product development team captures a customer’s specification for one attribute. Recommendations are provided based on this partial information. The customer can terminate the custom product-design process if he or she is satisfied or fine-tune the recommendations by specifying more attributes. Although the first product recommended may not be the optimal item for the customer, satisfactory products can suffice, given time and search cost constraints (Bordley et al., 2000). Customers’ specifications and data on their final choices are stored in the configuration database so that the model parameters can be updated.
4. Adaptive specification to elicit customer needs

We use an adaptive specification process to elicit customer’s needs step by step as stated in (Wang and Tseng, 2007). A well-designed specification process can capture customer needs without tedious rounds of communication. Most customers are not patient enough to indicate their preferences for long lists of items, and they may have difficulty in stating preferences for unfamiliar items (DeBruyn et al, 2008). In addition, the amount of information available about product attributes or components differs greatly. We will exploit these properties to improve the efficiency of the specification process.

The product specification process is considered as a sequence of questions and answers. A customer’s preferences and requirements for a product are unknown to the designers before the specification process begins. The questions try to capture customers’ needs efficiently and
accurately. During the specification process, a customer specifies the attributes of the product one by one, and the designers discover the customer’s needs gradually based on the partial specifications. The more attributes the customer configures, the more information about the customer’s needs is obtained. From the designers’ point of view, the specification procedure is an uncertainty elimination process. Decision-making theory dictates that we should eliminate the maximum uncertainty about a customer’s need in each round of communication to make the design process more efficient. Therefore, only the most informative item is proposed from the pool of attributes in each round.

Let a variable $A_i$ represent a component\(^1\) of a product. $A_i$ takes on the values from its specifications set, i.e. the corresponding variable domain, $\text{dom}(A_i) = \{a_{i1}, a_{i2}, \ldots, a_{im_i}\}$, where $m_i$ is the number of choices for attribute $A_i$. Without loss of generality, we assume that all of the attributes in the product are properly numbered. Thus, each product configuration (i.e., product variant) can be represented by $C = (a_1, a_2, \ldots, a_n)$, where $n$ is the number of attributes in the product. $a_i$ is an instantiation of the corresponding variables vector, i.e. $a_i \in \text{dom}(A_i)$. For example, a PC product can be generally represented by the vector (CPU, hard disk, RAM, display card, \ldots, monitor). One PC configuration may be $C = (i7\text{ processor}, 1\text{TB hard disk}, 8\text{GB RAM}, \ldots, 21’\text{ monitor})$.

---

\(^1\) In the literature, the terms ‘attribute’, ‘component’ and ‘feature’ are sometimes used interchangeably to represent the basic element of a product. In this paper, we mainly use the term ‘component’.
In information theory, entropy is used to describe the amount of uncertainty contained in a random variable. For a discrete random variable, \( X \), entropy is defined as:

\[
H(X) = -\sum_i P(X = x_i) \log P(X = x_i) \quad \text{(McEliece, 2002)}.
\]

The information gained about \( X \) by knowing the value of another variable, \( Y \), can then be quantified by \( H(X) - H(X \mid Y) \), where \( H(X \mid Y) = \sum_j P(Y = y_j)H(X \mid Y = y_j) \). Before the product customization process, the total uncertainty about what the customer needs in the product, from his or her perspective, is

\[
H(A_1, A_2, \ldots, A_n) = -\sum_{a_1^{*} \in \text{dom}(A_1)} \cdots \sum_{a_n^{*} \in \text{dom}(A_n)} P(A_1 = a_1^{*}, A_2 = a_2^{*}, \ldots, A_n = a_n^{*}) \log P(A_1 = a_1^{*}, A_2 = a_2^{*}, \ldots, A_n = a_n^{*})
\]

where \( a_i^{*} \) is the possible value of attribute \( A_i \). After round \( t \) of the specification process, the total uncertainty about the customer’s needs is reduced to \( H(A_1, A_2, \ldots, A_n \mid S_t) \), where \( S_t \) is the set of specifications after round \( t \). In round \( t+1 \), a new attribute, \( A_{t+1} \), will be selected for the customer to specify. As the customer can specify any value from the domain of \( A_{t+1} \), the expected uncertainty after the configuration round \( t+1 \) can be calculated by

\[
H(A_1, A_2, \ldots, A_n \mid S_t, A_{t+1}) = \sum_{a_{t+1}^{*} \in \text{dom}(A_{t+1})} P(A_{t+1} = a_{t+1}^{*})H(A_1, A_2, \ldots, A_n \mid S_t, A_{t+1} = a_{t+1}^{*}).
\]

Thus, we select the attribute based on the maximization of information gain from the specification of \( A_{t+1} \), i.e.

Maximise \( H(A_1, A_2, \ldots, A_n \mid S_t) - H(A_1, A_2, \ldots, A_n \mid S_t, A_{t+1}) \). The specification procedure can be outlined as follows:

- **Input:** the component variables of a customized product \( (A_1, A_2, \ldots, A_n) \), the choices of each attribute \( \text{dom}(A_i) = \{a_{i1}, a_{i2}, \ldots, a_{i\text{m_i}}\} \) and a customer’s partial specification \( S_t \) (\( S_0 \) is defined as an empty set).
Output: the attribute to be selected for the customer to configure in round \( t+1 \), i.e., \( A_{t+1} = \arg \max_{A_C \in \{A, A_2, \ldots, A_n\} \cup S_t} [H(A_1, A_2, \ldots, A_n | S_t) - H(A_1, A_2, \ldots, A_n | S_t, A_C)] \)

\[(1)\]

where \( A_C \) is the candidate component selected in round \( t+1 \) for the customer to specify. After obtaining the specification of \( A_{t+1} \), the specification set will be updated as \( S_{t+1} = S_t \cup \{a_{t+1}\} \).

It can be proved that \( H(X | Y) = \sum_i \sum_j P(X = x_i, Y = y_j) \log \frac{P(Y = y_j)}{P(X = x_i, Y = y_j)} \) (Blachman, 1968). Thus, to calculate \( H(A_1, A_2, \ldots, A_n | S_t) \), we must know the joint probabilities \( P(A_1, A_2, \ldots, A_n) \). The maximal likelihood estimator for the joint probabilities can be derived as

\[\hat{P}(A_i = a_{i1}, A_2 = a_{2i}, \ldots, A_n = a_{ni}) = \frac{\#(\text{specifications with value } A_i = a_{i1}, A_2 = a_{2i}, \ldots, A_n = a_{ni})}{\#(\text{all the specifications})} \] (Wang and Tseng, 2007). The probabilities needed to calculate \( H(A_1, A_2, \ldots, A_n | S_t, A_C) \) can be estimated in a similar way.

5. Maximal Marginal Relevance-Based Product Recommendation

5.1 Degree of relevance of the recommended items

In this section, we quantify the relevance of a recommendation by the probability of its relevance. Let \( I_R \) be a relevance indicator with value 1 indicating that a product variant, \( C \), will
meet the customer’s specification $S$ and value 0 otherwise. Then, $P(I_R = 1 | C, S)$ refers to the probability that $C$ will meet a customer’s specification, $S$, which may not be partial specifications of the product. According to Bayes’ rule, $P(I_R = 1 | C, S) = \frac{P(I_R = 1 | S)P(C | I_R = 1, S)}{P(C | S)}$. The corresponding odds function is

$$O(I_R = 1 | C, S) = \frac{P(I_R = 1 | C, S)}{P(I_R = 0 | C, S)} = \frac{P(I_R = 1 | S)P(C | I_R = 1, S)}{P(I_R = 0 | S)P(C | I_R = 0, S)} = O(I_R = 1 | S) \frac{P(C | I_R = 1, S)}{P(C | I_R = 0, S)}.$$  

As the odds function is monotonic with respect to the probability, they are equivalent in terms of quantifying the degree of relevance.

Assuming linked independence of the attributes, we get

$$P(C | I_R = 1, S) = \prod_{i=1}^{n} P(a_i | I_R = 1, S) \quad \text{where} \quad a_i \text{ is a binary indicator indicating whether the } i\text{-th component’s specification is in } C.$$  

The linked independence assumption means that the occurrence of different components’ specifications in a product variant is independent. This assumption has been widely used in the marketing literature (Kamakura et al, 1995).

Let $p_i = P(a_i = 1 | I_R = 1, S)$ and $q_i = P(a_i = 1 | I_R = 0, S)$. Then we can yield

$$O(I_R = 1 | C, S) = O(I = 1 | S) \prod_i p_i^{a_i} (1 - p_i)^{1 - a_i} q_i^{a_i} (1 - q_i)^{1 - a_i}.$$  

Taking a logarithm of both sides of this equation, we can get

$$\log O(I_R = 1 | C, S) = \sum_i \log \frac{p_i (1 - q_i)}{q_i (1 - p_i)} a_i + \sum_i \log \frac{1 - p_i}{1 - q_i} + \log O(I_R | S).$$  

---

2 The recommendation is made after each round of specification. Thus, the spec definition round $t$ does not affect the recommendation mechanism. To simplify the notation and make the content more readable, the index $t$ of $S$ used in the previous section is omitted here.
The second and third terms in this equation do not affect the ranking of odds for product variant \( C \) as they do not consist of the value of \( a_i \). Thus, we only consider the first term when making the recommendation. It is a linear function of \( a_i \)

\[
F(C,S) = \sum_i \log \frac{p_i(1-q_i)}{q_i(1-p_i)} \cdot a_i
\]

5.2 Marginal relevance-based recommendations

In a traditional product customization system, such as a configurator, customers need to specify all of the attributes or components. To minimize customer input, it is necessary to present a set of recommendations from which he or she can choose after each round of configuration or specification. Previous research on recommendations aimed at maximizing the number of relevant recommendations (Wang and Tseng, 2013). However, research on information retrieval has shown that the attempt to return more potentially relevant items can, counterintuitively, reduce the chances of finding a relevant product (Chen and Karger, 2006). If the model of relevance is wrong, the set of recommendations will return an entire list of irrelevant items. Thus, we will apply the idea of maximal marginal relevance (MMR) to the recommendation phase of our model. Initially, MMR was conceived in document retrieval research to identify relevant files (Carbonell et al, 1998). We modify it to fit a product-recommendation scenario. The aim of an MMR-based recommendation system is to maximize the likelihood that a relevant product is recommended while reducing redundancy (or increasing the degree of novelty) in the sets of recommended items.

Let \( R_{n-1} \) be the set of existing \( n-1 \) recommended items. Then the task is to determine which product to recommend next, i.e. the \( n \)-th recommendation in the recommendation list, to form the recommendation set \( R_n \). When presenting the \( n \)-th recommended item, we need to
maximize the probability of relevance, i.e. \( \max_{C_j \in C \setminus R_{n-1}} F(C_j, S) \) where \( C \) is the total set of product variants. In addition, the \( n \)-th recommendation should minimize redundancy with the existing recommendations \( R_{n-1} \). In other words, the newly added recommendation should maximize the marginal relevance of the recommendation set and contain more novelty with respect to \( R_{n-1} \).

The degree of novelty is negatively correlated with the degree of redundancy. Therefore, we need to quantify to what extent the \( n \)-th recommendation is similar to the top \( n-1 \) recommendations. If two recommended product variants are more similar or relevant, then there is more redundancy between these two items. Thus, we define the degree of novelty for the \( n \)-th recommendation as \( -\min_{C_s \in C \setminus R_{n-1}, C_j \in R_{n-1}} F(C_n, C_j) \). This quantifies the closest distance between the \( n \)-th recommended product variant and the existing recommendation set \( R_{n-1} \). The bigger the distance is, the more novelty \( C_n \) contains. Thus, a negative sign is added to transform the distance into the degree of novelty.

By combining the degree of relevance and the degree of novelty in the recommendation framework, we can obtain the recommendation criteria for the \( n \)-th item, \( \alpha \max_{C_s \in C \setminus R_{n-1}} F(C_n, S) - (1 - \alpha) \min_{C_s \in C \setminus R_{n-1}, C_j \in R_{n-1}} F(C_n, C_j) \). The first term quantifies the relevance of the \( n \)-th recommendation with respect to the specifications provided by the customer. The second term quantifies the degree of novelty of the \( n \)-th recommendation with respect to the existing recommended items \( R_{n-1} \). \( \alpha \) is a weight determining the relative importance of the degrees of relevance and novelty in the recommendation. We can see that the method proposed in (Wang and Tseng, 2013) is a special case of this, with parameter \( \alpha \) being 1.
It should be noted that the maximum marginal relevance approach, which is calculated using probabilistic analysis, could leverage the flexibility of customers’ choices and preferences. Customers' choices can be flexible (Moodie and Bobrowski 1999; Wang et al., 2021). For example, a survey of 1033 new car buyers in the UK, conducted in 2000 and 2001, revealed that 22% of the customers accepted alterations in their original product specifications (Holweg and Pil 2004). Thus, although a customer specifies one attribute in specification scenarios, alternatives may be acceptable. We incorporate this flexibility into the marginal relevance model, quantifying it as the probability of the acceptance of a product. Thus, the approach offers a broader potentially acceptable product set.

6. Numerical example

6.1 Introduction

We use a PC recommender to test the viability and performance of our approach. We treat a PC as a combination of six components: the processor, monitor, hard disk, display card, memory and display driver. There are six component specifications for the processor, three for the monitor, three for the hard disk, three for the display card, four for the memory and five for the display driver.

The sets of components and their number of specifications are shown in Table 1. The recommendation process forms an iteration loop as shown in Figure 1. Each time a new customer specifies a component, a recommendation will be presented. If the customer is satisfied with it, he or she can terminate the process. Otherwise, the customer can refine the recommendation by providing more specifications, and a new round of recommendations will be presented. In the worst case, all six components must be specified, which indicates that the customer was not satisfied with the recommendations offered in any of the rounds. In this experiment, we use the
number of communication rounds, i.e. the number of specifications needed to be inputted, as the metric of the recommendation’s efficiency. Fewer rounds of recommendations indicate that a recommendation method is more efficient.

Data on product specifications and accepted recommendations were obtained for 69 customers. To deal with the data sparsity problem, we use the perturbative bootstrap method proposed in (Wang and Tseng, 2011) to generate ten sets of specification-recommendation data pairs. Each set has 207 data pairs. We use 10-fold cross-validation to estimate the parameters and test the method. This means that we conduct the experiment 10 times, and each time 9 sets of data are used as a training set and the remaining set is used for testing. Due to space limitations, the conditional probabilities estimated from each training data set are omitted here.

The estimation of parameter is as follows. Suppose there are \( N \) product configurations and \( n \) satisfactory specifications. Let \( N_i \) be the number of configurations consisting of item \( i \) among which \( n_i \) can satisfy the specification, then \( p_i = \frac{n_i}{n} \) and \( q_i = \frac{N_i - n_i}{N - n} \) which are the maximum likelihood estimations of \( p_i \) and \( q_i \) in equation (1). Some smoothing corrections are needed to avoid a denominator value of 0. We use \( p_i = \frac{n_i + \epsilon}{n + \epsilon} \) and \( q_i = \frac{N_i - n_i + \epsilon}{N - n + \epsilon} \) as the estimator. In this paper, \( \epsilon \) is set to be 0.5.

6.2 The effect of \( \alpha \) values on the recommendation performance

We use expected search length as the performance measure. Expected search length is widely used in document retrieval research to quantify the number of documents that need to be evaluated before the desired item is found (Cooper, 1968). In the scenario of customized product
recommendation, multiple products can be recommended according to their probabilities of relevance and the redundancy in the recommendation list. If a customer’s target PC is in the set of recommendations, the process will end, and the corresponding configuration rounds will be recorded. Thus, the expected search length is equivalent to the number of communication rounds in the recommendation process. If the framework performs better, fewer rounds of communication should occur. In this section, we examine the effect of different $\alpha$ on the recommendation result. The results are shown in Table 1.

<table>
<thead>
<tr>
<th>$\alpha$ value</th>
<th># of communication rounds with 1 recommendation in each round</th>
<th># of recommendation rounds with 3 recommendations in each round</th>
<th># of recommendation rounds with 5 recommendations in each round</th>
<th># of recommendation rounds with 7 recommendations in each round</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5.16</td>
<td>5.11</td>
<td>4.96</td>
<td>4.86</td>
</tr>
<tr>
<td>0.2</td>
<td>4.92</td>
<td>4.81</td>
<td>4.7</td>
<td>4.65</td>
</tr>
<tr>
<td>0.4</td>
<td>4.59</td>
<td>4.3</td>
<td>4.13</td>
<td>3.79</td>
</tr>
<tr>
<td>0.6</td>
<td>4.34</td>
<td>4.09</td>
<td>3.76</td>
<td>3.29</td>
</tr>
<tr>
<td>0.8</td>
<td>4.38</td>
<td>4.07</td>
<td>3.71</td>
<td>3.31</td>
</tr>
<tr>
<td>1.0</td>
<td>4.67</td>
<td>4.15</td>
<td>3.84</td>
<td>3.39</td>
</tr>
</tbody>
</table>

Table 1. The average number of recommendations rounds with respect to different parameter settings
When \( \alpha = 1 \), the approach is degenerated to the method used in (Wang and Tseng, 2013), meaning that the redundancy in the recommendation list is not considered at all. When the value of \( \alpha \) decreases, the novelty term gains more weight. But when the \( \alpha \) value is too small, the probability of relevance will be dominated by the novelty term; consequently, this approach tends to recommend fewer product variants. Thus, we can observe that the number of communication rounds will decrease first and then increase in each column of Table 1. A trade-off should be made between the relevance and novelty of the recommended items. Based on the table, we can observe that when \( \alpha \)’s value is 0.8 or 0.6, the overall performance is the best. Thus, we choose \( \alpha = 0.8 \) for our subsequent experiments.

6.3 The effect of adaptive specification

In these experiments, we test the performance of four approaches that differ in their specification and recommendation methods. Based on the number of attributes in table 1, there are 6! = 720 possible query sequences for customers to configure their PCs. We calculate the average communication rounds of all the possible sequences and use two recommendation approaches, namely the probability ranking principle (PRP) used in (Wang and Tseng, 2013) and MMR. In the MMR methods, 0.8 is chosen as the \( \alpha \) value. These two methods are abbreviated as AverFixSeq+PRP and AverFixSeq+MMR. We use the adaptive specification process for the other two approaches. The recommendation approaches remain PRP and MMR and the \( \alpha \) value 0.8. These two specification and recommendation methods are represented by Adap+PRP and
Adap+MMR. 10-fold cross-validation is used for the testing. The average number of communication rounds is recorded. Figure 2 shows the results when employing the different approaches.

The x-axis is the number of recommendations in each round and the y-axis represents the number of recommendation rounds needed for the customer to find a satisfactory product. We can see that the “Adap+MMR” approach outperforms the others. The number of communication rounds decreases with the increasing number of recommendations per round because a longer recommendation list is more likely to contain a satisfactory product. The experiment results indicate that our approach holds promise for improving the efficiency of customized product specification and recommendation.
Figure 2: A comparison of the recommendation approaches

7. CONCLUSION

In this paper, an adaptive method of product specification is proposed. It enables the use of a customized query sequence to reduce redundant questions in the specification procedure. A maximal marginal relevance model is adopted to calculate the probability that a given product will meet a particular customer’s needs based on his or her partial specifications during the specification stage. A query will then be selected based on the amount of information that can be harvested. As soon as the information content reaches a certain threshold, product recommendations will be extended to elicit customer feedback. The product with the highest probability of relevance will receive the highest recommendation. If the recommendation is satisfactory, the product customization process can be terminated. Otherwise, the customer can fine-tune the recommendations by providing more specifications. The specification and recommendation form a closed loop to accelerate the elicitation of customer needs and customization of the product; the process may motivate customers to participate in co-creation (Mandolfo et al., 2020). The viability and performance of the approaches were tested using a numerical example. The experiment showed that the adaptive method outperformed benchmark recommendations significantly.
References


Li, Qing, Wei, Hailong, Yu, Chao and Wang, Shuangshuang. 2021 “Data and model-based triple V product development framework and methodology.” Enterprise Information Systems, DOI: 10.1080/17517575.2020.1867900


framework in the furniture industry.” Journal of Manufacturing Systems, DOI:

van Rijsbergen, C. J. 1977. “A theoretical basis for the use of co-occurrence data in information


Wang, Y. and X. Li. 2021. “Mining Product Reviews for Needs-Based Product Configurator
Design: A Transfer Learning-Based Approach.” IEEE Transactions on Industrial Informatics,
17(9): 6192 – 6199.

Wang, Y., X. Li and D. Mo. 2021. “Knowledge-Empowered Multi-Task Learning to Address the
Semantic Gap Between Customer Needs and Design Specifications.” IEEE Transactions on
Industrial Informatics, accepted, DOI: 10.1109/TII.2021.3067141.

Proceedings of IEEE International Conference on Industrial Engineering and Engineering

relevance model.” Journal of Intelligent Manufacturing, 24: 951-960

recommendation.” Proceedings of IEEE 3rd International Conference on Software Engineering
and Service Science (ICSESS): 681-684