J Intell Manuf (2013) 24:951–960
DOI 10.1007/s10845-012-0644-7

Customized products recommendation based on probabilistic relevance model

Yue Wang · Mitchell M. Tseng

Received: 31 October 2011 / Accepted: 7 April 2012 / Published online: 21 April 2012

Abstract Product customization is attracting more attentions in industry as a viable strategy to better meet customer requirements and gain more profit. However the vast number of product variants in product customization process often makes it difficult for consumers to make purchase decisions, a phenomenon referred to as information overload. In this paper we take a two-prong approach to tackle the issue of information overload in customized products recommendation. Basically, the method answers two questions, namely, which products to recommend and in what order to present the recommendations. Firstly, a probability relevance model is deployed to calculate the probability of relevance for each end product. Then a probability ranking principle is exploited to present the recommendations. The approach also takes customer flexibility into consideration and thus mitigates the effect of inconsistent specifications from customers. It does not require any prior knowledge about an active customer's preference and can accommodate the new customers challenge facing by recommendation approaches. Analytical results show that the method is optimal in terms of customer's utility and product recommendation efficiency. Numerical experiments are also conducted to test the presented approach.

Y. Wang (✉) · M. M. Tseng
Advanced Manufacturing Institute, The Hong Kong University of Science and Technology, Kowloon, Hong Kong
e-mail: yacewang@ust.hk

M. M. Tseng
e-mail: tseng@ust.hk

Y. Wang
Fok Ying Tung Graduate School, The Hong Kong University of Science and Technology, Kowloon, Hong Kong

Keywords Product design · Customized product recommendation · Probability ranking principle · Probability relevance model

Introduction

Nowadays, customer centric product development is gaining more attentions in the competitive global marketplace. The fulfillment of customer needs becomes a key factor for the customer's purchase decision (Risdiyono and Koomsap 2011). In this situation, product customization has been widely accepted in industry as a viable strategy to better meet customer requirements and gain more profits (Aldanondo and Vareilles 2008). There is a growing trend of customized products coming into the market, ranging from consumer products like consumer electronics, sports shoes, golf clubs, personal computers, apparels and automobiles to industrial products like professional fridge systems, telecommunication systems, cargo ships, escalators, airplanes, etc. The Federal Reserve of Dallas reported that since 1970s, the number of product varieties has increased sharply, for example, PCs from 0 to 400, car models from 140 to 260, car styles from 654 to 1,212 by 1998 (Cox and Alm 1998). It is also reported that the number of possible variations in the BMW 7 Series alone could reach 10^{17} (Zhu et al. 2008).

However, product proliferation as a result of product customization often makes it difficult for the customer to make a decision when facing a vast variety of products and the wide assortment of options, a phenomenon referred to as information overload. To mitigate the effect of information overload, product recommendation methods have been widely adopted in e-commerce business. However, the established product recommendation approaches are primarily for off-the-shelf products like books, movies, and CDs. Adaptation for customized products has been difficult because of the unique properties of customized products. For example, current recommendation for off-the-shelf products cannot handle *new customers* issue well because the underpinning assumption of most existing product recommendation approaches is that if two customers have a similar inclination for some items, they should have similar tastes and choices for other products. Existing methods are difficult to apply to a new customer in that there is no prior information about his/her preferences. In addition, the recommendation scenarios also differ. Recommendation process for customized products is a back and forth communication procedure. It requires a sequential decision making process during which the active customer's preference information is discovered gradually and end products are recommended accordingly. Thus customer preferences are elicited explicitly. The process is different from current online recommendation systems for off-the-shelf products which try to implicitly capture preferences by mining customer profile information, previous purchasing history, etc.

Basically, we seek a recommendation approach satisfying the following challenges which often exist in personalized product design practice:

- *New customers*: Since current recommendation approaches depend largely on prior knowledge of active customer's preferences, they may not be applicable for new customers. In this research, we want to develop methods which are free of prior preference information about a particular customer so as to overcome the new user problem.
- *Inconsistent specification*: A customer's specifications can be incomplete and ambiguous. Sometimes, the specifications even contradict with each other due to the lack of domain knowledge or other reasons. These facts further complicate the recommendation task. Therefore, the recommendation method should be robust enough to handle the incomplete or even inconsistent specification for the customized product.

In this paper we take a two-prong approach of personalized recommendation for customized products to tackle the issue of information overload. Basically, the method answers two questions, namely which products to recommend and in what order to present the recommendations. To solve the first question, a probability relevance model is deployed to calculate the probability (odds) of relevance for each end product. We notice that customer preferences may be flexible in the sense that although one alternative attribute is selected, other alternatives may also be acceptable. The probability (odds) of relevance based approach can quantify the likelihood that other attribute alternatives are also satisfactory by studying previous customers' choices data. The probability (odds) is adopted in the recommendation. Therefore our method takes the advantage of customer flexibility to calculate probability of relevance. Then a probability ranking principle is exploited to solve the second question. The idea is to provide recommendations according to the rank of probability of relevance. Although the recommendations may not be the best matched product for the customer, research has long acknowledged that when searching for a product, customers do not necessarily attempt to find the optimal solution due to time and searching cost constraints (Wright 1975). Identifying a satisfactory alternative may suffice, a process called "satisfying" by Simon (1957).

The paper is an extension of the methods in (Wang and Tseng 2009) by presenting analytical results of recommendation. Some comprehensive numerical studies are also conducted to verify the proposed approach. The paper is organized as follows; a brief introduction of related work is presented in "Literature review". "Probability relevance model" gives an introduction to probabilistic relevance model. Probability ranking principle and some of its properties are presented in "Probability ranking principle". "Numerical example" is the numerical example to demonstrate the viability of the approach. The paper is concluded in "Concluding remark".

Literature review

Product recommendation has been a staple in almost all types of commercial transactions. Recommendation methods have been widely studied to help customers find their desired products more efficiently and effectively than a traditional brute force search. Microsoft has even incorporated recommendation capabilities into their commerce software for servers (Peddy and Armentrout 2003). Basically, a recommendation is a special kind of information filter, trying to present items which are likely to meet a customer's requirements. "Word of mouth" is the original form of recommendation system, being used since the emergence of human beings. People consult domain experts for a product or service and get appropriate recommendations. With the development of the internet and information technology, automatic recommendation approaches have emerged to handle the information overload in current marketplace, especially e-commerce business. Examples of current online product recommendation system include Ebay, Amazon, Netflix (www.netflix.com), gifts.com, GiveToThem.com etc.

Memory-based and model-based recommendation approaches have been studied in literatures. Memory-based recommendation algorithms usually contain three phases, neighborhood formation, pairwise prediction, and prediction aggregation. It uses the weighted average of evaluations from other customers as the criterion of product recommendation. Examples of memory-based recommendation system include

GroupLens (Resnick et al. 1994) and Ringo (Shardanand and Maes 1995). Instead of working on the original data directly as memory-based approach, model-based recommendation approaches use the data on customer preferences to learn a general model, which is deployed to predict a new customer's preference and provide recommendations. The model is usually learned off-line over a long period of time. The recommendation is computed online, which is fast and as accurate as memory-based methods. Examples include Bayesian networks (Breese et al. 1998), clustering techniques (Breese et al. 1998), neural networks (Billsus and Pazzani 1998), induction rule learning (Basu et al. 1998), and linear classifiers (Zhang and Iyengar 2002). However, both kinds of recommendation approaches are for off-the-shelf products. In addition, they require prior knowledge of customers' rating or ranking to products. Thus they cannot cope with the new customer challenges very well.

Methods have also been investigated for customized products recommendation. Moon et al proposed an agent based recommender system to develop customized families of products (Moon et al. 2009). They determine customers' preferences for product recommendation according to a market-based learning mechanism in dynamic electronic market environment. A mass customization recommendation system is recently introduced in (Mavridou et al. 2011). Stormer applies collaborative recommendation system to calculate the most common product options and propose them as recommendations for customers during product design phases (Stormer 2009).

Unlike the aforementioned methods, this paper present a statistics based approach to tackle the customized products recommendation issue. Customers' preferences information is learned from existing product selection data. After getting partial specifications of a product from a new customer, the prior knowledge of preferences will be deployed to present recommendations accordingly. Therefore, we don't need any extra prior information of the new customer to generate recommendation.

Probability relevance model

Model set up

In this paper, we investigate the product recommendation task in the context of configuration specification definition. It means the possible attributes and components set of a product are predefined. An end product is a combination of the required attributes. Many current online configuration systems have adopted such toolkit to realize product customization, like Dell computer's online component selection system. Our recommendation approach is operated in such an environment. Each time a customer input some specifications to the product, an updated recommendation list will be present. The recommendations can be updated and refined if more specifications are inputted from the customer. Thus the approach integrates the needs elicitation and product recommendation in a unified framework.

To implement the method, previous customers' preferences and purchasing records data are needed. Here we assume customer preferences are expressed in detailed product specification level, i.e., the specification of each attribute or component. Specifically, a specification-configuration (S-C) data pair is associated with each customer. The specification (S) represents a customer's desired product. S is expressed by the customer step by step in specification definition process. The product configuration (C) stands for the customer's final selection of the product in the form of an instantiation of a set of components. Let $C = (\vec{A}_1, \vec{A}_2, \ldots \vec{A}_n)$ denote an end product configuration where $\vec{A}_i$ is the ith attribute/component vector, i.e., $\vec{A}_i = (a_{i1}, a_{i2}, \ldots a_{in_i})$. a_{ij} is the indicator with value 1 representing the existence of the ith attribute's jth alternative and value 0 otherwise. Thus in *one* product configuration, only *one* alternative of an attribute can have value $a_{ij} = 1$ and others should be 0, i.e., $\sum_j a_{ij} = 1$ for any i. In this sense, each configuration is represented by a serial of binary codes in the form of $C = (a_{11}, \ldots a_{1n_1} \ldots a_{n1}, \ldots a_{nn_n})$ with constraint $\sum_j a_{ij} = 1$ for any i. Similarly, the specification can also be represented in the same way.

Based on previous customers' specification-configuration (S-C) data pairs, the probability of relevance will be calculated as follows according to probability relevance model (Croft and Harper 1979; Harper and van Rijsbergen 1978; Crestani et al. 1998). Let $P(I = 1|C, S)$ be the probability of relevance, standing for the probability that a product configuration C will meet a customer's specification S. I is an indicator with value 1 representing that the configuration will meet the customer's requirements and 0 otherwise.

What we are interested in is to get the value of $P(I = 1|C, S)$ for each configuration C. Applying Bayes' rule, we can get

$$P(I|C, S) = \frac{P(I|S)P(C|I, S)}{P(C|S)} \tag{1}$$

Since the denominator does not have a clear meaning and is hard to calculate, we estimate the odds instead of the probabilities to eliminate the term $P(C|S)$,, i.e.,

$$O(x) = \frac{P(x)}{P(\overline{x})} = \frac{P(x)}{1 - P(x)} \tag{2}$$

It can be shown that $O(x)$ is strictly monotonic with respect to $P(x)$. Therefore it is equivalent to give a recommendation based on odds since we only care about the relative ranking of each end product configuration. Then we have

$$O(I=1|C,S)=\frac{P(I=1|C,S)}{P(I=0|C,S)} = \frac{P(I=1|S)}{P(I=0|S)}\cdot\frac{P(C|I=1,S)}{P(C|I=0,S)} \quad (3)$$

The first term in this expression contains no information about a product configuration. Thus it does not affect the result and can be omitted. Now the task is to decompose the second term in an appropriate form to facilitate the calculation of probability (odds) of relevance.

By chain rule in probability, the dependency in an uncertain domain can be represented as the product of conditional probabilities,

$$P(C|I,S)=P(a_1,\ldots a_m|I,S) = P(a_1|I,S)\cdot P(a_2|I,S,a_1)\ldots P(a_m|I,S,a_1,\ldots a_{m-1}) \quad (4)$$

However this is always computationally inefficient and not feasible. First, the data required to estimate the whole conditional probabilities, particularly the high order ones, is an exceptionally large amount. Secondly, low order dependency is usually more important to estimate the full joint probability (van Rijsbergen 1979). Thus a feasible way is to only consider the first order conditional probabilities which have already captured significant dependent information in most cases. Intuitively, we select the conditional variable which accounts for the most of the dependent relationship to a given variable. In a formal language,

$$P(a,\ldots,a_m|I,S)=\prod_{i=1}^{m}P(a_{i'}|I,S,a_{\pi(i')}),\quad 0\le\pi(i)<i \quad (5)$$

and $P(a_{0'}|a_{\pi(0')})=P(a_{0'})$ where $\{1',2',\ldots n'\}$ is a permutation of $\{1,2,\ldots n\}$ and $\pi(i')$ is a function mapping i to an integer less than i, i.e., $\pi(i')$ is the variable that affects i' most. In this paper we use WinMine, a toolkit developed by Microsoft (Chickering 2002), to find the first order dependency.

Considering the first order conditional dependency, we can get

$$P(C|I=1,S)=\prod_{i}P(a_{i'}|a_{\pi(i')},I=1,S) \quad (6)$$

Let

$$p_{i'}=P(a_{i'}=1|a_{\pi(i')}=1,I=1,S) \quad (7)$$
$$q_{i'}=P(a_{i'}=1|a_{\pi(i')}=0,I=1,S) \quad (8)$$
$$t_{i'}=P(a_{i'}=1|a_{\pi(i')}=1,I=0,S) \quad (9)$$
$$r_{i'}=P(a_{i'}=1|a_{\pi(i')}=0,I=0,S) \quad (10)$$

Then a general expression of $P(a_{i'}|a_{\pi(i')},I=1,S)$ can be stated as

$$P(a_{i'}|a_{\pi(i')},I=1,S)=\left[p_{i'}^{a_{i'}}(1-p_{i'})^{1-a_{i'}}\right]^{a_{\pi(i')}} \left[q_{i'}^{a_{i'}}(1-q_{i'})^{1-a_{i'}}\right]^{1-a_{\pi(i')}} \quad (11)$$

$$P(a_{i'}|a_{\pi(i')},I=0,S)=\left[t_{i'}^{a_{i'}}(1-t_{i'})^{1-a_{i'}}\right]^{a_{\pi(i')}} \left[r_{i'}^{a_{i'}}(1-r_{i'})^{1-a_{i'}}\right]^{1-a_{\pi(i')}} \quad (12)$$

Then we arrive at

$$\frac{P(C|I=1,S)}{P(C|I=0,S)} =\prod_{i'}\frac{\left[p_{i'}^{a_{i'}}(1-p_{i'})^{1-a_{i'}}\right]^{a_{\pi(i')}}\left[q_{i'}^{a_{i'}}(1-q_{i'})^{1-a_{i'}}\right]^{1-a_{\pi(i')}}}{\left[t_{i'}^{a_{i'}}(1-t_{i'})^{1-a_{i'}}\right]^{a_{\pi(i')}}\left[r_{i'}^{a_{i'}}(1-r_{i'})^{1-a_{i'}}\right]^{1-a_{\pi(i')}}} \quad (13)$$

Take logarithm on both sides of this equation, we can get the recommendation criterion under first order dependency, i.e.,

$$R=\sum_{i'}\left(a_{i'}\log\frac{q_{i'}(1-r_{i'})}{r_{i'}(1-q_{i'})}+a_{\pi(i')}\log\frac{(1-p_{i'})(1-r_{i'})}{(1-q_{i'})(1-t_{i'})} +a_{i'}a_{\pi(i')}\log\frac{p_{i'}(1-q_{i'})r_{i'}(1-t_{i'})}{q_{i'}(1-p_{i'})t_{i'}(1-r_{i'})}\right)+const \quad (14)$$

Parameter estimation

The meaning of p_i is the probability that a satisfactory product configuration consists of attribute i and its direct predecessor given the customer's partial specifications. Similarly, q_i means the probability that a satisfactory product configuration consists of attribute i but does not contain its direct predecessor given the customer's partial specifications. Based on the existing data, we can complete the following tables by filling in the count of corresponding specification-accepted recommendation pairs.

	$a_i=1$	$a_i=0$	Sum of row
$a_{\pi(i)}=1$	m	n	$m+n$
$a_{\pi(i)}=0$	k	l	$k+l$
Sum of col.	$m+k$	$n+l$	$m+n+k+l$

Thus we can get

$$p_{i'}=P(a_{i'}=1|a_{\pi(i')}=1,I=1,S)=\frac{m}{m+n} \quad (15)$$

and

$$p_{i'}=P(a_{i'}=1|a_{\pi(i')}=0,I=1,S)=\frac{k}{k+l} \quad (16)$$

In the same way, we can also fill in an identical table but using the remaining data which corresponds to $I=0$.

	$a_i = 1$	$a_i = 0$	Sum of row
$a_{\pi(i)} = 1$	m'	n'	$m' + n'$
$a_{\pi(i)}$=0	k'	l'	$k' + l'$
Sum of col.	$m' + k'$	$n' + l'$	$m' + n' + k' + l'$

$$t_{i'} = P(a_{i'} = 1 | a_{\pi(i')} = 1, I = 0, S) = \frac{m'}{m' + n'} \tag{17}$$

and

$$r_{i'} = P(a_{i'} = 1 | a_{\pi(i')} = 0, I = 0, S) = \frac{k'}{k' + l'} \tag{18}$$

It can be show that they are the maximum likelihood estimation of p_i, q_i, t_i and r_i. Smoothing corrections are needed to avoid 0 denominators. One of the frequently used smoothing methods takes the following form by adding a small number to both denominator and numerator, $p_{i'} = \frac{m+\varepsilon}{m+n+\varepsilon}$, $p_{i'} = \frac{k+\varepsilon}{k+l+\varepsilon}$, $t_{i'} = \frac{m'+\varepsilon}{m'+n'+\varepsilon}$, $r_{i'} = \frac{k'+\varepsilon}{k'+l'+\varepsilon}$.

It should be noted that this kind of data driven approach is consistent with the physical constraints in the traditional rule based expert system. Suppose the attribute value $A = a$ cannot exist together with $B = b$, then a well-defined set of data will not contain configuration data with "$A = a$" and "$B = b$". Based on the above mentioned estimation method, the conditional probability $P(A = a|B = b) = 0$ and $P(B = b|A = a) = 0$. Thus in the recommendations list, the contradicted product configurations will not appear. In this sense, the physical constraints are modeled as 0 or 1 conditional probabilities in our paper.

Complexity analysis

The calculation of the approach mainly comes from three parts, namely the identification of the first order dependency, the estimation of parameters and the calculation of recommendation criterion. The first two steps are offline parts and the last one is executed online. Chow et al. shown that finding the optimal first order dependency is equivalent to constructing the maximum spanning tree on the variables in corresponding uncertain domain (Chow and Liu 1968). The construction of maximum spanning tree can be reduced to the problem of constructing the minimal spanning tree by multiplying the edge weights by -1. There are different algorithms to build minimal spanning tree. Tarjan's randomized algorithm can find the MST in linear time with high probability (Karger et al. 1995). A more practical algorithm is proposed in (Katriel et al. 2003) with complexity *O(m+nlogn)* where n is the number of vertices and m is the number of edges. So this step can be done quite efficiently. As for the estimation of parameter, suppose there are m edges in the minimal spanning tree, then we need to estimate 4m conditional probabilities. If there are k specification-configuration data pairs, a brute force approach will have time complexity $O(m * k)$ to finish the computation. Thus the first two steps can be conducted with low time complexity. In addition, they usually don't demand on low time complexity since they can be carried out offline. The recommendations should be provided online and are of particularly interest for complexity analysis. The recommendation criterion (14) is a quadratic function of binary variable a for each product configuration. Given the conditional probabilities and the maximal spanning tree structure, this online part can also be conducted very fast. This guarantees the recommendations can be presented instantly.

Probability ranking principle

For a probabilistic recommendation method, an appropriate order to present the recommendation result to the user is to rank products by their estimated probabilities of relevance with respect to the information obtained (van Rijsbergen 1979). This is the basic idea of the probability ranking principle (PRP). It refers to the criterion that a retrieval system's response to each request is a ranking of the entities in the collection in the order of decreasing probability of relevance. It can be further proved that the probability ranking principle is optimal, in the sense that it minimizes the expected loss (Ripley 1996). In this session, we also adopt the probability ranking principle to present the final recommendation.

The probability ranking principle has some interesting properties for product customization practice. In this session, we try to show the validity of probability ranking principle from two aspects, namely the utility theory and the recommendation efficiency perspective.

Properties of probability ranking principle from utility theory point of view

From utility theory point of view, probability ranking principle can guarantee the most utility compared to other recommendation presenting approaches.

Proposition 1 *The probability ranking principle is optimal with respect to any non-decreasing utility function.*

Without loss of generality, we will consider two recommended approaches which operate on the same list of probability of relevance. Suppose m products are recommended. Let X_i be an indicator representing the number of products which will meet the customer's needs by approach i and $i = 1, 2$. Then the cumulative probability distribution of X_i is $F_{i,m}(x) = P(X_i \le x|m)$ given m products are recommended. Thus $F_{i,m}(x)$ is the probability that the total number of products which will meet a customer's needs is at most x when m products are recommended. If$F_{1,m}(x) \le$

$F_{2,m}(x)$, the distribution of a satisfactory product by applying approach 1 stochastically dominates approach 2 (Mas-Colel et al. 1995). The intuition behind this definition is that if approach 1 stochastically dominates approach 2, approach 1 is more likely to recommend more satisfactory products. We can assume that a customer's utility function is non-decreasing with respect to the portion of satisfactory recommendations. In this way, stochastic dominance provides a way to evaluate the recommendation methods with respect to their expected utilities.

To prove Proposition 1, we apply some results from Gordon et al. (1991) which are revised in the context of product recommendation.

Lemma 1 *(Gordon and Peter 1991): Suppose approach 1 proposes n recommendations in a sequence* $S_1 = (r_{11}, r_{12}, \ldots r_{1n})$*. Each recommendation* r_{1i} *has probability* P_{1i} *to meet the needs of the customer. The sequence is arranged such that* $P_{11} \geq P_{12} \geq \cdots \geq P_{1n}$*. Approach 2 also proposes n recommendations in a sequence* $S_2 = (r_{21}, r_{22}, \ldots r_{2n})$*. These n recommendations may be different from the ones in sequence* S_1*. Similarly, we also have corresponding probability serial* $\{P_{2i} : 1 \leq i \leq n\}$ *and* $P_{21} \geq P_{22} \geq \cdots \geq P_{2n}$*. If* $P_{1i} \geq P_{2i}$ *for all* $1 \leq i \leq n$*, then* X_1 *stochastically dominates* X_2 *where* X_i *is an indicator of the number of satisfactory recommendations by using approach i.*

Lemma 2 *Suppose approach 1 proposes n recommendations in a sequence* $S_1 = (r_{11}, r_{12}, \ldots r_{1n})$*. Each recommendation* r_{1i} *has probability* P_{1i} *to meet the needs of a customer. The sequence is arranged such that* $P_{11} \geq P_{12} \geq \cdots \geq P_{1n}$*. Approach 2 also proposes n recommendations in a sequence* $S_2 = (r_{21}, r_{22}, \ldots r_{2n})$ *which is a permutation of* $S_1 = (r_{11}, r_{12}, \ldots r_{1n})$*. Then the distribution of satisfactory product for approach 1 is identical to approach 2.*

Lemma 3 *(Gordon and Peter 1991): Let* $U(x)$ *be a non-decreasing utility function where x is the number of satisfactory recommendations. Let* X_i *be an indicator of the number of satisfactory recommendations by using approach i. If* X_1 *stochastically dominates* X_2*, then the expected utility by adopting approach 1 is greater or equal to that of approach 2, i.e.,* $E[U(X_1)] \geq E[U(X_2)]$.

Proof of Proposition 1 Without the loss of generality, suppose there are n end products in total. Probability ranking principal based approach (Approach 1) gives n recommendations in a sequence$S_1 = (r_{11}, r_{12}, \ldots r_{1n})$. Each recommendation r_{1i} has probability P_{1i} to meet the needs of the customer. The sequence is arranged such that $P_{11} \geq P_{12} \geq \cdots \geq P_{1n}$. Approach 2 also gives n recommendations in a sequence $S_2 = (r_{21}, r_{22}, \ldots r_{2n})$. Suppose a customer is willing to screen m recommendations to find his/her desired product, $1 \leq m \leq n$. Thus we only need to consider the utilities of two sub series $S_1^* = (r_{11}, r_{12}, \ldots r_{1m})$ and $S_2^* = (r_{21}, r_{22}, \ldots r_{2m})$ where $P_{11} \geq P_{12} \geq \cdots \geq P_{1m}$.

According to Lemma 2, rearranging the order of the recommendations will not change the probability distribution of the numbers of satisfactory products. We can sort S_2^* in descending order, $(r_{21'}, r_{22'}, \ldots r_{2m'})$ such that $P_{21'} \geq P_{22'} \geq \cdots \geq P_{2m'}$. Since $r_{2i'}$ corresponds to the ith biggest probability of relevance in a subset of the whole products set and r_{2i} corresponds to the ith biggest probability of relevance in the whole products set, we can get $P_{1i} \geq P_{2i'}$ for all i.

From Lemma 1, approach 1(PRP) stochastically dominates approach 2. From Lemma 3, the expected utility by adopting approach 1 is greater or equal to that of approach 2. □

Properties of probability ranking principle from recommendation accuracy point of view

In this session, some properties of the probability ranking principle are identified from a recommendation efficiency and accuracy point of view. Wang and Tseng already showed that probability ranking principle based approach can lead to the minimal expected search length (Wang and Tseng 2011). Precision and recall rate are used to measure the accuracy of the method.

In the study information retrieval, precision and recall rate are the most widely used metric to measure system performance (van Rijsbergen 1979). Precision rate is the proportion of relevant items with respect to all the retrieved items. It means the probability that a retrieved item is relevant to the user's interest. Thus it is a measure of exactness. Recall rate is used to measure completeness. It is defined as the number of relevant items retrieved by a system divided by the total number of existing relevant items which should have been retrieved. It stands for the probability that a relevant item is retrieved.

In the probability ranking principle, all the potential products are ranked in descending order based on their probabilities of relevance. The expected precision can be defined as follows, $P = \frac{\sum_{i=1}^{n} p_i}{n}$ where n is the number of recommended products. And expected recall is $Q = \frac{\sum_{i=1}^{n} p_i}{\min(N,n)}$ where N is the total number of end products which will meet the customer's requirements.

Proposition 2 *The recommendation based on probability ranking can guarantee the highest precision and recall rate.*

Proof Suppose there are N products and they are ranked by probability of relevance. Each product is with probability P_i to be a customer's desired one and $P_1 \geq P_2 \geq \cdots \geq P_N$.

Although all the end products are recommended in the order decided by probability of relevance, a customer may not screen all of them. Suppose a customer will screen n product with probability $\Pr(n)$.

Thus the precision and recall rate are functions of n. Then $P_{PR}(n) = \frac{\sum_{i=1}^{n} P_i}{n}$ and $Q_{PR}(n) = \frac{\sum_{i=1}^{n} P_i}{n'}$ are the precision and recall by applying probability ranking where $n' = \min(n, N')$ and N' is the total number of satisfactory products. The precision and recall rate of other recommendation methods are $P_{other}(n) = \frac{\sum_{i=1}^{n} P'_i}{n}$ and $Q_{other}(n) = \frac{\sum_{i=1}^{n} P'_i}{n'}$ where P'_i is the probability that the ith recommendation is satisfactory by applying other recommendation method. It is easy to see that $\{P'_1, P'_2, \ldots, P'_N\}$ is a permutation of $\{P_1, P_2, \ldots, P_N\}$.

Without loss of generality, suppose that P'_is are assumed to be ranked in descending order. Then we have $P_i \geq P'_i$. It is obvious that $P_{PR}(n) = \frac{\sum_{i=1}^{n} P_i}{n} \geq \frac{\sum_{i=1}^{n} P'_i}{n} = P_{other}(n)$ and $Q_{PR}(n) = \frac{\sum_{i=1}^{n} P_i}{n'} \geq \frac{\sum_{i=1}^{n} P'_i}{n'} = Q_{other}(n)$.

Thus the expected precision and recall with respect to the number of screened product is $\sum_{n=1}^{N} \Pr(n) \cdot P_{PR}(n) \geq \sum_{n=1}^{N} \Pr(n) \cdot P_{other}(n)$ and $\sum_{n=1}^{N} \Pr(n) \cdot Q_{PR}(n) \geq \sum_{n=1}^{N} \Pr(n) \cdot Q_{other}(n)$.

The equation holds only when $n = N$. □

Numerical example

In this part, a PC recommender is used to exemplify and test the viability and performance of this approach. In the previous session, we have shown that the probability ranking principle outperforms other methods in terms of expected utility and efficiency. This part will demonstrate the merit of the probabilistic relevance model by using the example of a PC recommender. We will show the combination of probability relevance model under first order dependency assumption and probability ranking principle performs better than two other recommendation approaches, namely random recommendation and probability relevance model under independency assumption + probability ranking principle.

A PC is considered as a combination of six components, i.e., a processor, monitor, hard disk, display card, memory and display driver. The set of components and their alternatives are listed in Table 1. The recommendation process forms an iteration loop as shown in Fig. 1. Each time a new customer gives a specification to one component, a recommendation will be presented accordingly. If the customer is satisfied with it, he/she can terminate the process. Otherwise the customer can refine the recommendation by giving more specifications. In the worst case scenario, all the six components are needed to be specified which means the customer is not satisfied with the recommendations in each round. In this experiment, we use the number of communication rounds, i.e., the number of specifications needed to be inputted as the metric of recommendation's efficiency. The fewer rounds occur, the more efficient the recommendation method is.

Table 1 List of components and their alternatives for a PC

Component	Code	Description
Processor (A)	A1	Pentium E2160 1.8G
	A2	Pentium E2180 2.0G
	A3	AMD Athlon™ 64 X2
	A4	Intel Core 2 Duo E4300 1.8G
	A5	Intel Core 2 Duo E4500 2.2G
	A6	Intel Core 2 Duo E4600 2.4G
Memory (B)	B1	512 MB DDR2
	B2	1 GB × DDR2
	B3	2 GB × DDR2 dual channel
Monitor (C)	C1	17′ LCD
	C2	19′ LCD
	C3	20′ LCD and above
Hard disk (D)	D1	80 GB
	D2	160 GB
	D3	250 GB
Disk driver (E)	E1	24X CD-RW/DVD* Combo
	E2	48X CD-RW/DVD* Combo
	E3	8X DVD+/−RW*
	E4	16X DVD+/−RW*
Display card (F)	F1	NVIDIA® GeForce® 6150
	F2	Intel® Graphics Media Accelerator 3000
	F3	Intel® Graphics Media Accelerator X3000
	F4	128 MB PCIe™ × 16 ATI Radeon™ X1300
	F5	256 MB PCIe™ × 16 nVidia® GeForce® 7300 LE TurboCache

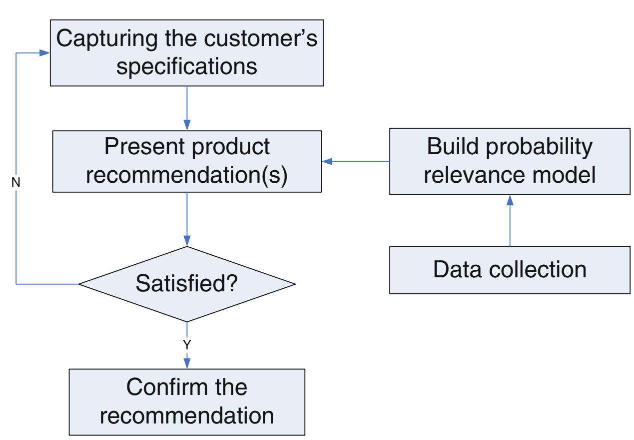

Fig. 1 The schematic framework of recommendation

We collected 69 sets of customer specification data and the corresponding accepted recommendation data. However, this is not enough to estimate the required parameters. To deal with the issue of data sparsity, we apply perturbative bootstrap to generate 1,380 specification-recommendation data pairs as the training data to learn the parameters (Wang and Tseng 2011). 207 specification-recommendation data pairs are also generated as the testing data. The details of perturbative bootstrap can be found as follows;

Suppose the attributes set of a product's is $\{A_i : 1 \le i \le k\}$ where k is the number of attributes. Each attribute A_i has a corresponding alternatives set $\{a_{ij} : 1 \le i \le k, 1 \le j \le n_i\}$where n_i is the number of alternatives for attribute A_i. Each attribute alternative a_{ij} has a substitute set $\overline{a_{ij}}$.

For $l = 1$ to m //m is the data size of the original data set
  For i = 1: k
    For $j = 1 : n_i$
    u = rand(0,1); // generating a random number following uniform(0,1)
    if $u > h$ and $\overline{a_{ij}}$ is not empty, a^*_{ij} equals to one value in $\overline{a_{ij}}$ with probability $\frac{1}{|\overline{a_{ij}}|}$ where $|\overline{a_{ij}}|$ is the cardinal of set $\overline{a_{ij}}$ and h is a predetermined threshold;
    else $a^*_{ij} = a_{ij}$;
    end
  end
end

By this method, sufficient training and testing data can be generated. It should be noted that the threshold h can also indicate a customer's preference flexibility. For example, we have generated a virtual customer's specification data D1' based on an existing customer's specification data D1. Now we want to use the customer's accepted configuration data S1 to generate S1', i.e., the virtual customer's acceptable configuration. If a bigger h is set, it means S1' and S1 are more diversified. Then the difference between D1' and S1' will be larger than that between D1 and S1, indicating the virtual new customer is more flexible to the recommendations. Since we also try to test the effect of the flexibility of customer preference on the communication rounds, we set h from 0.05 to 0.95 with step being 0.1 and generate 10 groups of testing data.

Three recommendation approaches are considered in this paper. The first one is random recommendation, meaning that each time a customer gives a specification to one component, a random configuration which is consistent with the specification will be proposed. The second recommendation is based on a probabilistic relevance model under independent assumption, meaning that there is no dependency among the attributes. Under this assumption, $p_{i'}, q_{i'}$ in Eq. (2) will degenerate to a common parameter p_i and so are $t_{i'}, r_{i'}$. Suppose they degenerate to t_i. Then the ranking criteria corresponding to (2) becomes $R = \sum_i \log \frac{p_i(1-q_i)}{q_i(1-p_i)} \cdot a_i$. The third one is probabilistic relevance model under first order dependency assumption mentioned in this paper. Figure 2 illustrates the curve of the number of rounds versus a customer's flexibility if only one product is recommended in each round. X axis is h value used to generate accepted configuration data which "*controls*" customer flexibility. The worst case requires 6 rounds of communication. The random recommendation will need about 5.6 rounds of recommendation on average. The probabilistic relevance model under first order dependency assumption performs better than recommendation under independent assumption. It can also be anticipated that with the increasing of customer's flexibility, the recommendation are more likely to meet the customer's requirements since the curves tend to decrease with h.

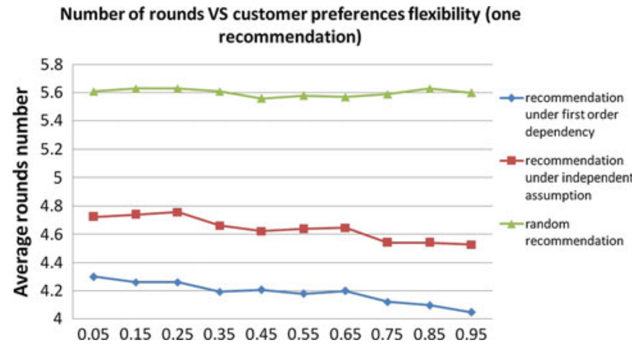

Fig. 2 The comparison between different recommendation approaches give one recommendation

Similarly, we can obtain the results of communication rounds versus customer preferences under different numbers of recommendations. The detailed results are shown in Fig. 3. Similar results can be achieved, showing that the probabilistic relevance model under the first order dependency assumption is better than probabilistic relevance model under an independency assumption and both decrease with respect to h.

Concluding remark

In this paper, we adopt a new approach to address personalized recommendation for customized products. Basically two questions are answered in this paper, namely which end products should be recommended and in which order to present the recommendations given a customer's specification for a product. Because a customer's specification is often incomplete and ambiguous, traditional methods cannot represent and manipulate the partial information properly to give appropriate recommendation. Occasionally the specifications contradict each other. The inconsistency fur-

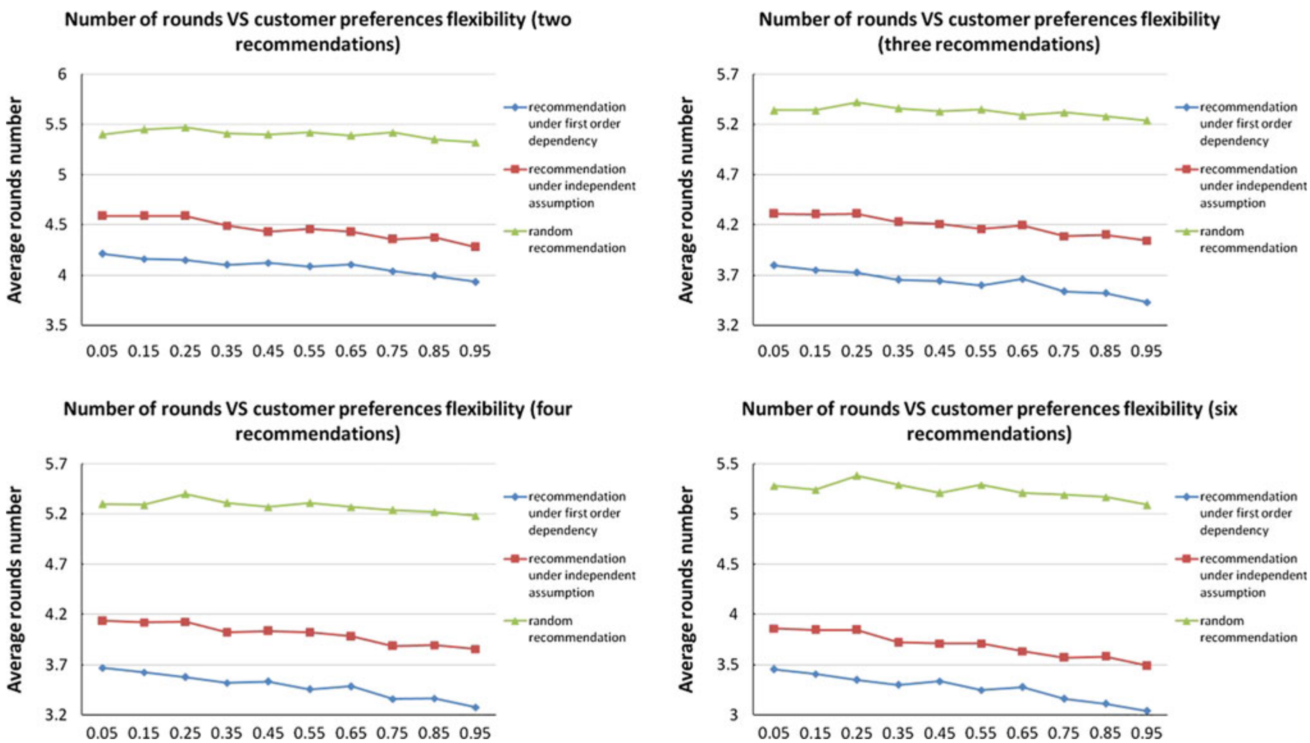

Fig. 3 The comparison between different recommendation approaches under different numbers of recommendations

ther makes the recommendation task complicated because in this case, dead ends will occur quite frequently. In this paper, a probability relevance model is adopted to tackle this particularly complex issue. The idea is to calculate the likelihood or probability that a product will meet an active customer's specification based on the incomplete or even distorted information. The ranking of a potential product based on the probability of relevance is then provided as the recommendation. This method also incorporates the flexibility of customers' choices to different attributes because even a customer who only specifies one attribute may find other attributes acceptable. This paper investigates existing configuration data, i.e., the "specification - final choice" data pair, to discover previous customers' preference pattern in the form of conditional probability. Then the knowledge is deployed to provide recommendations for new customers, i.e., using "social" preference to recommend potential likely end product. A probability ranking principle is also exploited to present the recommendations. Analytical results show that the probability ranking principle is optimal with respect to any non-decreasing utility function and expected search length. In this way, the recommendation approach can overcome the challenges of *new customer issue*, *inconsistent specification issue* by making use of *customer flexibility*.

In this paper, we assume that there is no correlation between different recommendations. In the next step, we are trying to take the correlations into consideration. One intuitive extension to the recommendation method is to present the most relevant product and at the meantime minimize the redundancy of the whole recommendation list. In this way, we anticipate the variance of search length will be minimized while maintaining a sufficient small expected search length.

Acknowledgments This research is supported by Hong Kong Research Grants Council (GRF HKUST 620308 and 620609).

References

Aldanondo, M., & Vareilles, E. (2008). Configuration for mass customization: How to extend product configuration towards requirements and process configuration *19*(5), 521–535.

Basu, C., Hirsh, H., & Cohen, W. (1998). Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the 1998 workshop on recommender systems* (pp. 11–15). AAAI Press.

Billsus, D., & Pazzani, M. J. (1998). Learning collaborative information filters. In *Proceedings of the 15th international conference on machine learning* (pp. 46–53).

Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th conference on uncertainty in artificial intelligence*.

Chickering, D. M. (2002). *The WinMine toolkit*. Microsoft Research Technical Report MSR-TR-2002-103.

Chow, C. K., & Liu, C. N. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory, 14*(3), 462–467.

Cox, W. M., & Alm, R. (1998). *The right stuff: America's move to mass customization*. 1998 annual report of Federal Reserve Bank of Dallas, Dallas, TX.

Crestani, F., Lalmas, M., van Rijsbergen, C. J., & Campbell, I. (1998). Is this document relevant?…probably: A survey of probabilistic models in information retrieval. *ACM Computing Surveys, 30*(4), 528–552.

Croft, W., & Harper, D. (1979). Using probabilistic models of information retrieval without relevance information. *Journal of Documentation, 35*, 285–295.

Gordon , M. D., & Peter, L. (1991). A utility theory examination of the probability ranking principle in information retrieval. *Journal of the American Society for Information Science, 42*(10), 703–714.

Harper, D. J., & van Rijsbergen, C. J. (1978). An evaluation of feedback in document retrieval using co-occurrence data. *Journal of Documentation, 34*(1), 189–216.

Karger, D. R., Klein, P. N., & Tarjan, R. E. (1995). A randomized linear-time algorithm to find minimum spanning trees. *Journal of the Association for Computing Machinery, 42*(2), 321–328. doi:10.1145/201019.201022 (MR1409738).

Katriel I., Sanders, P., Larsson T., Jesper, T., & Jesper, L. (2003). A practical minimum spanning tree algorithm using the cycle property. In *11th European symposium on algorithms (ESA)* (pp. 679–690). LNCS number 2832.

MasColel, A., Whinston, M., & Green, J. (1995). *Microeconomic theory*. Oxford: Oxford University Press.

Mavridou, E., Kehagias, D. D, Tzovaras, D., & Hassapis, G. (2011). Mining affective needs of automotive industry customers for building a mass-customization recommender system. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-011-0579-4 (accepted).

Moon, S. K., Simpson, T. W., & Kumara, S. R. T. (2009). An agent-based recommender system for developing customized families of products. *Journal of Intelligent Manufacturing, 20*(6), 649–659.

Peddy, C. C., & Armentrout, D. (2003). *Building solutions with microsoft commerce server 2002*. Redmond, WA: Microsoft Press.

Resnick, P., Iacovou, N., Sushak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: An open architecture for collaborative filtering of Netnews. In *Proceedings of the 1994 computer supported collaborative work conference*.

Ripley, B. D. (1996). *Pattern recognition and neural networks*. Cambridge: Cambridge University Press.

Risdiyono, & Koomsap, P. (2011). Design by customer: Concept and applications. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-011-0587-4 (accepted).

Shardanand, U., & Maes, P. (1995) Social information filtering algorithms for automating 'Word of Mouth'. In *Proceedings of CHI'95*.

Simon, H. H. (1957). *Models of man*. New York: Wiley.

Stormer, H. (2009). Improving product configurators by means of a collaborative recommender system. *International Journal of Mass Customisation, 3*(2), 165–178.

van Rijsbergen, C. J. (1979). *Information retrieval*. London: Butterworths.

Wang, Y., & Tseng, M. M. (2009). Recommendation for custom product via probabilistic relevance model. In *Proceedings of IEEE conference on engineering management*, Hong Kong.

Wang, Y., & Tseng, M. M. (2011). Adaptive attribute selection for configurator design via Shapley value. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 25*(1), 185–195.

Wright, P. (1975). Consumer choice strategies: Simplifying vs. optimizing. *Journal of Marketing Research, 12*(February), 60–67.

Zhang, T., & Iyengar, V. S. (2002). Recommender systems using linear classifiers. *Journal of Machine Learning Research, 2*(1), 313–334.

Zhu, X., Hu, S. J., & Yoram, K. (2008). Modeling of manufacturing complexity in mixed-model assembly lines. *Journal of Manufacturing Science and Engineering, 130*(5).