Electron Commer Res (2015) 15:427–452
DOI 10.1007/s10660-015-9191-6

Predicting customer purchase behavior in the e-commerce context

Jiangtao Qiu[1] · Zhangxi Lin[2] · Yinghong Li[3]

Published online: 30 June 2015

Abstract Predicting customer purchase behavior is an interesting and challenging task. In the e-commerce context, meeting this challenge requires confronting many problems not observed in the traditional business context. Recommender system technology has been widely adopted by e-commerce websites. However, a traditional recommendation algorithm cannot perform well the predictive task in this context. This study intends to build a predictive framework for customer purchase behavior in the e-commerce context. This framework, known as *C*ust*O*mer purchase p*RE*diction mode*L* (COREL), may be understood as a two-stage process. First, associations among products are investigated and exploited to predicate customer's motivations, i.e., to build a candidate product collection. Next, customer preferences for product features are learned and subsequently used to identify the candidate products most likely to be purchased. This study investigates three categories of product features and develops methods to detect customer preferences for each of these three categories. When a product purchased by a particular consumer is submitted to COREL, the program can return the top n products most likely to be purchased by that customer in the future. Experiments conducted on a real dataset show that customer preference for particular product features plays a key role in decision-making and that COREL greatly outperforms the baseline methods.

✉ Jiangtao Qiu
jiangtaoqiu@gmail.com

Zhangxi Lin
zhangxi.lin@ttu.edu

Yinghong Li
Lily.yinghong@gmail.com

[1] School of Information, Southwestern University of Finance and Economics, Chengdu, China

[2] The Rawls College of Business Administration, Texas Tech University, Lubbock, TX, USA

[3] School of Humanities, Southwestern University of Finance and Economics, Chengdu, China

Keywords Predictive model · Purchase behavior · E-commerce

1 Introduction

A firm that can predict customer purchase behavior will achieve numerous benefits, including an improved customer acquisition rate, increased sales, and enhanced competitiveness. Research efforts from the marketing and customer relationship management (CRM) perspectives have made major strides in predicting customer purchase behavior. For example, market basket analysis [29, 31] examines groups of items purchased in supermarkets or other outlets to identify purchase patterns, and discrete choice models [16, 30] predict which products a specific customer is likely to select from a candidate product set. Certain studies abstract the prediction of purchase behavior into a classification task [7, 10] that requires customers' demographic features, such as age, gender, education and occupation.

Unlike traditional businesses, firms in the e-commerce context find it difficult to obtain information about customer demography or family background because these data are usually regarded as private [19]. Rather, it is more convenient to access customer reviews, product ratings and visiting tracks. Therefore, the methods and algorithms used to predict customer purchase behavior in the traditional business context must be modified for e-commerce. To meet the challenge of predicting purchase behavior in the e-commerce context, we first examine the purchasing decision process of customers. Guo and Barnes [8] propose a three-stage purchasing decision process in the e-commerce context, which is illustrated in Fig. 1.

The first stage, problem/motivations[1] recognition, captures consumers' perceptions of how products may help them to bridge the gap between the desired and actual states. During the second stage, a customer must seek information about product performance or other criteria and must evaluate product alternatives based on price, brand and other attributes.

Recommender system technology has been widely adopted by e-commerce websites. This technology not only provides appropriate recommendations to users but also yields substantial profits for the service provider. Nevertheless, we believe that traditional recommendation algorithms cannot effectively predict purchasing behavior in the e-commerce context, for three reasons:

(1) Recommender systems [12, 17, 32, 33] predict the items most likely to interest a customer either by exploiting product ratings by other customers with similar tastes [collaborative filtering (CF)] or by using past product ratings by the target customer (content-based recommendations). However, the rating predicted by a recommender system for a candidate product only indicates the impression the customer will likely have of that product, i.e., it predicts a "like" signal. This is a far cry from the goal of predicting purchase behavior. According to our experiments, the use of CF alone to predict

[1] In this paper, we use the term motivation as shorthand for motivations for purchasing products.

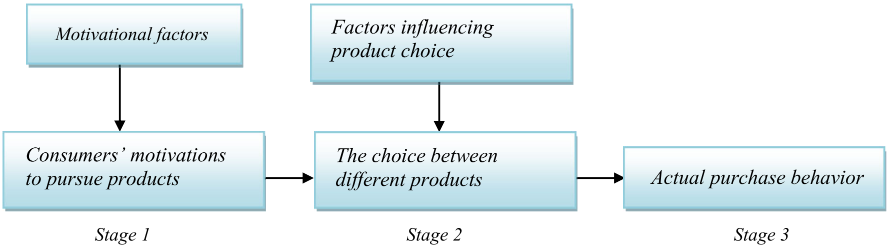

Motivational factors
Factors influencing product choice
Consumers' motivations to pursue products
The choice between different products
Actual purchase behavior
Stage 1
Stage 2
Stage 3

Fig. 1 Consumer behavior and the purchasing decision in the e-commerce context

customer purchase behavior leads to a poor performance. Reference [28] notes that a consumer usually makes purchasing decisions based on a product's marginal net utility. A rational consumer chooses to purchase the product that maximizes total net utility.

(2) Many works on recommender systems focus on the user–item matrix and employ tensor factorization. However, those studies only consider associations among users and items and ignore rich product features.[2] We think that side information plays an important role in the second stage of the purchasing decision process depicted in Fig. 1.

(3) Recommender systems usually do not address the first stage of the purchasing decision process shown in Fig. 1. Rather, recommender systems address only a single component of the second stage of the purchasing decision process. In addition, some recommender systems aim to increase product awareness [4] and thus consider freshness and novelty in product recommendations [18]. Such systems cannot be employed to perform the predictive task.

In this study, we propose a two-stage predictive framework. First, customer's motivations are investigated. However, not all motivations that affect purchases of real products can be investigated in the e-commerce; one such example is Maslow's psychological and safety needs. We exploit the associations among products to predict a customer's motivations. The second step of the framework addresses the numerous factors that affect customer decision making, such as price [25], brand preference, economic needs [23], etc. We seek to integrate side information about products into the predictive task by learning customers' preferences for particular product features. Furthermore, we leverage these preferences to select a collection of candidate products based on customer's motivations. When a customer submits one purchased product into the predictive framework, this framework can return the top *n* products most likely to be purchased by that customer in the future.

The remainder of this paper is organized as follows. Section 2 reviews the related research on predicting customer purchase behavior. Section 3 introduces the proposed predictive framework. Section 4 presents the experimental results.

[2] In this paper, we use the term "product feature" to refer both to product attributes (e.g., color, size, etc.) and to the related information displayed on e-commerce websites (e.g., ratings, reviews, sales, etc.).

Section 5 contains the analysis of customer characteristics in the experimental dataset. Section 6 presents the conclusions.

2 Related work

The three principal research streams are related to predicting customer purchase behavior. Researchers in the marketing and retailing fields have expended significant effort to predict customer purchase behavior, which can help firms to identify likely purchasers of their products or services and to implement cross-selling and up-selling campaigns. The association rule technique developed for market basket analysis has become a popular method for identifying customers' purchase patterns by extracting associations, or co-occurrences, from stores' transaction databases. These purchase patterns can then be exploited to predict customers' future purchase behavior. Chi-Wing Wong et al. [29] conceive the *loss rule*, which is similar to the association rule, to model the cross-selling effect. Mobasher et al. [15] use frequent item mining techniques to discover sequential patterns that are used to generate recommendations. Yang et al. [31] define online shopping patterns and develop methods to perform market basket analysis across websites. Discrete choice models may be employed to analyze customer preferences, which play an important role in purchasing decisions. Yang and Allenby [30] introduce a Bayesian autoregressive discrete choice model to study preference interdependence among individual consumers. Moon and Russell [16] develop a product recommender system based on the autologistic choice model. However, we believe that using association rules to identify purchase patterns only finish first step of the purchasing decision illustrated in Fig. 1.

The CRM system of a firm generally maintains a large quantity of customer data, including age, gender, income and purchase lists. Data mining techniques are often employed in analytical CRM to transform this large volume of data into valuable knowledge that may be used to support marketing decision making. Based on such data mining techniques, customers can be segmented into clusters with internally homogenous and mutually heterogeneous characteristics [9]. Customers can also be ranked on their probability to behave a certain way (e.g., to buy a specific product or to respond to a particular marketing campaign). With the help of segmentation and ranking schemes, a firm may approach a carefully selected group of customers, which leads to higher success rates for its marketing campaigns [26]. In general, it is difficult to acquire customer demographic information such as income, age and gender in the e-commerce context because such information is usually regarded as private [19]. Instead, it is easier to acquire web access information such as product reviews and ratings, which provide richer information than traditional CRM. Therefore, CRM prediction techniques based on private customer information do not transfer well to the e-commerce domain.

E-commerce firms generally use recommender systems to predict customer purchase behavior. Recommender systems based on a Markov chain model utilize sequential basket data by predicting the user's next action based on his/her last action. Zimdars et al. [33] describe a sequential recommender based on Markov chains and investigate how to extract sequential patterns to learn the next state using a standard predictor [e.g., a decision tree (DT)]. Mobasher et al. [15] use pattern mining methods to discover sequential patterns that are used to generate recommendations. Nevertheless, this category of recommender system focuses only on associations among products and is not suitable for predicting purchase behavior.

Most recommender systems recommend products based on the user's entire history. Extensive effort has been expended to develop factorization-based CF approaches. Reference [12] uses customer's purchase and click history to build customer's profile and then incorporate the profile into a one-class CF model. The factorizing results of the model are used to make product recommendation. Ulrich and Koenigstein [17] present a novel Bayesian generative model for implicit CF. Specifically, they incorporate random graphs, which can be leveraged to predict the presence of edges in a CF model, into an inference procedure. As a result, their model is able to explicitly extract a "like" probability that is largely agnostic to product popularity. Rendle et al. [22] develop a factorized personalized Markov chain (FPMC) model that subsumes both a common Markov chain and the normal matrix factorization model. Experiments show that the FPMC model outperforms both common matrix factorization and the unpersonalized Markov chain model.

Content-based filtering emphasizes a focal customer's preferences and can recommend products that have received few or even no ratings from other customers [20]. Content-based filtering builds a preference profile for each customer, uses it to search prospective products with attributes or characteristics that are highly relevant and similar to those specified by the focal customer's profile, and makes product recommendations accordingly [3]. Content-based filtering can be supported by traditional inductive learning techniques that construct automated classifiers based on important patterns in customer preference profiles.

However, traditional recommendation algorithms often recommend products with the highest predicted ratings. This is far cry from the goal of predicting purchase behavior. Reference [6] maintains that the price factor, which is represented as a quantity, can significantly improve the recommendation performance. Reference [28] notes that a rational consumer chooses to purchase the product with the maximum total net utility. Traditional recommendation algorithms ignore the significant impact of product features on customer purchasing decisions. In this study, we seek to incorporate customers' preference for product features into predictions of purchasing behavior.

3 Predicting customer purchase behavior

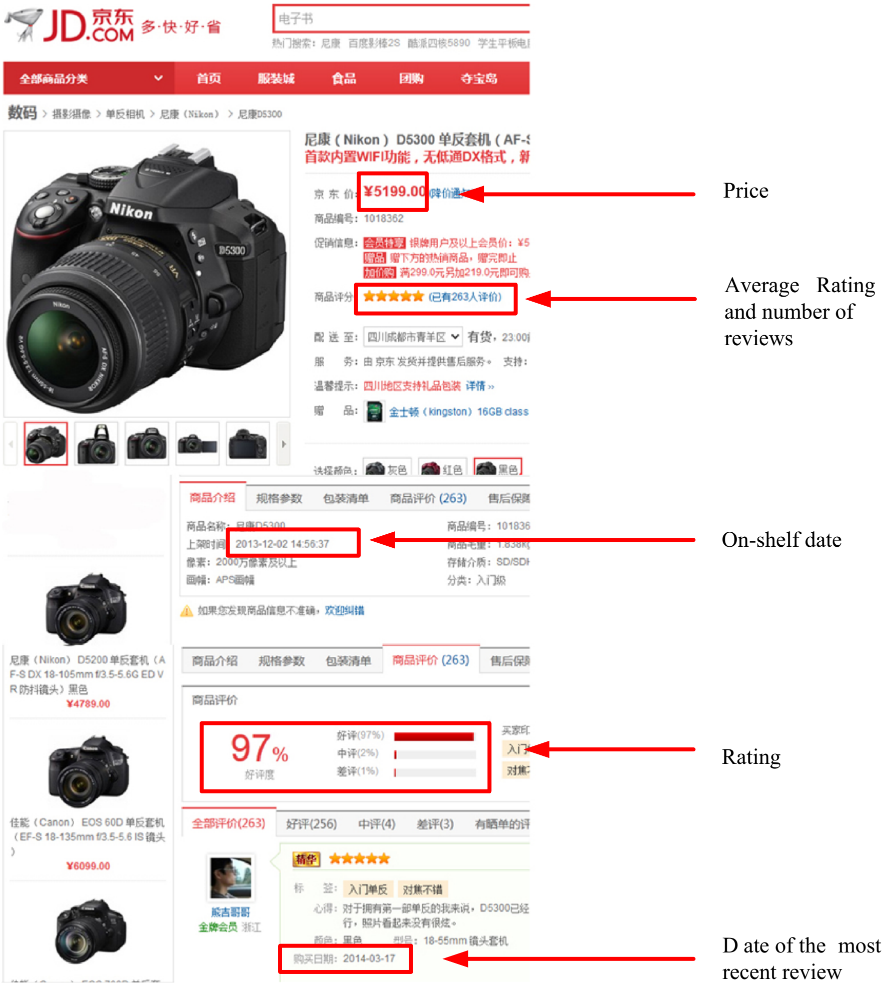
This section proposes a framework to predict customers' purchase behavior.

3.1 Motivation of the framework

As an initial step, we investigate the decision-making process in online shopping. Let c_k be a customer and both d_i and d_j be two products. We use $P(d_i)$ to denote the probability that product d_i is purchased and use P(d_j|d_i) to indicate the conditional probability that a user purchasing item d_i will purchase item d_j as well [11]. Suppose that customer c_k is likely to purchase d_i with the probability P(d_i|c_k) [4] and P(c_k) is a prior probability with a uniform distribution. We can make sure that P(c_k, d_j) = P(d_j|c_k)P(c_k) also indicates the probability that c_k purchase product d_j. Furthermore, P(c_k, d_j|c_k, d_i) = P(d_j|c_k, d_i) denotes a probability that after c_k purchased d_i in time t, c_k will also purchase d_j later.

$$P\left(d_j|c_k,\ d_i\right) = \frac{P(d_i|c_k,\ d_j)P(d_j,\ c_k)}{P(d_i,\ c_k)}.$$

If the event c_k purchasing a product is random, i.e., uncorrelated to purchasing d_i, the probability will be


JD.COM 京东 多·快·好·省

电子书

热门搜索：尼康 百度影棒2S 酷派四核5890 学生平板电

全部商品分类 首页 服装城 食品 团购 夺宝岛

数码 > 摄影摄像 > 单反相机 > 尼康（Nikon） > 尼康D5300

尼康（Nikon）D5300 单反套机（AF-S

首款内置WIFI功能，无低通DX格式，新

京 东 价：¥5199.00 降价通知

商品编号：1018362

促销信息：会员特享 银牌用户及以上会员价：¥5

赠品 赠下方的热销商品，赠完即止

加价购 满299.0元另加219.0元即可购

商品评分：★★★★★ (已有263人评价)

配 送 至：四川成都市青羊区 有货，23:00

服　　务：由 京东 发货并提供售后服务。 支持：

温馨提示：四川地区支持礼品包装 详情 »

赠　　品：金士顿（kingston）16GB class

选择颜色：灰色 红色 黑色

商品介绍 规格参数 包装清单 商品评价 (263) 售后保障

商品名称：尼康D5300 商品编号：101836

上架时间：2013-12-02 14:56:37 商品毛重：1.838kg

像素：2000万像素及以上 存储介质：SD/SDH

画幅：APS画幅 分类：入门级

如果您发现商品信息不准确，欢迎纠错

尼康（Nikon）D5200 单反套机（AF-S DX 18-105mm f/3.5-5.6G ED VR 防抖镜头）黑色 ¥4789.00

佳能（Canon）EOS 60D 单反套机（EF-S 18-135mm f/3.5-5.6 IS 镜头） ¥6099.00

商品介绍 规格参数 包装清单 商品评价 (263) 售后保障

商品评价

97% 好评度

好评(97%)

中评(2%)

差评(1%)

买家印象 入门 对焦

全部评价(263) 好评(256) 中评(4) 差评(3) 有晒单的评

熊吉哥哥 金牌会员 浙江

精华 ★★★★★

标　签：入门单反 对焦不错

心得：对于拥有第一部单反的我来说，D5300已经行，照片看起来没有很炫。

颜色：黑色 型号：18-55mm 镜头套机

购买日期：2014-03-17

Price
Average Rating and number of reviews
On-shelf date
Rating
D ate of the most recent review

Fig. 2 Product information presented on a Chinese e-commerce website

$$P\left(d_j|c_k,\ d_i\right) = \frac{P(d_j|d_i)P(d_j|c_k)}{P(d_j)}.$$

Let $\omega = \{d_1,\ldots,\ d_{\mathrm{i}-1},\ d_{\mathrm{i}+1},\ldots,\ d_{\mathrm{m}}\}$ be a collection of candidate products. We calculate $P(d_{\mathrm{j}}|c_{\mathrm{k}},\ d_{\mathrm{i}})$ for each product $d_{\mathrm{j}} \in \omega$ and then rank them. Inspired by the query likelihood model [14, 21], where the prior probability of a document $P(doc)$ is often treated as uniform across all documents to reduce model complexity, we also assume that the prior probability of a product $P(d)$ is uniform across all products. Therefore, $P(d_{\mathrm{j}})$ can be ignored for the purpose of rank. A predictive framework may be specified as

$$P\left(d_j|c_k,\ d_i\right) \propto P\left(d_j|d_i\right)P\left(d_j|c_k\right).$$

Parameter $P(d_{\mathrm{j}}|c_{\mathrm{k}})$ may also be interpreted as c_{k}'s preference for d_{j}. Each product contains many features and a customer may have a preference for these features. We make the conditional independence assumption, that is, for each customer c_{k}, his/her preferences for the features of product d_{j}, $\{f_{j1},\ldots,f_{j\mathrm{n}}\}$, are independent of each other. Thus, $P(d_{\mathrm{j}}|c_{\mathrm{k}})$ may be in the form of

$$P\left(d_j|c_k\right) = P\left(f_{j1},\ldots,f_{jn}|c_k\right) = \prod_{l=1}^{n} P\left(f_{jl}|c_k\right).$$

In other words, customer c_{k}'s preferences for these features determine the probability of c_{k} purchasing d_{j}. We propose a predictive framework for customer purchase behavior called *C*ust*O*mer purchase p*RE*diction mode*L* (COREL), which takes the form of

$$P\left(d_j|c_k,\ d_i\right) = \frac{1}{Z}P\left(d_j|d_i\right)\prod_{l=1}^{n} P\left(f_{jl}|c_k\right),$$

where Z is a normalization factor.

Jingdong (www.jd.com) is a well-known B2C e-commerce website in People's Republic of China. Figure 2 shows the product-related information presented on Jingdong. An analysis of these product features indicates that they may be classified into three categories.

- Certain product features are depicted only in the image that may be easily recognized by the user but not by the automatic program, such as product appearance (e.g., color, size, style).
- Dynamic product features. In the e-commerce context, certain product information will change each time the product is purchased and rated, such as the number of reviews, the average rating and sales.
- Product features that are static and observable by the analyst, e.g., price and brand.

It is necessary to employ different methods to learn users' preferences for these three categories of product features. This study conceives of the following methods:

(a) It may be extremely difficult to learn customers' preferences for dynamic product features. However, we find that many customers exhibit similar tendencies with respect to part of dynamic product features. For instance, these customers always feel inclined to buy hot-selling products with high ratings and numerous customer reviews. Therefore, we propose a model called the *heat model* to summarize these features and thereby calculate a popularity score for each candidate product based on the current status of its dynamic features.

(b) Discrete choice models are widely used by firms to analyze individuals' choices among a set of products [30]; these choices reveal customer preference for static and observable product features [2]. We develop a hierarchical Bayesian discrete choice model to learn customers' price sensitivity and brand preferences.

(c) CF may be used to estimate a customer's rating for a particular e-commerce product by exploiting product ratings by customers with similar taste. We employ CF to learn customers' preferences for features that cannot be directly observed by the analyst.

3.2 Methodology

The predictive framework comprises as a two-stage process. First, $P(d_\mathrm{j}|d_\mathrm{i})$is used to predict the customer's motivations, i.e., to build a collection of candidate products ω. Second, the preferences of customer c_k for features $\prod_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{P}(f_i|c_k)$ are used to determine which products of ω are most likely to be purchased. Figure 3 presents the general process of the predictive framework.

Estimating parameter $P(d_\mathrm{j}|d_\mathrm{i})$, constructing a *heat model* and developing a hierarchical Bayesian discrete choice model are key to building the predictive framework.

3.2.1 Estimating $P(d_j|d_i)$

Parameter $P(d_\mathrm{j}|d_\mathrm{i})$ represents the association between both products d_i and d_j. If a customer bought d_i, the parameter may reveal his/her motivations for d_j. Market basket analysis can be employed to estimate this parameter. Specifically, when two products occur in the same market basket, it is generally thought that there exists an association between them. Using maximum likelihood estimation, $P(d_\mathrm{j}|d_\mathrm{i})$ takes the form of

$$P\left(d_j|d_i\right) = \frac{|d_i \cap d_j|}{|d_i|}, \tag{1}$$

where $|d_\mathrm{i}|$ denotes the number of product d_i purchased by the customer and $|d_\mathrm{i} \cap d_\mathrm{j}|$ is the frequency with which both products d_i and d_j co-occur in the same market basket. However, the experiment discussed in Sect. 4.3 demonstrates that the

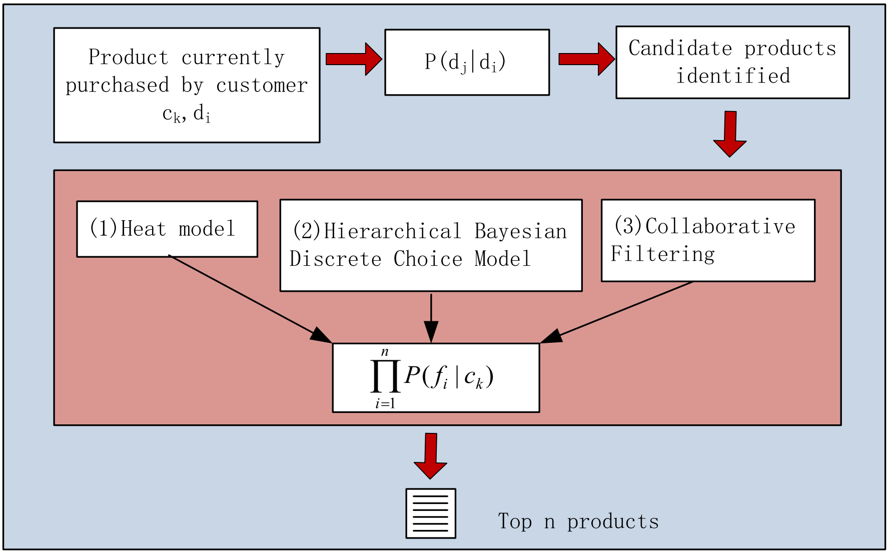

Product currently purchased by customer c_k, d_i
$P(d_j|d_i)$
Candidate products identified
(1)Heat model
(2)Hierarchical Bayesian Discrete Choice Model
(3)Collaborative Filtering
$\prod_{i=1}^{n} P(f_i | c_k)$
Top n products

Fig. 3 The working process of COREL

collection of product candidates developed using formula (1) is so small that *COREL* fails to achieve a good performance.

Therefore, we propose to build associations between categories and then obtain product candidates from categories associated with a particular product. Generally, e-commerce websites organize products into multi-level categories. For instance, Jingdong uses three category levels for its products; the categories of the “EPSON LQ-630k Printer” in order from first level to third level are “Computer or Office Equipment→Printing related Office Equipment→Printer”. We generate category associations at the third-level category using formula (2). $Thr(d_i)$ denotes the third-level category of product d_i.

$$P\left(d_j|d_i\right) = \frac{|Thr(d_i) \cap Thr(d_j)|}{|Thr(d_i)|}. \tag{2}$$

The experiments discussed in Sect. 4.3 demonstrate that the association of categories can broaden the candidate collection and thereby lead to a better performance.

3.3 *Heat model*

As discussed above, it is difficult to learn customers’ preferences for dynamic product features. However, we find that many customers exhibit similar tendencies with respect to particular product features. Figure 2 illustrates four product-related features described on the Jingdong website: (1) *Qr*, the number of reviews; (2) *Qs*, the average rating; (3) *Qa*, the number of days since the product’s on-shelf date; and

(4) *Qu*, the number of days since the most recent review. We seek to summarize customers' preferences for these features by calculating a popularity score.

Prior work has shown that support vector regression (SVR) [27] is an excellent tool for predictive tasks. We develop an SVR-based model called the *heat model* $H(d_i)$ that can calculate the popularity of products based on features *Qr*, *Qs*, *Qa* and *Qu*. A training set is a necessary component for learning the *heat model*. To the best of our knowledge, no e-commerce website provides labeled data regarding product popularity. However, we observe that a visitor might be able to determine which of two online products is more popular. Inspired by this observation, we conceive of the following steps to generate at raining set (steps 1–3) and to train an SVR model using this training set (step 4). Figure 4 illustrates the process of building the *heat model*.

Step 1 crowdsourcing is an online, distributed problem-solving and production model. Gathering data to train an algorithm is a common use of crowdsourcing [24, 1]. With this notion in mind, we develop a crowdsourcing system in which crowdworkers must select the more popular product from a pair of candidates displayed on a web page. The interface of this system is shown in Fig. 5.

The system runs according to the following process. Two products, *A* and *B*, are randomly selected from a product database; the webpage displays features *Qr*, *Qs*, *Qa* and *Qu* for both products; and the participant selects the more popular candidate. If the choice is product *A*, the two instances listed in the following table are generated for the training set.

Err_Qr	*Err_Qs*	*Err_Qa*	*Err_Qu*	Label
$Qr(A) - Qr(B)$	$Qs(A) - Qs(B)$	$Qa(A) - Qa(B)$	$Qu(A) - Qu(B)$	1
$Qr(B) - Qr(A)$	$Qs(B) - Qs(A)$	$Qa(B) - Qa(A)$	$Qu(B) - Qu(A)$	−1

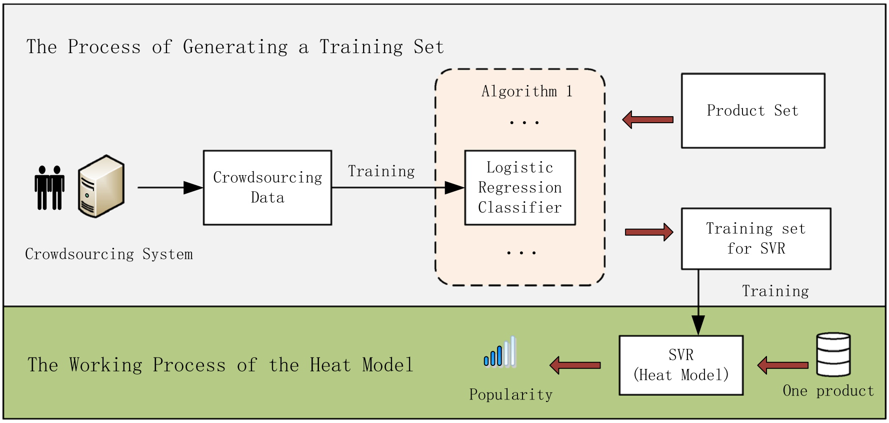
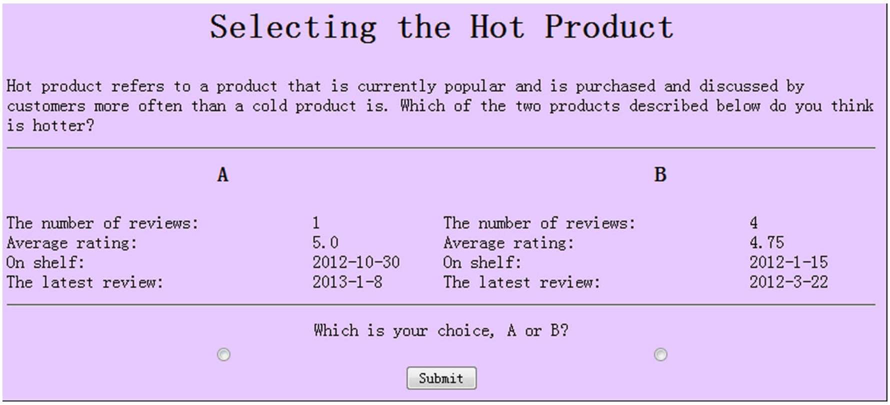


Fig. 4 The process of building the *heat model*



Selecting the Hot Product

Hot product refers to a product that is currently popular and is purchased and discussed by customers more often than a cold product is. Which of the two products described below do you think is hotter?

A		B	
The number of reviews:	1	The number of reviews:	4
Average rating:	5.0	Average rating:	4.75
On shelf:	2012-10-30	On shelf:	2012-1-15
The latest review:	2013-1-8	The latest review:	2012-3-22

Which is your choice, A or B?

○ ○

Fig. 5 The interface of a crowdsourcing system

In the training set obtained with this system, one instance includes five fields: *Err_Qr*, *Err_Qs*, *Err_Qa*, *Err_Qu* and label. *Qr*(*A*) denotes the feature *Qr* of product *A*.

Step 2 a classifier must be constructed to compare the popularity of two products. Logistic regression (LR) has proved to be an excellent tool for addressing binary classification problems. Additionally, the experiments discussed in Sect. 4.2 demonstrate that LR outperforms k-nearest neighbor (KNN) and DT models in classifying the crowdsourcing data. Thus, we build a LR model $f(\varphi)$ to compare the popularity of two products. φ in the model is a vector where the elements denoted as *Err_Qr*, *Err_Qs*, *Err_Qa*, and *Err_Qu* represent differences between the features of the two products being compared.

$$f(\varphi) = \frac{\exp(\pi(\varphi))}{1 + \exp(\pi(\varphi))},$$

where

$$\pi(\varphi) = \beta_0 + \beta_1 \times Err\_Qr + \beta_2 \times Err\_Qs + \beta_3 \times Err\_Qa + \beta_4 \times Err\_Qu.$$

We employ the training set generated in step 1 to learn the parameters of the LR model.

Step 3 to generate a training set for the SVR model, we collect a product set ω by randomly selecting 1000 products from Jingdong. Each product in the set must be assigned a popularity score. We propose algorithm 1 to accomplish this task.

```
Algorithm 1:    Calculating the Popularity of Products
Input: a collection of products $\omega$, logistic regression model $f(\varphi)$
Output: the popularity of products in $\omega$

$Pd \leftarrow []$
For each $<a, b>$, $a, b \in \omega$, $a \neq b$
    $\varphi = V(a)-V(b)$
    $r=f(\varphi)$
    $Pd[a]= Pd[a]+r-0.5$; $Pd[b]= Pd[b]+0.5-r$
End
normalize $Pd$ to range [0,1]
Return $Pd$
```

In algorithm 1, the array *Pd* stores the calculated popularity of all products in ω within the range [0, 1]. $V(a)$ denotes the feature vector of a product a.

Employing algorithm 1 to calculate the popularity of each product in ω generates a training set for the SVR model. Two instances in the training set are shown in the following table, where *score* refers to the popularity of a product and $ln(Qr)$ is the natural log of the *Qr* attribute.

$ln(Qr + 1)$	Qs	$ln(Qa + 1)$	$ln(Qu + 1)$	*Score*
1.0986	4	5.9054	5.8833	0.23539
0.69315	5	6.0497	5.9636	0.32821

Step 4 we use the training set generated in step 3 to train an SVR model called the *heat model*. In this study, we compare ε-SVR and μ-SVR, which use the polynomial kernel and the radial basis function as the kernel function of SVR, respectively. Because there is little general guidance on determining the parameters of SVR, this study varies the parameters to select the optimal values for the best prediction performance. We use the LIBSVM software system [5] to build an SVR model. Experimental results show that ε-SVR with a radial basis function fits well with our purpose, i.e., calculating popularity.

Given a set of data points $\{(X_1, z_1),\ldots,(X_l, z_l)\}$, where $X_i \in R^n$ is an input and $z_i \in R^1$ is a target, the standard form of ε-SVR [27] is

$$\min_{w,b,\xi,\xi*} \frac{1}{2} w^T w + C\sum_{i=1}^{l} \xi_i + C\sum_{i=1}^{l} \xi_i^* .$$

Subject to

$$\begin{gathered} W^T \phi(X_i) + b - z_i \le \varepsilon + \xi_i, \\ z_i - W^T \phi(X_i) - b \le \varepsilon + \xi_i^*, \\ \xi_i,\ \xi_i^* \ge 0, \quad i = 1, \ldots, l. \end{gathered}$$

Experimental analysis indicates that the *heat model* obtains the best performance in this study when the parameters of ε-SVR are $C = 1$ and $\varepsilon = 0.3$.

3.3.1 A hierarchical Bayesian discrete choice model

Economic models of choice typically assume that an individual's latent utility is a function of brand and attribute preference [30]. We develop a hierarchical Bayesian discrete choice model to calculate the probability of c_k choosing d_j based on his/her brand preference and price sensitivity, $DC(c_k, d_j)$.

We divide the price and brand of each product into three levels: high, medium and low price; large, moderate and small brand (Sect. 4 gives an example). Accordingly, the feature vector x of a product d has six binary value features $x = (p\_hi, p\_me, p\_lo, b\_la, b\_mo, b\_sm)$ corresponding to three price levels and three brand levels. Only one of the three price levels in the feature vector has a value of 1 whereas the others have a value of 0. As an example, ($p\_hi = 1$, $p\_me = 0$, $p\_lo = 0$) indicates that the price of a product is at a high level. Brand features are also subject to this rule. For instance, ($b\_la = 0$, $b\_mo = 0$, $b\_sm = 1$) means that a product belongs to a small brand. A hierarchical Bayesian discrete choice model is in the form of

$$DC\left(c_k,\ d_j\right) = P\left(y_j = 1\right) = \frac{1}{1 + \exp(-V(d_j,\ c_k))}. \tag{3}$$

Utility function

$$U\left(d_j,\ cf_k\right) = V\left(d_j,\ c_k\right) + e_{jk},$$

$$\begin{aligned} V\left(d_j,\ c_k\right) = {} & \beta_1 \times p\_hi + \beta_2 \times p\_me + \beta_3 \times p\_lo + \beta_4 \times b\_la \\ & + \beta_5 \times b\_mo + \beta_6 \times b\_sm. \end{aligned}$$

$P(y_j = 1)$ denotes the probability of a customer selecting product d_j. Every customer may have particular preferences regarding price and brand. For example, one customer may prefer large-brand products whereas other customers may not care about a product's brand as long as the product's price is low. In the hierarchical Bayesian model, the coefficients of utility function are decided by customer features, which are described in Sect. 5.

Use B denotes $\beta_1 \sim \beta_6$.

$$B = Z\Delta + U; \quad u_i \sim N\left(0,\ V_\beta\right).$$

Matrix Z contains customer features. The coefficient matrix Δ has a normal distribution with means $vec(\overline{\Delta})$ and covariance matrices given by the Kronecker product of A^{-1} and V_β.

$$\beta_n \sim N\left(\Delta' Z_n,\ V_\beta\right),$$

$$vec(\Delta) \sim N\left(vec(\bar{\Delta}),\ A^{-1} \otimes V_\beta\right),$$

$$V_\beta \sim IW(v,\ V).$$

The *vec* operator creates a column vector from a matrix by stacking the column vectors of [13]. Hyperparameter V_β has an inverted Wishart prior. We set noninformative prior v, V, and A to $v = m + 3$, $V = v \cdot I$, $= 0$ and $A = 0.01$, respectively, where m is the number of coefficients in the utility function.

We employ the Metropolis–Hastings MCMC algorithm to estimate the parameters of the hierarchical Bayesian model, proposing a normal distribution for the MCMC algorithm. The log-likelihood function is

$$L(X,\ Y,\ B) = \sum_i \log(P(x_i) * y_i + (1 - P(x_i)) * (1 - y_i)),$$

$$P(x_i) = \frac{\exp(x_i * B)}{1 + \exp(x_i * B)}.$$

The steps for estimating the parameters of the hierarchical Bayesian model are as follows.

Algorithm 2. Using the Metropolis-Hastings MCMC algorithm to estimate parameters

```
1. initiating $\beta_{\text{old}}$
2. draw from $V_\beta | v, V \sim IW(v+n, V+S)$
3. draw from $vec(\Delta) \,|\, \bar{\Delta}, A, V_\beta \sim N(vec(\bar{\Delta}), A^{-1} \otimes V_\beta)$
4. draw $\beta_{\text{new}} \sim N(\beta_{\text{old}}, V_\beta)$
5. Compute
      $\alpha(\beta_{old}, \beta_{new}) \sim \min(1, p(\beta_{new}) q(\beta_{new}, \beta_{old}) / p(\beta_{old}) q(\beta_{old}, \beta_{new}))$
      where
      $p(\beta_{new}) / p(\beta_{old}) = \exp(L(X, Y, \beta_{new}) - L(X, Y, \beta_{old}))$,
      $q(\beta_{new}, \beta_{old}) / q(\beta_{old}, \beta_{new}) = \exp\{(\beta'_{new} - Z\Delta) * V_\beta * (\beta_{new} - (Z\Delta)') - (\beta'_{old} - Z\Delta) * V_\beta * (\beta_{old} - (Z\Delta)'))\}$
6. If $\alpha < 1$ then
7.    $\beta_{\text{old}} = \beta_{\text{new}}$ with probability $\alpha$
8. else
9.    $\beta_{\text{old}} = \beta_{\text{new}}$
10. End
11. Goto step (2) until convergence
```

Using saved draws, we can plot the posterior distribution of coefficients. As illustrated in Fig. 6, the means of the posterior distributions of the three coefficients *p_hi*, *p_me* and *p_lo* for one customer are approximately −2.7, 0.8 and 0.5, respectively. We can conclude based on the point estimates of these three coefficients that the customer generally rejects high-priced products and tends to prefer medium-priced products over low-priced products.

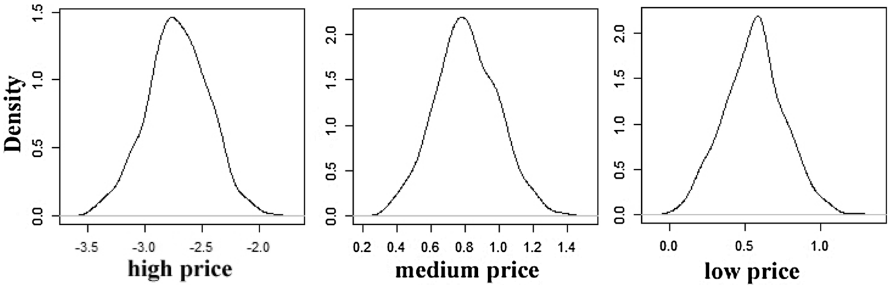


Fig. 6 The posterior distributions of the coefficients of a customer

To train a hierarchical Bayesian discrete choice model, it is necessary to know customers' options in a finite alternative set. However, in the e-commerce context, we can only know which products a customer purchased; we cannot know which products were available to but declined by the customer. When training a hierarchical Bayesian discrete choice model, both positive and negative samples are necessary components. Regarding the purchased products as positive data, we develop a technique to generate one negative instance from each positive instance. One instance in the training dataset is a feature vector of a purchased product combined with a label. The six features *p_hi*, *p_me*, *p_lo*, *b_la*, *b_mo*, and *b_sm* in each instance represent the product's price and brand levels.

p_hi	*p_me*	*p_lo*	*b_la*	*b_mo*	*b_sm*	Label
1	0	0	1	0	0	1

When each feature in the positive instance is inverted, we can derive a negative instance.

p_hi	*p_me*	*p_lo*	*b_la*	*b_mo*	*b_sm*	Label
0	1	1	0	1	1	0

3.3.2 Collaborative filtering

CF may be used to estimate a customer's rating for a particular e-commerce product based on the product ratings by customers with similar tastes. We employ CF to learn customers' preferences for product features that cannot be observed by analyst. We predict c_k's rating for d_j using collaborative filtering $CF(c_k, d_j)$ that is calculated using formula (4).

$$CF\left(c_k,\ d_j\right) = \frac{\sum_{s \in S} Sim(c_k,\ s) \times rating(s,\ d_j)}{|S|}, \tag{4}$$

where S denotes a set of customers that comprises the top 10 customers most similar to c_k and rating(s, d_j) refers to a rating that customer s gives to product d_j. The possible rating values are defined on a numerical scale from 0 (strongly dislike) to 5 (strongly like). $Sim(c_k,\ s)$ stands for the similarity between customers c_k and s, which can be calculated using the cosine measure. A customer feature vector is defined as a set of product ratings. Consider the following case in the feature vector of c_k: $V(c_k) = (0, 4, 1, 0, 5)$. This case indicates that c_k did not purchase product d_1 (or that he/she gave product d_1 a rating value of 0) and gave d_2 a rating value of 4, etc.

$$Sim(c_k,\ c_l) = \frac{V(c_k)V(c_l)}{|V(c_k)||V(c_l)|}. \tag{5}$$

4 Experiments

We collected customer information and product reviews from Jingdong using a web crawler. The collected data contain 727,878 product items, 342,451 customers and 14,634,059 reviews from 2004 to 31 January 2013. Jingdong's products are assigned three levels of categories. There are 19 first-level categories, 124 second-level categories and 1078 third-level categories.

It is difficult to collect customer purchase information directly from e-commerce websites because these data are generally regarded as private. Therefore, our study is based on the following assumption: if a customer frequently writes reviews on an e-commerce website, his/her reviews can reveal nearly all of his/her product purchases (on Jingdong, only customers who have purchased a product are authorized to write reviews for that product). Therefore, we identify customers with high reviewing frequencies and generate purchase data based on their reviews. In addition, we recruited 55 participants on the crowdsourcing platform; these participants generated training data containing 1351 * 2 instances.

We use an IBM computer with 16 × 2 GHz CPU and 64G memory to cope with very large matrixes generated in the experiments.

4.1 Data processing

The collected data are processed as follows.

- *Dataset division* the dataset is divided into three sections by date. A section: before 30th June 2012, B section: from 30th June 2012 to 31st July 2012, and C section: after 31st July 2012. Figure 7 illustrates these divisions.
- *Customer selection* we identify customers who purchased products in all three sections and for whom the number of items in A section is greater than 30. A total of 2770 customers meet this requirement.
- *Training set* the purchase data in A section.
- *Target set* the products in B section.
- *Test set* the products in C section.
- *Setting price levels for products* let d be an item and thr be its third-level category. If the price of d is above the 75th percentile for all product prices in

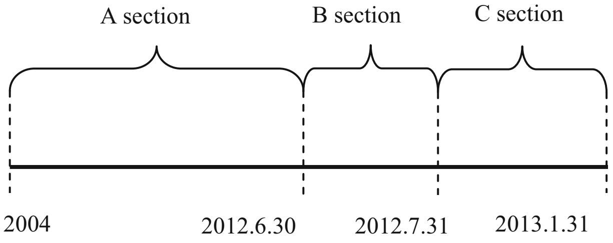


Fig. 7 Dividing the dataset into three sections

thr, we assign the features of d to ($p\_hi = 1$, $p\_me = 0$ and $p\_lo = 0$). If the price of d is below the 25th percentile for price, its features are assigned to ($p\_hi = 0$, $p\_me = 0$ and $p\_lo = 1$). Otherwise, its features are assigned to ($p\_hi = 0$, $p\_me = 1$ and $p\_lo = 0$).

- *Setting brand levels for products* we examine the distribution of the number of products for each brand. If the number of a brand's products is greater than the 75th percentile of the distribution, the features of all items under that brand are set as ($b\_la = 1$, $b\_mo = 0$ and $b\_sm = 0$). If a brand lies below 25th percentile of the distribution, features of its products are set as ($b\_hi = 0$, $b\_mo = 0$ and $b\_sm = 1$). Otherwise, its features are assigned to ($b\_hi = 0$, $b\_mo = 1$ and $b\_sm = 0$).

This paper refers to the processed data as the JD dataset.

4.2 Exploring crowdsourcing data

In this section, we examine whether the data samples collected from the crowdsourcing system are suitable for our task. In other words, we evaluate whether participants make similar judgments regarding product popularity based on the product features presented to them. If participants exhibit similar judgments, the feature space of the collected data should be partitionable. Based on this idea, we employ LR, DT and KNN models to build classifiers based on the collected data and use a 10-fold validation method to examine the precision of the three classifiers.

$$\text{Precision} = \frac{\text{The number of correctly classified instances}}{\text{The total number of instances in the dataset}}.$$

The results presented in Fig. 8 demonstrate that all three models perform well, with precision scores of 92.4, 90.6 and 84.5 % for the LR, DT and KNN models, respectively. This experiment demonstrates that the feature space of the collected data is partitionable; it also implies that crowdworkers have similar views regarding product popularity.

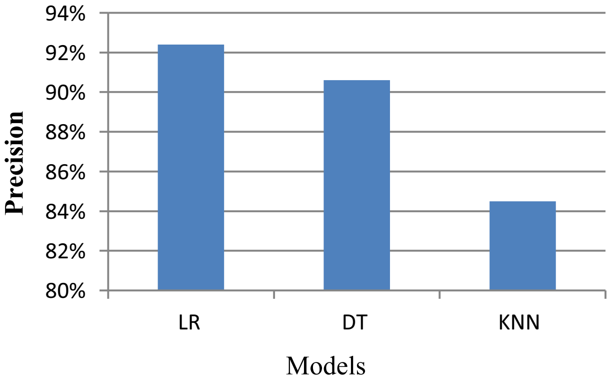


Fig. 8 Classification of data gathered from the crowdsourcing system

The uncertain quality of crowdsourced labels is a challenge for the crowdsourcing system, and a detailed discussion on this subject is beyond the scope of this paper. In this study, a classifier with 92.4 % precision is adequate for our purpose, i.e., calculating product popularity, even if unreliable crowdworkers might exist.

4.3 Exploring parameter $P(d_j|d_i)$

In this section, we investigate the impact of parameter $P(d_j|d_i)$ on the performance of the predictive framework as follows: we choose the last product purchased d_i by customer c_k in *B* section; we acquire an associated product set ω for d_i based on data in *A* section using formula (1) [if $P(d_j|d_i) > 0$, we say that d_j is associated with product d_i]; we use the products in ω as our prediction result and the products purchased by c_k in *C* section as the test set Φ. If any product in ω occurs in Φ, we say that customer c_k is successfully predicted. An analysis of all customers yields the results provided in Table 1.

When the top 10 associated products are obtained for each customer, only 9.9 % of customers are successfully predicted. When we obtain the top 100 associated products, the precision improves to 26.2 %. However, if the predictive framework is to be used as a recommender system, it is totally impractical to recommend 100 items to an e-commerce customer in one time.

In the JD dataset, the application of formula (1) to each product yields an average of 33 associated products. However, when we obtain the top 100 associated products for each item, the predictive performance only reaches 26.2 %. This result means that the use of formula (1) to acquire associated products harms *COREL*'s performance.

Continuing this scenario, we build a basic model that uses associated categories for prediction. We obtain the third-level category of d_i, named *thr*; we use formula (2) to acquire the top 1, 5 and 10 associated categories of *thr*; and we reduce all products in test set Φ to their third-level category. If any associated category of d_i occurs in Φ, we say that customer c_k is correctly predicted. Table 2 presents the results.

Table 2 indicates that obtaining the top 10 associated categories achieves the best performance. Specifically, 75.3 % of customers will purchase products in at least 1 of the top 10 categories in the future. However, this result does not mean that

Table 1 Precision of using associated products for predictions

	Top 10 (%)	Top 100
Precision	9.9	26.2

Table 2 Exploring associated categories

	Top 1 (%)	Top 5 (%)	Top 10 (%)
Precision	26.7	56	75.3

continuing to increase top n will improve COREL's performance; rather, expanding the candidate set further can introduce numerous marginally associated products.

4.4 Exploring collaborative filtering for prediction

For customer c_k, we use formula (4) to acquire a collection of products ω, which contains the top 10 CF-score products not reviewed by c_k in the A and B sections. The products purchased by c_k in C section are employed as test set Φ. If any product $d \in \omega$ occurs in test set Φ, we say that c_k is correctly predicted. An analysis of all customers achieves a precision rate of only 2.6 %. This result means that the use of CF alone is unsuitable for predicting customer purchase behavior (Table 3).

Continuing the experiment, we build a model called M2 = $P(d_j|d_i)$ * CF. This model uses the top 10 associated categories of d_i to build candidate sets and employs formula (4) to calculate CF scores for products in the candidate sets. We build three subsets by obtaining the top 1, 5 and 10 products. If any one of the products in a particular subset occurs in test set Φ, we say that c_k is correctly predicted. After exploring all customers in the JD dataset, we determine the precision of the predictive framework for each of the three subsets. These results are shown in Table 4.

The results show that combining CF and $P(d_j|d_i)$ dramatically improves predictive performance compared with using CF alone. From a recommender system perspective, it can be said that when the model recommends 10 products to customers, 27.7 % of those customers will purchase at least 1 of the recommended products. The model that combines $P(d_j|d_i)$ and CF (M2) also outperforms the basic model that uses only $P(d_j|d_i)$ (Table 1).

4.5 Exploring the *heat model*

Let d_i be the last product purchased by customer c_k in B section and $Thr(d_i)$ be the third-level category of d_i. As described in Sect. 4.4, we obtain the top 10 associated categories of $Thr(d_i)$ to generate a candidate set. Then, the *heat model* is used to calculate the popularity of each product in the candidate set and to form three product subsets, namely, the top 1, 5 and 10 most popular products. If a product in one of these subsets occurs in test set Φ, we say that c_k is successfully predicted. An

Table 3 Precision of CF

	CF (%)
Precision	2.6

Table 4 Precision of model M2

	Top 1 (%)	Top 5 (%)	Top 10 (%)
Precision	9.9	19.9	27.7

analysis of all customers in the JD dataset is conducted to acquire precision rates for the three subsets. Table 5 presents the results.

This experiment shows that the use of the *heat model* alone for the prediction task yields a poor performance. Accordingly, we build model M3 = M2 * H(d_i), which combines the *heat model* and the M2 model. We repeat the above experiment using M3 and obtain the results shown in Table 6.

A comparison of models M3 and M2 indicates that M3 outperforms M2 by up to 5 % when the top 10 candidates are obtained. This result demonstrates that incorporating the *heat model* into M2 significantly improves the performance of the predictive framework.

4.6 Exploring the performance of *COREL*

This section investigates the performance of *COREL* in predicting customer purchase behavior. We compare *COREL* to several baseline models, which are identified in Table 7.

Recommender systems based on a Markov chain model utilize sequential basket data by predicting the user's next purchase action based on the user's last purchase action. By comparison, a factorization method based on matrix or tensor decomposition learns the general tastes of the user but disregards sequential

Table 5 Performance of the *heat model*

	Top 1 (%)	Top 5 (%)	Top 10 (%)
Precision	5.6	9.3	13.1

Table 6 Performance of model M3

	Top 1 (%)	Top 5 (%)	Top 10 (%)
Precision	11.4	23.5	32.9

Table 7 Predictive models

Names	Models	Description
M1	$P(d_i \mid d_j)$	Uses formula (1) to obtain associated products
M2	$P(d_i \mid d_j)$ * CF	Uses formula (2) to build candidate sets and employs collaborative filtering to obtain products from candidates
M3	$P(d_i \mid d_j)$ * CF * H	Combines M2 with the *heat model*
M4	*COREL* without brand	*COREL* is used without considering customers' brand preferences
M5	*COREL*	Incorporates customers' price and brand preferences
M6	*FPMC*	Factorized personalized Markov chain recommendation model
M7	$\mathrm{SVD}_{\mathrm{util}}^{0.7}$	Utilizes marginal net utility for recommendation

information. The FPMC model [22] subsumes both a common Markov chain and the normal matrix factorization model. In this study, we implement an FPMC algorithm-based predictive model that makes item recommendations based on sequential basket data in the JD dataset. The parameters of implementing the FPMC algorithm are listed in Table 8.

We also implement the recommendation algorithm $\mathrm{SVD_{util}}$ [28], which utilize marginal net utility for recommendation. $\mathrm{SVD_{util}}$ assumes that each entry $\mathrm{P_{u,i}}$ in a user–product matrix $\mathrm{P_{M\times N}}$ can be estimated using the form $\overline{\mathrm{P}}_{\mathrm{u,i}} = \mathrm{q_i^T p_u}$ where $\mathrm{q_i}$ and $\mathrm{p_u}$ are vectors, which are the hidden representation of product *i* and user *u*. These vectors can be estimated based on all given entries in $\mathrm{P_{M\times N}}$. The value of $\mathrm{P_{u,i}}$ in the observed matrix $\mathrm{P_{M\times N}}$ is determined by the user purchase history. $\mathrm{SVD_{util}}$ ranks all products by their estimated $\overline{\mathrm{P}}_{\mathrm{u,i}}$ values and selects the top *n* to recommend. On implementing $\mathrm{SVD_{util}}$, we set parameter $\theta = 0.7$ whereas other parameters such as learning rate β and regularization parameters λ are the same with those in [28]. Additionally, [28] uses product titles to calculate similarity of both products Sim(i, j). In the experiment, we calculate the similarity use the following form of

$$\mathrm{Sim}(i,\ j) = 0.5 \times \text{category similarity} + 0.5 \times \text{similarity of product title},$$

where the category similarity $= \begin{cases} 1 & \text{same in third-level category,} \\ 0 & \text{otherwise.} \end{cases}$

The seven models listed in Table 7 make predictions based on the JD data set. These models use the last product purchased d_i by customer c_k in *B* section to predict items purchased by c_k in *C* section. The candidates are generated by obtaining the top 10 associated categories of d_i. Products purchased by c_k in *C* section are used as test set Φ. These models calculate a prediction score for each candidate and select the top *n* candidates to build product subset ω. If any product in ω occurs in test set Φ, we say that customer c_k is successfully predicted. Using precision as a measure of model performance, we present the results for all customers in the JD dataset in Table 9.

Table 8 Parameters of implementing the FPMC model

Parameters	Values
Factorization dimensions $\mathrm{K_{U,I}}$, $\mathrm{K_{I,U}}$	128
Factorization dimensions $\mathrm{K_{I,L}}$, $\mathrm{K_{L,I}}$	1000
Regularization parameters $\lambda_{\mathrm{U,I}}$, $\lambda_{\mathrm{I,U}}$, $\lambda_{\mathrm{I,L}}$, $\lambda_{\mathrm{L,I}}$	0.075

Table 9 Evaluation of the predictive performance of seven models

N	M1 (%)	M2 (%)	M3 (%)	M4 (%)	M5 (%)	M6 (%)	M7 (%)
1	4.7	9.9	11.4	15.8	13.2	5.2	5.0
3	6.8	13.1	15.7	17.5	15.9	6.8	6.8
5	8.9	19.9	23.5	23.5	23.5	9.1	8.1
10	9.9	27.7	32.9	32.2	32.3	12.4	11.9

As shown in Table 9, market basket analysis (M1) exhibits the poorest performance in predicting customer purchase behavior. The combination of CF and market basket analysis (M2) dramatically improves the model's predictive performance. When M2 is combined with the *heat model* (M3), precision is further increased. Model M4, which incorporates customers' price sensitivity, outperforms other models when $n = 1$; this result means that the introduction of price sensitivity into the predictive task can improve performance. However, the addition of customers' brand preferences to the model (M5) does not significantly improve model performance and even decreases model precision when $n = 3$ and 1. Section 5 discusses reasons for this decrease in performance.

We also observe that model M4 significantly outperforms the FPMC model (M6) and $\mathrm{SVD}_{\mathrm{util}}$ (M7). We believe that using only sequential basket data and user–item correlation and omitting customers' preferences regarding product features make FPMC and $\mathrm{SVD}_{\mathrm{util}}$ inappropriate for predicting customer purchase behavior.

5 An analysis of customers in the JD dataset

In this section, we discuss the characteristics of customers in the JD dataset by examining the estimated parameters of the hierarchical Bayesian model introduced in Sect. 3.3. Table 10 lists customer variables.

To explore the relationships between customers' features and their preferences, all variables are normalized into the range [0, 1]. Table 11 shows the estimated parameters of the hierarchical Bayesian model.

Based on the data in Table 11, we can conclude that customers with high purchase frequencies, large *SD*s and low monetary values tend to prefer small-brand products, whereas customers who have low purchase frequencies and high monetary values show an inclination toward large brands. It can be further inferred that small-brand products are generally less expensive than large- or moderate-brand products.

Figure 9 shows the posterior distributions of the six model parameters. Although the posterior distributions of both moderate and small brands place most of their respective mass on negative values, the posterior distribution of large brands retains the bulk of its mass approximately 4. We can thus conclude that taken as a whole,

Table 10 Customer variables

Variables	Description
Recency	Number of months since the customer's last purchase
Frequency	Number of purchases during the last 12 months
Monetary	The dollar value of the customer's purchases during the last 12 months
Age	Amount of time in years between the present date and the date of the customer's first purchase from the website
SD	Standard deviation of the prices of all products purchased by the customer. This value reveals the customer's online shopping habits. A smaller SD means that the customer likes to purchase a fixed variety of products whereas a larger SD indicates that the customer makes online purchases of both expensive and inexpensive products

Table 11 Posterior mean of parameter Δ in the hierarchical Bayesian model

	Intercept	Age	Recency	Frequency	Monetary	SD
High price	−2.127	1.271	−1.688	−6.212	5.182	1.166
Medium price	0.645	2.493	−1.873	−1.362	2.569	−1.998
Low price	0.919	3.775	−0.924	2.14	5.994	−2.659
Large brand	4.801	−2.228	1.27	−10.405	24.665	−7.52
Moderate brand	0.03	−3.54	0.904	−10.904	13.126	−4.907
Small brand	−5.643	−1.732	2.266	26.607	−52.015	16.112

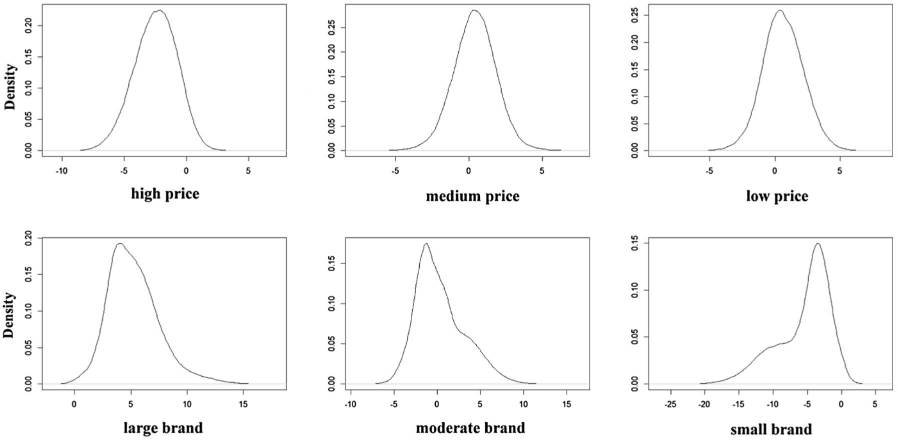


Fig. 9 Posterior distributions of parameters in model M5

the customers in the JD dataset have a greater preference for large-brand products than for small-brand products. In addition, customers prefer medium- and low-priced products to high-priced products. Brand has a lager impact on customer choice than price does. However, further analysis shows that large-brand products may have a higher probability of being purchased than small-brand products when brand level is divided by the number of items under that brand. Therefore, the brand preference derived from our model is not suitable for prediction purposes. Our experiments also demonstrate that it is not helpful to improve model performance in predicting purchase behavior. This result explains why model M5 exhibits a poorer performance than M4 does.

6 Conclusions

Researchers in the marketing and CRM fields have made numerous significant contributions to the prediction of customer purchase behavior in the traditional business context. However, new methods and techniques must be developed to perform the predictive task in the e-commerce context.

We propose a predictive framework called COREL for customer purchase behavior. This framework comprises a two-stage process. First, the associations between products are investigated and exploited to predicate customer's motivations, i.e., to build a candidate product collection. Second, customers' preferences for product features are learned to identify which candidate products are most likely to be purchased. This study investigates three categories of product features: dynamic product features, features that may be observed by the user but not by the analyst and product features that are static and observable by the analyst. We exploit the purchase data from an e-commerce website to develop methods to learn customer preferences for each of these three categories.

The results prove that our approach to calculating product popularity is feasible and that customer preference for product features has a significant impact on purchasing decisions.

Economic models of choice typically assume that an individual's latent utility is a function of brand preference. In this study, however, brand preference does not significantly improve the performance of COREL when we divide product brand level by the number of items under the brand. In the future, we will investigate an approach that can improve the performance of the predictive framework by incorporating customers' brand preferences into the model.

Acknowledgments This work was supported by Major Program of National Natural Science Foundation of China (No. 91218301), National Natural Science Foundation of China (No. 71473201) and Humanities and Social Science Foundation of Ministry of Education of China (No. 14YJAZH063).

References

1. Ahn, L. V., & Dabbish, L. (2004). Labeling images with a computer game. In *The Proceedings of the SIGCHI'04 Conference on Human Factors in Computing Systems* (pp. 319–326).
2. Archak, N., Ghose, A., & Ipeirotis, P. G. (2011). Deriving the pricing power of product features by mining consumer reviews. *Management Science, 57*(8), 1485–1509.
3. Balabanovic, M., & Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. *Communications of the ACM, 40*(3), 66–72.
4. Bodapati, A. V. (2008). Recommendation systems with purchases data. *Journal of Marketing Research, 45*(1), 77–93.
5. Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology, 2*(3), 27:1–27:27.
6. Chen, J., Jin, Q., Zhao, S., Bao, S., Zhang, L., Su, Z., et al. Does product recommendation meet its waterloo in unexplored categories?: No, price comes to help. In *Proceedings of the 37th International ACM Conference on Research and Development in Information Retrieval*, *SIGIR'14*.
7. Chen, Z.-Y., & Fan, Z.-P. (2012). Distributed customer behavior prediction using multiplex data: A collaborative MK-SVM approach. *Knowledge-Based Systems, 35*, 111–119.
8. Guo, Y., & Barnes, S. (2011). Purchase behavior in virtual worlds: An empirical investigation in second life. *Information and Management, 48*(7), 303–312.
9. Hung, C., & Tsai, C. (2008). Market segmentation based on hierarchical self-organizing map for markets of multimedia on demand. *Expert Systems with Applications, 34*(1), 780–787.
10. Kim, E., Kim, W., & Lee, Y. (2003). Combination of multiple classifiers for the customer's purchase behavior prediction. *Decision Support Systems, 34*(2), 167–175.
11. Koenigstein, N., & Koren, Y. (2013). Towards scalable and accurate item-oriented recommendations. In *Proceedings of the 7th ACM Conference on Recommender Systems* (pp. 419–422).

12. Li, Y., Hu, J., Zhai, C., & Chen, Y. (2010). Improving one-class collaborative filtering by incorporating rich user information. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management* (pp. 959–968).
13. Magnus, J. R., & Neudecker, H. (1995). *Matrix differential calculus with applications in statistics and econometrics*. New York: Wiley.
14. Manning, C. D., Raghavan, P., & Schutze, H. (2009). *Introduction to information retrieval*. Cambridge: Cambridge University Press.
15. Mobasher, B., Dai, H., Luo, T., & Nakagawa, M. (2002). Using sequential and non-sequential patterns in predictive web usage mining tasks. In *Proceedings of the 2002 IEEE International Conference on Data Mining* (pp. 669–672).
16. Moon, S., & Russell, G. J. (2008). Predicting product purchase from inferred customer similarity: An autologistic model approach. *Management Science, 54*(1), 71–82.
17. Paquet, U., & Koenigstein, N. (2013). One-class collaborative filtering with random graphs. In *Proceedings of the 22nd International Conference on World Wide Web* (pp. 999–1008).
18. Parikh, N., & Sundaresan, N. (2009). Buzz-based recommender system. In *Proceedings of the 18th International Conference on World Wide Web* (pp. 1231–1232).
19. Park, S.-H., Huh, S.-Y., Oh, W., & Han, S. P. (2012). A Social network-based inference model for validating customer profile data. *MIS Quarterly, 36*(4), 1217–1237.
20. Pazzani, M., & Billsus, D. (2007). Content-based recommendation systems. The adaptive web - Methods and Strategies of Web Personalization. *Lecture Notes in Computer Science, 4321*, 325–341.
21. Ponte, J., & Croft, W. B. (1998). A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 275–281).
22. Rendle, S., Freudenthaler, C., & Schmidt-Thieme, L. (2010). Factorizing personalized Markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web* (pp. 811–820).
23. Shi, S. W., & Zhang, J. (2014). Usage experience with decision aids and evolution of online purchase behavior. *Marketing Science, 33*(6), 871–882.
24. Snow, R., O'Connor, B., Jurafsky, D., & Ng, A. Y. (2008). Cheap and fast—But is it good?: Evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 254–263).
25. Somervuori, O., & Ravaja, N. (2013). Purchase behavior and psychophysiological responses to different price levels. *Psychology and Marketing, 30*(6), 479–489.
26. Suh, E. H., Noh, K. C., & Suh, C. K. (1999). Customer list segmentation using the combined response model. *Expert Systems with Applications, 17*(2), 89–97.
27. Vapnik, V. N. (1998). *Statistical learning theory*. New York: Wiley.
28. Wang, J., & Zhang, Y. (2011). Utilizing marginal net utility for recommendation in E-commerce. In *Proceedings of the 34th International ACM Conference on Research and Development in Information Retrieval* (pp. 1003–1012).
29. Wong, C. R., Fu, A. W., & Wang, K. (2005). Data mining for inventory item selection with cross-selling considerations. *Data Mining and Knowledge Discovery*, *11*(1), 81–112.
30. Yang, S., & Allenby, G. M. (2003). Modeling interdependent consumer preferences. *Journal of Marketing Research, 40*(3), 282–294.
31. Yang, Y., Liu, H., & Cai, Y. (2013). Discovery of online shopping patterns across website. *Informs Journal on Computing, 25*(1), 161–176.
32. Zhang, Y., & Pennacchiotti, M. (2013). Predicting purchase behaviors from social media. In *Proceedings of the 22th International Conference on World Wide Web* (pp. 1521–1531).
33. Zimdars, A., Chickering, D. M., & Meek, C. (2001). Using temporal data for making recommendations. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence* (pp. 580–588).