

Opportunity Models for E-commerce Recommendation: Right Product, Right Time

Jian Wang, Yi Zhang
School of Engineering
University of California, Santa Cruz
Santa Cruz, CA 95060 USA
{jwang30, yiz}@soe.ucsc.edu

ABSTRACT

Most of existing e-commerce recommender systems aim to recommend the right product to a user, based on whether the user is likely to purchase or like a product. On the other hand, the effectiveness of recommendations also depends on the time of the recommendation. Let us take a user who just purchased a laptop as an example. She may purchase a replacement battery in 2 years (assuming that the laptop's original battery often fails to work around that time) and purchase a new laptop in another 2 years. In this case, it is not a good idea to recommend a new laptop or a replacement battery right after the user purchased the new laptop. It could hurt the user's satisfaction of the recommender system if she receives a potentially right product recommendation at the wrong time. We argue that a system should not only recommend the most relevant item, but also recommend at the right time.

This paper studies the new problem: how to recommend the right product at the right time? We adapt the proportional hazards modeling approach in survival analysis to the recommendation research field and propose a new *opportunity model* to explicitly incorporate time in an e-commerce recommender system. The new model estimates the joint probability of a user making a follow-up purchase of a particular product at a particular time. This joint purchase probability can be leveraged by recommender systems in various scenarios, including the zero-query pull-based recommendation scenario (e.g. recommendation on an e-commerce web site) and a proactive push-based promotion scenario (e.g. email or text message based marketing). We evaluate the opportunity modeling approach with multiple metrics. Experimental results on a data collected by a real-world e-commerce website(shop.com) show that it can predict a user's follow-up purchase behavior at a particular time with decent accuracy. In addition, the opportunity model significantly improves the conversion rate in pull-based systems and the user satisfaction/utility in push-based systems.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.

Copyright 2013 ACM 978-1-4503-2034-4/13/07 ...\$15.00.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval

General Terms

Algorithms, Design, Experimentation

Keywords

Recommender System; Opportunity Model; E-commerce

1. INTRODUCTION

As online shopping becomes popular, e-commerce recommendation is an increasingly important business tool for promoting sales. Researchers and industry practitioners are looking for all possible approaches to improve the recommendation performance. Even a minor improvement could lead to a big business return.

Traditional recommender systems focus on finding the right item to recommend. Major approaches include content-based methods, collaborative filtering methods and hybrid methods. For example, if a user viewed or purchased some camera(s) in the website, the system recommends more similar items (e.g. similar cameras) to the user. Recent research [19] proposed that recommender systems should recommend items that maximize the users' marginal utility, instead of only items that a user likes. As the marginal utility of a camera decreases immediately after a user purchased a camera, a system's follow-up recommendation should include camera accessories instead of similar cameras.

On the other hand, the user satisfaction/utility depends on both the relevance and the time of the recommendation. While an irrelevant recommendation results in a negative utility, the opportunity cost of recommending a relevant item at the wrong time could also be high, as we wasted the space while giving the user a negative impression. This is especially a problem for email or message based recommendations, as it wastes a user's time and effort to receive product recommendation emails/messages that are full of products she does not need to purchase at the time. In the long term, the user may have negative impression about the company, unsubscribe from the marketing email list, label the emails as spams, or uninstall the message application.

To address these issues, recommender systems need to answer the following question: when is the *right time* for the system to make recommendations of the *right product(s)*? For example, after a user purchased a camera, whether and

when should the system recommend related accessories including camera lenses, batteries, digital photo frames, etc.?

Multiple heuristic approaches [18, 24] have been proposed to tackle this problem. In this paper, we propose a theoretical model to learn the probability of a user making a follow-up purchase at a particular time. The model is inspired by the hazards model in survival analysis in statistics. The purchase time would be influenced by multiple factors, such as the user’s characteristics, the user’s purchase history, the product promotion information, the global environment and so on. Thus we propose to leverage the proportional hazards modeling approach which incorporates related factors as covariates (i.e., features). We further extend the model with the hierarchical Bayesian framework to handle the data sparsity issue. The new model is denoted as the *Opportunity Model* in this paper. It predicts the joint purchase probability, i.e., the probability of a user purchasing a product at a particular time. It helps to promote a right item at the right time which further enhances the user satisfaction. Experimental results are performed with a dataset from a real-world e-commerce website. Detailed analysis shows that the opportunity model could help to significantly improve the conversion rate and the user satisfaction.

The major contribution of this paper includes:

- We propose a new research problem in recommender systems: When is the right time to make recommendation of the right product in the e-commerce domain?
- To solve this problem, we propose a principled approach, (i.e. the **opportunity model**), to predict the joint probability of purchasing a product and the time of the event. We extend the proportional hazards model in statistics with the hierarchical Bayesian framework as part of the solution, and derive detailed inference steps based on the variational Bayesian algorithm.
- We leverage the joint probability in both the zero-query pull-based recommendation scenario and the proactive push-based email/message promotion scenario¹. In particular, the probability enables a proactive recommendation agent to decide whether to send recommendations of certain items to a user at a particular time based on a solid utility optimization framework. Experimental results show that the opportunity modeling approach significantly improve the user satisfaction and the conversion rate of the system.

2. RELATED WORK

To provide recommendations to a user, recommendation systems usually predict a user’s ratings for each item or probability of purchasing, then rank all items in the descending order. There are two major recommendation approaches: content-based filtering and collaborative filtering. Content-based filtering [12] assumes that descriptive features of an item indicate a user’s preferences. Thus a recommender system estimates the score of each item based on descriptive features of other items the user likes or dislikes. Usually, the system recommends items that are similar to what the user liked before. On the other hand, collaborative filtering [7, 18, 13, 5] assumes that users with similar

¹In the pull-based scenario, a user comes to the website and views some recommendation. In the push-based scenario, a system sends promotion emails or messages to the user.

tastes on some items may also have similar preferences on other items. Thus the main idea is to use the behavior history from other like-minded users to provide the current user with good recommendations. Research on collaborative filtering algorithms reached a peak due to the 1 million dollar Netflix movie recommendation competition. Factorization-based collaborative filtering approaches [3, 9, 19, 17], such as the regularized Singular Value Decomposition, performed well on this competition. A common characteristic of these models is the introduction of user latent factors and/or item latent factors to solve the data sparsity issue. In the field of recommender systems, the effect of time [10, 18, 24] has received some research attention recently. One focus is about the drift of the user’s preference over time [10, 23, 15]. Koren [10] revamped two popular collaborative filtering methods by modeling the time drifting factor of user preferences. Rendel et al. [15] proposed a factorized personalized model that subsumes both a common Markov chain and the normal matrix factorization model.

Recommendation in the e-commerce domain is a topic that has been studied in the IR community [8, 20]. Several methods have been tried in this domain, including neighborhood-based method, graph models [6], MDP-based methods [16], multi attribute utility theory based methods [11] and so on. In general, most of these existing methods, directly or indirectly, only estimate whether a user would like an item or purchase an item. Some recent work studied modeling the time interval between purchase orders in the e-commerce domain [18, 24]. Wang et al. [18] discovered different post-purchase behavior in different time windows after purchasing. Zhao et al. [24] used the purchasing time interval to improve the temporal diversity of recommendations [15]. The time interval and the corresponding purchase probability is modeled inside the framework of a utility-based recommender system [19]. The hybrid system takes the time interval into consideration when ranking all candidate items. Compared to the prior work about time, our work explicitly models the joint probability of purchasing the product at the time based on a solid theoretical foundation. We explicitly model the conditional probability as a white box by leveraging Weibull distribution with various covariates to estimate the time-based probabilistic density. These covariates enable us to capture various time-dependent patterns such as local changes, cyclic behavior, seasonable pattern, general trend of follow-up purchase behaviors, which are not captured by the prior work. Besides, the joint probability of product and time enables us to improve the recommendation accuracy.

3. OPPORTUNITY MODEL IN E-COMMERCE

To recommend the right product at the right time, we exam each candidate product for each user at a particular decision time. We propose to build an opportunity model to estimate the probability of a user purchasing the product at a particular time interval (i.e. $(y, y + \Delta t]$). That is the joint probability $P(p_product, T \in (y, y + \Delta t])$, where T is the purchasing time. Let $P(p_product)$ represent the probability of the user purchasing the product, and $P(T \in (y, y + \Delta t] | p_product)$ represent the conditional probability of the user purchasing the product at a particular time period conditioned on that the user will purchase the product.

Based on the chain rule, we have the joint probability:

$$\begin{aligned} P(p_product, T \in (y, y + \Delta t]) \\ = P(T \in (y, y + \Delta t] | p_product) P(p_product) \end{aligned} \quad (1)$$

We propose to adapt hazards models in survival analysis to estimate $P(T \in (y, y + \Delta t] | p_product)$, and adapt existing recommendation algorithms to estimate $P(p_product)$.

3.1 Hazards Model in Survival Analysis

Survival analysis is a heavily studied topic in statistics, which is named as duration modeling in economics or reliability analysis in engineering. One major part of survival analysis is to estimate the time of an event, such as when a machine fails to work or when a patient fails to survive. In the e-commerce domain, we can view the task of conditional opportunity model as predicting the time of the follow-up purchase event of the product. Follow-up purchases may include repurchases that happen regularly or new purchases that are triggered by the previous purchase. Given the similar nature of survival analysis and our e-task, we propose to use the hazards model in survival analysis to estimate $p(y | p_product)$ in this paper.

Let us review basic hazards models in survival analysis. Let $p(y)$ denote the density function of the time distribution of an event. Let y be a value of time and T be a random variable representing the event time. The cumulative distribution function is denoted as $P(y) = Pr(T \leq y)$ and survival function is denoted as $S(y) = Pr(T > y) = 1 - P(y)$. The hazards function is $h(y) = \frac{p(y)}{S(y)}$. It indicates the instantaneous potential per unit time for the event to occur at time y given that the event has not occurred up to time y .

The survival analysis further extends the model with covariates. Covariates are features that would affect the survival time. For example, if a product is on promotion, it might shorten the time before a user waits to purchase it. The covariates could include different types of features, including time independent variables (user age, gender, household income, product brand etc.), time dependent internal variables (time since last purchase of the same product, recent user search queries or clicks etc.) and time dependent external variables (global economy, seasonal index, day of the week, etc.). There are two common approaches to incorporate covariates \mathbf{x} (a vector of features) in a hazards model. The first approach is the *Cox proportional hazards model* [14]. It assumes that covariates are multiplicatively related to the hazards. The second approach is the *Accelerated life model* [22]. It assumes that covariates are multiplicatively related to the survival time. Research in statistics discovered that the Weibull distribution satisfies assumptions in both directions. The density function of the basic Weibull distribution is shown in Equation 2 [21].

$$p(y) = \gamma \theta y^{\gamma-1} \exp\{-\theta y^\gamma\} \quad (2)$$

where γ is sometimes called the shape parameter and held fixed. If $\gamma = 1$, the Weibull model reduces to the exponential model and the hazard is constant. If $\gamma > 1$, the hazard increases as time increases. If $\gamma < 1$, the hazard decreases over time. θ is the scale parameter and can be re-parameterized based on a regression parameter β and covariates \mathbf{x} as follows:

$$p(y) = \gamma \exp\{\beta^T \mathbf{x}\} y^{\gamma-1} \exp\{-\exp\{\beta^T \mathbf{x}\} y^\gamma\} \quad (3)$$

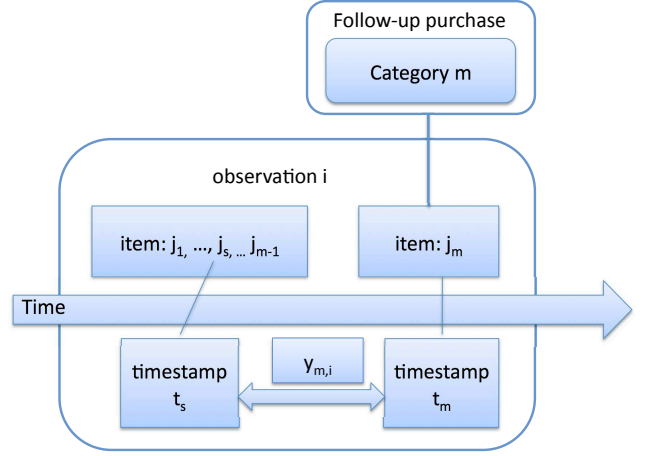


Figure 1: Illustration of the relationship of variables. The user first makes a series of product purchases before timestamp t_m . Then the user makes a purchase of item j_m in category m at timestamp t_m . This follow-up purchase in category m is the i^{th} observation in category m . Suppose that the purchase of j_m is triggered by item j_s at timestamp t_s . Purchase time $y_{m,i} = t_m - t_s$ is the time gap between t_m and t_s . An observation i is associated with two major variables: 1) purchase time $y_{m,i}$ and 2) covariates $\mathbf{x}_{m,i}$ (which is not shown in the figure).

where $\exp\{\beta^T \mathbf{x}\}$ represents the new scale parameter θ' . This density function represents the basic proportional hazards model that models the event time with associated covariates. The corresponding hazards function is simply $h(y) = \gamma \theta' y^{\gamma-1}$.

3.2 Notations

Here we describe notations in the e-commerce domain in this paper. The relationship between different variables is shown in Figure 1.

- $u = 1, 2, \dots, U$: the index of users.
- $m = 1, 2, \dots, M$: the index of item categories. In the e-commerce domain, it is the category of the product, such as *Apparel & Accessories*|*Swimwear*, *Baby*|*Car Seats*, *Sports and Fitness*|*Football*, etc.
- $j_m = 1_m, 2_m, \dots, J_m$: the index of items in category m . In this paper, an item is a product.
- t_m : the purchase timestamp of item j_m .
- Δt : the window size of the purchase time in consideration (such as 3 days or 1 hour)
- D : The observed data of all follow-up purchases from all users. Each category m has N_m follow-up purchase observation from all users. Each follow-up purchase observation $i = 1, \dots, N_m$ in category m is associated with the purchase time $y_{m,i}$ and covariates $\mathbf{x}_{m,i}$.
- $y_{m,i}$: the purchase time of the i^{th} observation in category m . $y_{m,i} = t_m - t_s$ is the time distance between

the user's purchase timestamp t_m of item j_m and the user's purchase timestamp t_s of the triggering item j_s .

- $\mathbf{x}_{m,i}$: the k -dimensional vector of covariates associated with the i^{th} observation in category m . Covariates could be associated with user u who makes the purchase, the user's purchase history j_1, \dots, j_{m-1} , the purchase item j_m , the global environment, etc.
- $P(p_product = yes)$: the probability of a user purchasing the product. The probability is calculated for each user-product pair.
- *Opportunity model*: density function/model to predict the joint purchase probability $p(y, p_product)$, i.e., the probability of a user purchasing of the product at the particular time.
- *Conditional opportunity model*: density function/model to predict the conditional time probability $p(y|p_product)$, i.e., the probability of a user purchasing the product at the particular time, given that the user will purchase the product.

3.3 Conditional Opportunity Model

In this paper we propose to adapt the Cox proportional hazards model to the e-commerce domain and use it to estimate $P(T \in (y, y + \Delta t] | p_product)$. To do that, we learn the *conditional opportunity model* $p(y|p_product)$, which is the density function/model of the purchasing time. We can either learn one model per product or one model per product category. Without loss of generality, we describe the model by assuming one model per category in this paper.

In the real world, a small number of categories are often purchased while most categories have few purchases. To solve the data sparsity issue, we follow a common practice and extend the conditional opportunity model with a hierarchical Bayesian framework as illustrated in Figure 2. This framework helps the category with few observations by borrowing information from other categories through a common prior for parameters of all proportional hazards models.

For each category m , β_m is sampled from a Gaussian distribution: $\beta_m \sim N(\mu_\beta, \Sigma_\beta)$ and γ_m is sampled from the Gamma distribution: $\gamma_m \sim \text{Gamma}(a_\gamma, b_\gamma)$. We denote $\phi = (\mu_\beta, \Sigma_\beta, \mathbf{a}_\gamma, \mathbf{b}_\gamma)$. For each i^{th} observation in category m with its observed covariates $\mathbf{x}_{m,i}$, its purchase time $y_{m,i}$ is sampled from the conditional opportunity model

$$p(y_{m,i} | \beta_m, \gamma_m) = \gamma_m \exp\{\beta_m^T \mathbf{x}_{m,i}\} y_{m,i}^{\gamma_m - 1} \exp\{-\exp\{\beta_m^T \mathbf{x}_{m,i}\} y_{m,i}^{\gamma_m}\} \quad (4)$$

Consider that data D consists of a series of observations from all categories. Purchase times in the observations are generated by using a set of hidden variables $\theta = \{\theta_1, \theta_2, \dots, \theta_M\}$ ($\theta_m = \{\beta_m, \gamma_m\}$). The likelihood can be written as a function of $\phi = (\mu_\beta, \Sigma_\beta, \mathbf{a}_\gamma, \mathbf{b}_\gamma)$. We use y_m to represent $\{y_{m,1}, \dots, y_{m,i}, \dots, y_{m,N_m}\}$, i.e., observation times in category m .

$$p(D|\phi) = \prod_{m=1}^M p(y_m|\phi) = \prod_{m=1}^M \int p(\theta_m, y_m|\phi) d\theta_m \quad (5)$$

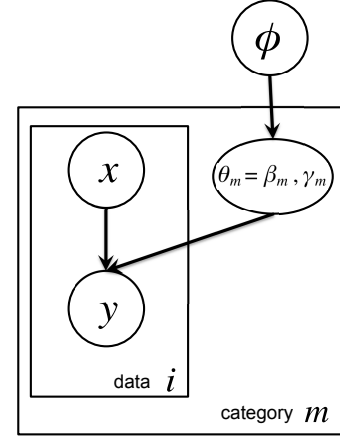


Figure 2: Illustration of dependencies of variables in the hierarchical conditional opportunity model. It shows the i^{th} observation of category m . $y_{m,i}$ is the purchase time which is dependent on the conditional opportunity model $\theta_m = \{\beta_m, \gamma_m\}$ of category m , as well as the observed covariates $\mathbf{x}_{m,i}$ of this purchase. Each category m has its own parameters of the conditional opportunity model β_m, γ_m . Models of each category share information through the prior, $\phi = (\mu_\beta, \Sigma_\beta, \mathbf{a}_\gamma, \mathbf{b}_\gamma)$.

3.4 Parameter Inference with Variational Bayesian

There is no closed-form solution for the estimation of the model parameters. We follow the variational Bayesian method[1] for constrained (approximate) optimization to derive an iterative process to find the approximate solution. Maximizing the likelihood in Equation 5 is equivalent to maximizing the log likelihood $L(\phi)$.

$$L(\phi) = \ln p(D|\phi) = \sum_{m=1}^M \ln p(y_m|\phi) = \sum_{m=1}^M \ln \int p(\theta_m, y_m|\phi) d\theta_m$$

We can simplify the problem by introducing an auxiliary distribution $q(\theta_m)$ for each hidden variable θ_m [1]. In the variational approach, we constrain $q(\theta_m)$ to be a particular tractable form for computational efficiency. In particular, we assume that $q(\beta_m) = N(\mu_{\beta_m}, \Sigma_{\beta_m})$ and $q(\gamma_m) = \text{Gamma}(a_{\gamma_m}, b_{\gamma_m})$. The process to infer parameters is to iterate between the following E-step and M-step until convergence.

3.4.1 E-Step

In the **E-step**, we infer the posterior distributions over hidden variables θ_m given the current parameter setting ϕ . There is no closed-form solution. Instead, we find a tractable approximation of the posterior distribution of θ_m given ϕ (i.e, $q(\theta_m)$ that maximizes $L(\phi)$).

$$\begin{aligned}
L(\phi) &= \sum_{m=1}^M \ln \int q(\theta_m) \frac{p(\theta_m, y_m | \phi)}{q(\theta_m)} d\theta_m \\
&\geq \sum_{m=1}^M \int q(\theta_m) \ln \frac{p(\theta_m, y_m | \phi)}{q(\theta_m)} d\theta_m \\
&= \sum_{m=1}^M \int q(\theta_m) \ln p(y_m | \phi) d\theta_m - \sum_{m=1}^M \int q(\theta_m) \ln \frac{q(\theta_m)}{p(\theta_m | y_m, \phi)} d\theta_m \\
&\equiv \mathbf{F}(q(\theta_1), \dots, q(\theta_M), \phi) \quad (6)
\end{aligned}$$

To maximize $L(\phi)$, it is the same as minimize the following equation to find each distribution $q(\theta_m)$:

$$q(\theta_m)^{(t)} = \arg \min_{q(\theta_m)} \int q(\theta_m) \ln \frac{q(\theta_m)}{p(\theta_m | y_m, \phi)} d\theta_m \quad (7)$$

Given that $p(\theta_m | y_m, \phi) = \frac{p(y_m | \theta_m) p(\theta_m | \phi)}{p(y_m | \phi)}$, we have:

$$q(\theta_m) = \arg \min_{q(\theta_m)} KL[q(\theta_m) || p(\theta_m | \phi)] - \int q(\theta_m) \ln p(y_m | \theta_m) d\theta_m \quad (8)$$

The first part in Equation 8 is the KL-divergence between the posterior distribution $q(\theta_m)$ and the prior distribution $p(\theta_m | \phi)$.

$$KL[q(\theta_m) || p(\theta_m | \phi)] = KL[q(\beta_m) || p(\beta_m | \phi)] + KL[q(\gamma_m) || p(\gamma_m | \phi)]$$

The KL-divergence between two Gaussian distributions is

$$\begin{aligned}
&KL[q(\beta_m) || p(\beta_m | \phi)] \\
&= \frac{1}{2} [tr(\Sigma_\beta^{-1} \Sigma_{\beta_m}) + (\mu_\beta - \mu_{\beta_m})^T \Sigma_\beta^{-1} (\mu_\beta - \mu_{\beta_m}) \\
&\quad - \ln(\frac{\det \Sigma_{\beta_m}}{\det \Sigma_\beta}) - k]
\end{aligned}$$

The KL-divergence between two gamma distributions is

$$\begin{aligned}
&KL[q(\gamma_m) || p(\gamma_m | \phi)] \\
&= (a_{\gamma_m} - a_\gamma) \Psi(a_{\gamma_m}) - \log \Gamma(a_{\gamma_m}) + \log \Gamma(a_\gamma) \\
&\quad + a_\gamma (\log b_{\gamma_m} - \log b_\gamma) + a_{\gamma_m} \frac{b_\gamma - b_{\gamma_m}}{b_{\gamma_m}}
\end{aligned}$$

where $\Psi(\cdot)$ is the digamma function.

The second part in Equation 8 is to maximize the data likelihood of $y_m = \{y_{m,1}, \dots, y_{m,i}, \dots, y_{m,N_m}\}$ with the current $\theta_m = \{\beta_m, \gamma_m\}$.

$$\begin{aligned}
&\sum_{i=1}^{N_m} \int q(\theta_m) \ln p(y_{m,i} | \theta_m) d\theta_m \quad (9) \\
&= \sum_{i=1}^{N_m} \int \int q(\beta_m) q(\gamma_m) \ln p(y_{m,i} | \beta_m, \gamma_m) d\beta_m d\gamma_m \\
&= \sum_{i=1}^{N_m} [E(\ln \gamma_m) + E(\beta_m)^T \mathbf{x}_{m,i} + (E(\gamma_m) - 1) \ln y_{m,i} \\
&\quad - E(\exp(\beta_m^T \mathbf{x}_{m,i})) E(y_{m,i}^{\gamma_m})]
\end{aligned}$$

The expectations in the above equation are

$$\begin{aligned}
E(\ln \gamma_m) &= \Psi(a_{\gamma_m}) - \ln(b_{\gamma_m}) \\
E(\beta_m) &= \mu_{\beta_m} \\
E(\gamma_m) &= \frac{a_{\gamma_m}}{b_{\gamma_m}} \\
E(\exp(\beta_m^T \mathbf{x}_{m,i})) &= \exp\{\mu_{\beta_m}^T \mathbf{x}_{m,i} + \frac{1}{2} \mathbf{x}_{m,i}^T \Sigma_{\beta_m} \mathbf{x}_{m,i}\} \\
E(y_{m,i}^{\gamma_m}) &= \frac{b_{\gamma_m}^{a_{\gamma_m}}}{(b_{\gamma_m} - \ln y_{m,i})^{a_{\gamma_m}}}
\end{aligned}$$

We can combine the above derivations with Equation 8, then find $q(\theta_m)$ using the conjugate gradient descent method.

3.4.2 M-Step

In the **M-step**, the goal is to maximize $\mathbf{F}(q(\theta_1), \dots, q(\theta_M), \phi)$ in Equation 6 with respect to ϕ given all θ_m . It is same as to maximize the following quantity:

$$\phi^{(t+1)} \leftarrow \arg \max_{\phi} \sum_{m=1}^M \int q(\theta_m) \ln p(y_m | \phi) d\theta_m \quad (10)$$

The optimal ϕ at this step can be estimated with the following closed form:

$$\begin{aligned}
\mu_\beta &= \frac{\sum_{m=1}^M \mu_{\beta_m}}{M} \\
\Sigma_\beta &= \frac{\sum_{m=1}^M [\Sigma_{\beta_m} + (\mu_{\beta_m} - \mu_\beta)(\mu_{\beta_m} - \mu_\beta)^T]}{M} \\
a_\gamma &= \Psi^{-1}(\ln b_\gamma + \frac{\sum_{m=1}^M \Psi(a_{\gamma_m}) - \ln(b_{\gamma_m})}{M}) \\
b_\gamma &= \frac{M a_\gamma}{\sum_{m=1}^M \frac{a_{\gamma_m}}{b_{\gamma_m}}}
\end{aligned}$$

where $\Psi^{-1}(\cdot)$ is the inverse digamma function.

3.5 Joint Purchase Probability

At a time y , we can decide whether to recommendation a particular product in category m to a particular user or not based on the following estimation: whether a user is likely to purchase the product in the near future (i.e. between time y and time $y + \Delta t$), given that the user has not purchased the product since a triggering time point. To do so, we need to estimate $P(p_product = yes, T \in (y, y + \Delta t]) = P(p_product = yes)P(y < T \leq y + \Delta t | T > y, p_product = yes)$, where

$$\begin{aligned}
&P(y < T \leq y + \Delta t | T > y, p_product = yes) \\
&= \frac{P(T \leq y + \Delta t) - P(T \leq y)}{1 - P(T \leq y)} \quad (11)
\end{aligned}$$

where:

$$\begin{aligned}
&P(T \leq y) \quad (12) \\
&= 1 - E_{\beta_m, \gamma_m} [\exp\{-y^{\gamma_m} \exp\{\beta_m^T \mathbf{x}_i\}\}] \\
&\approx 1 - \exp\{E_{\beta_m, \gamma_m} [-y^{\gamma_m} \exp\{\beta_m^T \mathbf{x}_i\}]\} \\
&= 1 - \exp\{E_{\gamma_m} [-y^{\gamma_m}] E_{\beta_m} [\exp\{\beta_m^T \mathbf{x}_i\}]\} \\
&= 1 - \exp\{-\frac{b_{\gamma_m}^{a_{\gamma_m}}}{(b_{\gamma_m} - \ln y)^{a_{\gamma_m}}} \exp\{\mu_{\beta_m}^T x_i + \frac{1}{2} x_i^T \Sigma_{\beta_m} \mathbf{x}_i\}\}
\end{aligned}$$

The approximation is used because there is no closed-form solution of the integration for calculating the expectation.

To estimate $P(p_product = yes)$, i.e., the probability of the user purchasing the product, we can use any existing recommender systems through the logistic regression model:

$$P(p_product = yes) = \frac{1}{1 + e^{-\mathbf{f}^T \mathbf{x}}}$$

where \mathbf{x} is a vector of features that are associated with the purchasing, which includes the score/output of the existing recommender system (such as SVD), as well as other features that might help to predict the user's purchase probability. \mathbf{f} is a vector of coefficients that can be learnt by maximizing the likelihood of the training data.

3.6 Implementation Details

The purchase time y_m is determined by the purchase times-tamp of product j_m and that of the triggering product j_s . The triggering product is not necessarily the product in the most recent purchase. There are multiple heuristic approaches to find the triggering item j_s in the user's purchase history. Here we leverage the transition probability $P(j_s, j_m)$ in Equation 13.

$$P(j_a, j_b) = \frac{\#(j_a, j_b) + \lambda}{\sum \#(j_a, *) + J \cdot \lambda} \quad (13)$$

where $\#(j_a, j_b)$ is the number of follow-up purchases of j_b that happened after j_a . $\sum \#(j_a, *)$ is the number of follow-up purchases after j_a . J is the number of products and λ is the smoothing factor, which is set as 0.1.

We first rank all items $\{j_1, \dots, j_s, \dots, j_{m-1}\}$ that happened less than k_t days before j_m and have $P(j_s, j_m) > Thres_{tran}$. Then we use the top one as the triggering item. Other approaches can be explored in the future work. Although other purchases $\{j_1, \dots, j_{s-1}, j_{s+1}, \dots, j_{m-1}\}$ in the user history are not treated as the triggering item, they are incorporated into the opportunity model as covariates $\mathbf{x}_{m,i}$.

In this paper, we use the following covariates for each purchase of product j_m made by user u at timestamp t_m : whether user u purchased any product in category m in time bin t_{b1}, \dots, t_{bk} ; how many times the user u purchased any product in category m in time bin t_{b1}, \dots, t_{bk} ; whether the user purchased the product j_m in time bin t_{b1}, \dots, t_{bk} ; how many times the user u purchased the product j_m in time bin t_{b1}, \dots, t_{bk} ; which season t_m is in, whether t_m is in the holiday season, etc. Time bins t_{b1}, \dots, t_{bk} are set as one day, one week, one month, two months, three months, six months, one year, etc. We choose these covariates to show the effect of incorporating covariates in the conditional opportunity model. These covariates capture the change of the time distribution with the user's purchase history, the seasonal change, the cyclic pattern, etc. All covariates are normalized in the scale of $[0, 1]$.

4. EVALUATION METHODOLOGY

As we are studying a new problem of recommending the right product at the right time, there is no standard evaluation methodology. We design various experiments to evaluate the performance of the opportunity model. Major research questions that we aim to answer are:

Predictability of the conditional opportunity model

How accurate is the conditional time probability that is predicted by the conditional opportunity model? How accurate is the predicted purchase time, compared to the actual purchase time? Are covariates useful?

Predictability of the opportunity model Is the joint purchase probability a good signal of making recommendations? Does it generate a better ranking in traditional zero-query pull-based recommendation systems? Can it help to improve the user satisfaction/utility in proactive push-based recommendation systems?

4.1 Evaluation of Conditional Opportunity Model

4.1.1 Metrics

It is a relatively new research topic to predict the purchase time and evaluate the performance of such model. We introduce the following two metrics to evaluate the performance of the conditional opportunity model.

The first metric is the perplexity of the model. It is motivated by the perplexity metric used to evaluate language models and speech recognition [2]. The perplexity measures how well a model predicts the testing data. It is defined as follows:

$$\begin{aligned} perplexity &= \left(\prod_{m=1}^M \prod_{i=1}^{N_m} \frac{1}{P(T \in (y, y + \Delta t] | p_product)} \right)^{\frac{1}{\sum_{m=1}^M N_m}} \\ &= 2^{-\frac{\sum_{m=1}^M \sum_{i=1}^{N_m} \frac{1}{\sum_{m=1}^M N_m} \log_2 P(T \in (y, y + \Delta t] | p_product)}} \end{aligned} \quad (14)$$

where $P(T \in (y, y + \Delta t] | p_product)$ is defined in Equation 11, and i is the index for a testing data point. A better model tend to give a higher data likelihood to the actual follow-up purchase time in the testing data, thus they have lower *perplexity*, which means they are less surprised by the testing data.

The second metric focuses on the difference between the estimated time $y_{m,i}$ and the actual time $\hat{y}_{m,i}$. After a model predicts the distribution $p(y_{m,i} | p_purchase)$ of the purchase time, we use the median of the distribution as the estimated purchase time $\hat{y}_{m,i}$. The median of the distribution is $\exp(\beta_m \mathbf{x}_{m,i})^{-\frac{1}{r_m}} (\ln 2)^{\frac{1}{r_m}}$. The error across all testing data can then be used to analyze and compare. The smaller the error, the better the model. Here we use three types of errors: mean absolute error(MAE), mean squared error(MSE) and mean absolute percentage error(MAPE = $\frac{1}{\sum_{m=1}^M N_m} \sum_{m=1}^M \sum_{i=1}^{N_m} \frac{|\hat{y}_{m,i} - y_{m,i}|}{y_{m,i}}$).

4.1.2 Conditional Opportunity Models to Compare

In our experiments, we compare the following four conditional opportunity models.

Uniform assigns a uniform distribution to all time. $p(y_{m,i} | p_purchase) = \frac{1}{T_{uniform}}$ where $T_{uniform}$ is the maximum time in consideration. We set $T_{uniform} = 500$, i.e., 500 days, arbitrarily in this paper.

O-One is a conditional opportunity model that fits a **single** set of parameters with no covariates (i.e., a basic Weibull distribution) to the purchase data. All follow-up purchases in different category m use the same model.

O-Dest is a hierarchical conditional opportunity model that fits one Weibull distribution per product category, with no covariates.

O-DestCov further incorporates covariates into *O-Dest*. In this case, the probability density function changes for each user and product as values of the associated covariates change over time.

All models smooth the conditional probability estimation $P(T \in (y, y + \Delta t] | p_product) = \max(P(\minThres, T \in (y,$

Table 1: The utility set of the recommender system. There are four types of utilities, depending on whether the system recommends the item to the user and whether the user purchases the item.

	show:Y	show:N
accept:Y	u_{TP}	u_{FN}
accept:N	u_{FP}	u_{TN}

$y + \Delta t | p_product$)) to avoid having a probability that is too low (such as when there is no triggering item before the purchase). minThres is set as 0.001. Δt in Equation 11 is set as 7 (i.e., 7 days) for all models during the prediction step. The threshold for the transition probability to consider the an item j_s as the triggering item is set as 0.01.

4.2 Evaluation of Opportunity Model

4.2.1 Metrics

Here we evaluate the predictability of the opportunity model with the joint purchase probability in two scenarios.

1) **[Zero-query pull-based scenario]** assumes the user comes to the site to look for products to purchase without issuing any search query. In this scenario, the goal is to discover products that the user would purchase and recommend them to the user proactively. The system ranks all products by their joint purchase probability and recommends top K products to the user.

The evaluation metric in this scenario is the conversion rate which reflects whether a user receives at least one good recommendation. Each testing time corresponds to the time when a user comes to the site and makes purchases. Let $S_{purchased}$ contain all unique products in the order. Let $C_{purchased}$ contain all unique categories in the order. Let $S_{K, recommended}$ contain top K unique product recommendations. Let $C_{K, recommended}$ top K unique category recommendations. $CR_{product}$ is the conversion rate at the product level and $CR_{category}$ is the conversion rate at the category level. Significance level of 0.05 with the paired two-tailed t -test is used to compare two models.

$$CR_{product}@K = \begin{cases} 1 & S_{purchased} \cap S_{K, recommended} \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

$$CR_{category}@K = \begin{cases} 1 & C_{purchased} \cap C_{K, recommended} \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

2) **[Push-based email promotion scenario]** assumes that recommender systems send email/message proactively to a user regularly regardless of whether the user comes to the site or not.

The evaluation metric in this scenario is the average utility/user satisfaction. The utility for each type of recommendations is shown in Table 1. The $utility_g$ for each email of recommendations is calculated by Equation 15.

$$utility_g = \sum (u_{TP}I_{show, accept} + u_{FP}I_{show, accept} + u_{FN}I_{show, accept} + u_{TN}I_{show, accept}) \quad (15)$$

I_* is the indicator function where $I_* = 1$ if $*$ is true. Unlike traditional metrics such as conversion rate@K, the utility metric considers both the positive effect for *good* recommendations and the negative effect for *bad* recommendations. The higher the utility, the better the model.

To achieve a better utility, the system with a filtering component should send emails only if the expected utility of adding the product is higher than zero (i.e. the joint probability is higher than the threshold). The recommendation threshold is automatically determined by the following equation [4]: $\frac{u_{FP} - u_{TN}}{u_{FP} - u_{TN} + u_{FN} - u_{TP}}$.

We don't have a real push-based email promotion system for a user study. Instead, we create an evaluation dataset with the purchase data from a pull-based e-commerce website. The goal is to evaluate each model's predictability of discovering the "real opportunity" and avoiding the "fake opportunity" in the email marketing. For each purchase at time t in the testing dataset, we let each model consider two opportunities: sending a recommendation email the weekend right before t and sending an email on some other random weekend before t . Since existing models can not tell whether to send an email or not, they always send an email with top K recommendations for each opportunity. The opportunity model with a filtering component will add a product to the email if the expected utility of adding the product is higher than zero. If no product is added to the email, the opportunity model would skip this opportunity and do not send an email. Recommendations in the email are compared with actual products that the user purchases in the week after the email is sent. Assume that there are G opportunities to send recommendation emails in the testing period. The average utility/user satisfaction *utility* can be calculated as following: $utility = \frac{\sum_{g=1}^G utility_g}{G}$.

4.2.2 Recommendation Models to Compare

We choose the following recommendation models to compare:

TopPop recommends most popular products to the user.

SVD is a widely-used recommendation algorithm with a decent performance on the well-known Netflix competition. It is the basis of several recommendation algorithms based on latent factors.

SVD.util is the state-of-art recommendation algorithm in the e-commerce domain without time consideration. It modifies SVD with the marginal net utility framework [19].

Regression.Model is an alternative new approach to predict the probability of a user purchasing a product at a particular time using a logistic regression model, with various important time-dependent features. This model contains features such as the SVD score of the product, whether the product/category has been purchased t days ago, how many time the product/category has been purchased t days ago, etc.

Conditional.Opportunity.Model recommends products based on $P(T \in (y, y + \Delta t])$, which is estimated by the hierarchical conditional opportunity model with covariates.

Opportunity.Model recommends products based on the joint probability of a user making a purchase of a product in the near future ($P(p_product, T \in (y, y + \Delta t])$).

Opportunity.Model.Filtering adds a filtering component to *Opportunity.Model*. In the push-based email scenario, it adds a product to the promotion email only

if the expected utility of adding the product is higher than zero.

4.3 Dataset

The purchase history from 2004-01-01 to 2009-03-08 collected on a real-world e-commerce website, shop.com, is used for our experiments. We use all purchases that have category information of the product. Tail users that made less than 5 product purchases are filtered out in the training data, which follows the similar pre-processing in related work [19, 24]. In addition, 10 possible spam users that made more than 200 product purchases are filtered out as well. The remaining data contains 11,351 users and 67,291 products. There are 105,550 unique (user, product) pairs. This user-product matrix is quite sparse, with only 0.014% density. There are 380 categories in total.

To evaluate the conditional opportunity model as in Section 4.1, 10-fold cross validation is used. To evaluate the opportunity model with the joint purchase probability as in Section 4.2, we sort all purchase history by time. The first 90% is used as the training data (data before 2008-11-24) and the last 10% is used as the testing data (data after 2008-11-24). There are 7,014 testing cases in total. At the product level, there are 1,143 repurchase cases(16.29%) and 6,269 new purchase cases(89.37%). At the categorical level, there are 2,850 repurchase cases(i.e. 40.63% cases with purchase from the same category) and 4,850 new purchase cases(i.e. 69.14% cases with purchase from a new category).

To train the regression model with SVD as one of the features, the first half training data is used to train the SVD model and the second half training data is used to learn coefficients in the regression model. The number of latent factors for all SVD-related models is set to 50. In the recommendation step, all models recommend top K ($K=5$) products to the user. To save the computation time, *Regression.Model*, *Conditional.Opportunity.Model* and *Opportunity.Model* rank among top N recommendations from SVD and selects top K to recommend, where $N = 100$ and $K = 5$.

The time density plots of some common transitions from products in the *Baby|Feeding* category to the target item j_m are shown in Figure 3. This supports our motivation of identifying triggering items in the user history. For example, users who purchased from *Baby|Feeding* would purchase items in other *Baby* related categories in the future. The purchase time of the follow-up purchase does follow different distributions for different products or different covariates (take the triggering product as a covariate). For example, users who purchased from the *Baby|Feeding* category would purchase products from this category again in one month. Later on when the baby grows up, they would purchase products from *Toys|Board, Card&Dice Games*.

5. EXPERIMENTAL RESULTS

5.1 Analysis of Conditional Opportunity Model

The perplexity of all conditional opportunity models in Section 4.1.2 are compared in Table 2. All conditional opportunity models have lower perplexity than the baseline model *Uniform*. This demonstrates that conditional opportunity models have better predictability of the time-based purchase probability in the data. Among all conditional op-

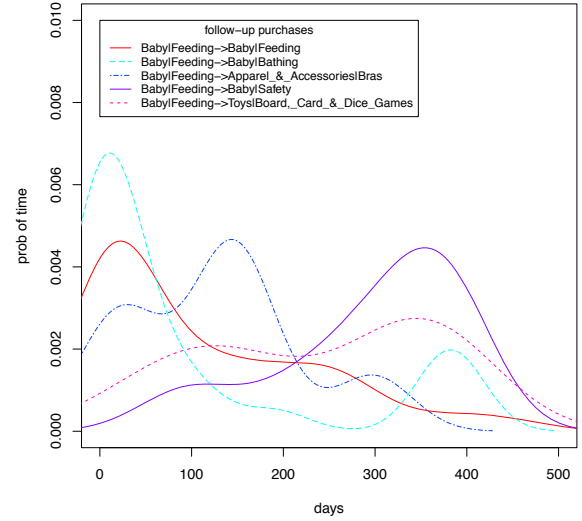


Figure 3: Density plot of the purchase time between different follow-up purchases from Baby|Feeding

Table 2: Perplexity of different conditional opportunity models in 10-fold cross validation.

Data	Uniform	O-One	O-Dest	O-DestCov
All purchases	46.95	23.75	22.45	18.95
repurchases	58.99	16.68	14.80	9.75
new purchases	44.25	26.04	25.01	22.48

portunity models, *O-DestCov* achieves the lowest perplexity, followed by *O-Dest*, and *O-One*. *O-DestCov* incorporates related covariates in addition to fitting parameters of a conditional opportunity model for each category m . It shows the importance of considering covariates when modeling the purchase time of a follow-up purchase.

To further analyze the effect of covariates, we compare the perplexity of all models in the repurchase data and the new purchase data in Table 2. As expected, the major gain is in reducing the perplexity of repurchases. More feature exploration would be useful to improve the prediction of new purchases in the future.

Now we compare the estimated purchase time with the actual time of a follow-up purchase in Table 3. All conditional opportunity models perform better than the baseline model *Uniform*. *O-DestCov* gives the most accurate estimation of the purchase time, followed by *O-Dest*, and *O-One*. According to MAE, the estimated purchase time from *O-DestCov*

Table 3: Error between the estimated purchase time and the actual purchase time. MAE stands for mean absolute error. MSE stands for mean squared error. MAPE stands for mean absolute percentage error.

Data	Error rate	Uniform	O-One	O-Dest	O-DestCov
All Purchases	MAE	159.74	105.279	102.617	77.695
	MSE	30859.346	23322.373	21920.462	14335.505
	MAPE	16.592	3.796	3.763	2.434
Repurchases	MAE	183.337	63.7	62.119	34.536
	MSE	37728.853	9406.591	8799.832	4456.393
	MAPE	27.895	6.165	5.631	0.658
New Purchases	MAE	154.244	114.956	112.042	87.736
	MSE	29260.204	26560.746	24973.753	16634.239
	MAPE	13.964	3.246	3.329	2.847

Table 4: Conversion rate of recommendation models in the zero-query pull-based scenario. Numbers in bold are significantly better than the corresponding value in baseline models.

Product level						
Model	All purchases CR@1 CR@5		Repurchases CR@1 CR@5		New purchases CR@1 CR@5	
TopPop	0.0	0.0155	0.0	0.0936	0.0	0.0003
SVD	0.0134	0.0344	0.0822	0.2073	0.0	0.0011
SVD.util	0.0287	0.0369	0.1750	0.2248	0.0002	0.0003
Regression.Model	0.0339	0.0525	0.2082	0.3202	0.0	0.0003
Conditional.Opportunity.Model	0.0160	0.0282	0.0962	0.1706	0.0003	0.0006
Opportunity.Model	0.0289	0.0452	0.1750	0.2712	0.0005	0.0011
Category level						
Model	All purchases CR@1 CR@5		Repurchases CR@1 CR@5		New purchases CR@1 CR@5	
TopPop	0.0204	0.0848	0.0130	0.1382	0.0219	0.0435
SVD	0.0577	0.1688	0.1309	0.3267	0.0066	0.0540
SVD.util	0.0751	0.1344	0.1779	0.2453	0.0041	0.0511
Regression.Model	0.0999	0.1792	0.2330	0.3646	0.0076	0.0472
Conditional.Opportunity.Model	0.1102	0.2154	0.2467	0.3856	0.0144	0.0907
Opportunity.Model	0.1313	0.2345	0.3091	0.4495	0.0082	0.0816

is 77 days away(higher or lower) from the actual time. It is a decent performance given that the total time range is 500 days. Further analysis shows that *O-DestCov* predicts more accurately in both the repurchase and the new purchase data.

5.2 Analysis of Opportunity Model

5.2.1 Pull-based Scenario

Now we evaluate the performance of the opportunity model with the joint purchase probability. The conversion rate of different recommendation models in the zero-query pull-based scenario is compared in Table 4.

The opportunity model achieves higher conversion rate at both the product level and the category level. Further analysis shows that the contribution of the conditional opportunity model is mainly at the category level. It is not surprising because the model is designed at the category level. Product-level models can be explored in the future. The key challenge is to solve the sparsity and scalability issues.

We further analyze the performance of the conversion rate in different purchase scenarios. In Table 4, we show the conversion rate of repurchases and new purchases. It is clear that the major contribution of the opportunity model is in the repurchase scenario. The observation is consistent with the evaluation of the conditional opportunity model in Section 5.1: the conditional opportunity model predicts more accurately in repurchases.

5.2.2 Push-Based Scenario

Now we evaluate all models in the push-based scenario. In the *Opportunity.Model.Filtering* model, a product is recommended only if its joint purchase probability is higher than the threshold. We evaluate the model with three sets of utility at both the product level (i.e. true positive means that the user purchased the exact product) and the category level (i.e. a true positive means that the user purchased a product in same category as the recommendation). The results are shown in Table 5. There are 3,014 email opportunities in the testing period. 55% of them have follow-up

Table 5: Average utility of email recommendations in the push-based scenario. *Coverage* is the percentage of emails that are sent among all possible opportunities. *Rec count* is the average number of unique recommendations in a email when the email is sent.

utility set 1 $u_{TP} = 3, u_{FN} = 0$ $u_{FP} = -1, u_{TN} = 0$ Threshold = 0.25					
Product level					
Model	TopPop	SVD	SVD.util	Opportunity.Model	Opportunity.Model Filtering
Utility	-4.994	-4.862	-4.768	-4.810	-0.048
Coverage	1	1	1	1	0.023
Rec count	5	5	5	5	2.432
category level					
Model	TopPop	SVD	SVD.util	Opportunity.Model	Opportunity.Model Filtering
Utility	-4.823	-4.515	-4.570	-4.255	0.006
Coverage	1	1	1	1	0.023
Rec count	5	5	5	5	1.081
utility set 2 $u_{TP} = 10, u_{FN} = 0$ $u_{FP} = -1, u_{TN} = 0$ Threshold = 0.09					
Product level					
Model	TopPop	SVD	SVD.util	Opportunity.Model	Opportunity.Model Filtering
Utility	-4.983	-4.621	-4.363	-4.477	-0.399
Coverage	1	1	1	1	0.180
Rec count	5	5	5	5	3.007
Category level					
Model	TopPop	SVD	SVD.util	Opportunity.Model	Opportunity.Model Filtering
Utility	-4.514	-3.667	-3.819	-2.951	0.282
Coverage	1	1	1	1	0.180
Rec count	5	5	5	5	1.880
utility set 3 $u_{TP} = 3, u_{FN} = -1$ $u_{FP} = -1, u_{TN} = 0$ Threshold = 0.20					
Product level					
Model	TopPop	SVD	SVD.util	Opportunity.Model	Opportunity.Model Filtering
Utility	-4.992	-4.828	-4.710	-4.762	-0.066
Coverage	1	1	1	1	0.033
Rec count	5	5	5	5	2.356
Category level					
Model	TopPop	SVD	SVD.util	Opportunity.Model	Opportunity.Model Filtering
Utility	-4.779	-4.394	-4.463	-4.069	0.015
Coverage	1	1	1	1	0.033
Rec count	5	5	5	5	1.115

purchases in the following week. Among those with follow-up purchases, the average number of purchases is 2.227.

In the *first* utility set, the utility u_{TP} for a good recommendation that the user purchases is set to 3. When the system shows a product and the user does not purchase it, the utility u_{FP} is -1 . The resulting filtering threshold is 0.25. In this case, *Opportunity.Model.Filtering* has a better utility than other models by filtering out many false alarms. The average utility at the category level is higher, which is expected. It rewards a recommendation if it matches the user's purchase at the category level, which is an easier task. In general, the model with higher conversion rate in the pull-based scenario has high utility in the push-based scenario.

In the *second* utility set, the utility u_{TP} is set to 10, indicating that the user is more tolerant to **bad** recommendations. Thus the threshold is lower, being 0.09. The coverage of *Opportunity.Model.Filtering* increases while the average utility drops. There is a tradeoff between the utility and the recommendation coverage, which can be tuned in a real-world application by a user or the system designer.

In the *third* utility set, u_{FN} is set to -1 . It is the penalty for not recommending a product that would be purchased

by the user in the following week. Thus the coverage of *Opportunity.Model.Filtering* is higher to avoid missing good recommendations with the threshold being 0.02. It still achieves the highest utility among all models.

6. CONCLUSION AND FUTURE WORK

In this paper, we propose to develop the *opportunity model* to predict the probability of a user purchasing a product at a particular time. This is achieved by modeling the joint probability of time and product, which can be calculated by combining a proportional hazards model and a logistic regression model. The joint probability guides the system to make the right product recommendation at the right time.

This is just the first step to tackle the research problem of capturing the time-based recommendation opportunity. Besides Weibull distribution, other distributions such as Exponential, Log-logistic, Lognormal, Generalized gamma, can be explored. Further improvements could be achieved with better covariates. More systematic approaches of discovering the triggering item could be explored. It would be helpful to carry out experiments with a real push-based system to evaluate the effectiveness of the opportunity model in a real user study. Due to the limited number of research data available and our own computational limits, we only learn the category-level conditional opportunity model, which clearly work well at the category level prediction. Another follow-up work is to evaluate this approach on a large-scale system to learn the product-level models, which might lead to better prediction at the product level.

Acknowledgments

We would like to thank shop.com for sharing the data. This work was funded by National Science Foundation IIS-0713111 and IIS-0953908. Any opinions, findings, conclusions or recommendations expressed in this paper are the authors, and do not necessarily reflect those of the sponsors.

7. REFERENCES

- [1] M. Beal. *Variational algorithms for approximate Bayesian inference*. PhD thesis, University of London, 2003.
- [2] S. Chen, D. Beeferman, and R. Rosenfeld. Evaluation metrics for language models. In *DARPA Broadcast News Transcription and Understanding Workshop (BNTUW)*, Lansdowne, Virginia, USA, Feb. 1998.
- [3] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the 4th ACM Recommender systems*, pages 39–46, 2010.
- [4] C. Elkan. The foundations of cost-sensitive learning. In *Proceedings of the 17th international joint conference on Artificial intelligence - Volume 2, IJCAI'01*, pages 973–978, 2001.
- [5] N. Golbandi, Y. Koren, and R. Lempel. Adaptive bootstrapping of recommender systems using decision trees. In *Proceedings of the fourth ACM WSDM'11*.
- [6] Z. Huang, W. Chung, and H. Chen. A graph model for e-commerce recommender systems. *J. Am. Soc. Inf. Sci. Technol.*, 55:259–274, February 2004.
- [7] R. Jin, L. Si, C. Zhai, and J. Callan. Collaborative filtering with decoupled models for preferences and ratings. In *Proceedings of the twelfth CIKM*, pages 309–316, New York, NY, USA, 2003. ACM.
- [8] Y. S. Kim, B.-J. Yum, J. Song, and S. M. Kim. Development of a recommender system based on navigational and behavioral patterns of customers in e-commerce sites. *Expert Syst. Appl.*, 28:381–393, February 2005.
- [9] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceeding of the 14th ACM SIGKDD KDD'08*.
- [10] Y. Koren. Collaborative filtering with temporal dynamics. In *KDD*, 2009.
- [11] S. li Huang. Designing utility-based recommender systems for e-commerce: Evaluation of preference-elicitation methods. *Electronic Commerce Research and Applications*.
- [12] R. J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *DL '00*, pages 195–204, 2000.
- [13] D. Parra-Santander and P. Brusilovsky. Improving collaborative filtering in social tagging systems for the recommendation of scientific articles. *Web Intelligence and Intelligent Agent Technology*, 1:136–142, 2010.
- [14] C. D. R. Regression models and life tables. *Journal of the Royal Statistic Society*, B(34):187–202, 1972.
- [15] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th WWW*.
- [16] G. Shani, D. Heckerman, and R. I. Brafman. An mdp-based recommender system. *J. Mach. Learn. Res.*, 6:1265–1295, 2005.
- [17] E. Shmueli, A. Kagian, Y. Koren, and R. Lempel. Care to comment?: recommendations for commenting on news stories. In *Proceedings of the 21st international conference on World Wide Web, WWW '12*, pages 429–438, New York, NY, USA, 2012. ACM.
- [18] J. Wang, B. Sarwar, and N. Sundaresan. Utilizing related products for post-purchase recommendation in e-commerce. In *Proceedings of the 5th ACM Recommender systems*.
- [19] J. Wang and Y. Zhang. Utilizing marginal net utility for recommendation in e-commerce. In *Proceedings of the 34th ACM SIGIR'11*, pages 1003–1012, 2011.
- [20] J. Wang, Y. Zhang, and T. Chen. Unified recommendation and search in e-commerce. In *Information Retrieval Technology*, pages 296–305. Springer Berlin Heidelberg, 2012.
- [21] J. Wang, Y. Zhang, C. Posse, and A. Bhasin. Is it time for a career switch? *Proceedings of the 22nd International World Wide Web Conference*, 2013.
- [22] L. J. Wei. The accelerated failure time model: A useful alternative to the cox regression model in survival analysis. *Statistics in Medicine*, 11(14-15):1871–1879, 1992.
- [23] L. Xiang, Q. Yuan, S. Zhao, L. Chen, X. Zhang, Q. Yang, and J. Sun. Temporal recommendation on graphs via long- and short-term preference fusion. In *Proceedings of the 16th ACM SIGKDD KDD'10*.
- [24] G. Zhao, M. L. Lee, W. Hsu, and W. Chen. Increasing temporal diversity with purchase intervals. In *Proceedings of the 35th ACM SIGIR*.