Utilizing Related Products for Post-Purchase Recommendation in E-commerce

Jian Wang
Department of Computer Science
University of California, Santa Cruz
Santa Cruz, CA 95060 USA
jwang30@soe.ucsc.edu

Badrul Sarwar, Neel Sundaresan
eBay Research Labs
San Jose, CA 95125 USA
{bsarwar, nsundaresan}@ebay.com

ABSTRACT

In this paper, we design a recommender system for the post-purchase stage, i.e., after a user purchases a product. Our method combines both behavioral and content aspects of recommendations. We first find the most related categories for the active product in the post-purchase stage. Among these related categories, products with high behavioral relevance and content relevance are recommended to the user. In addition, our algorithm considers the temporal factor, i.e., the purchase time of the active product and the recommendation time. We apply our algorithm on a random sample of the purchase data from eBay. Comparing to the baseline item-based collaborative filtering approach, our hybrid recommender system achieves significant coverage and purchase rate gain for different time windows.

Categories and Subject Descriptors

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

General Terms

Algorithms, Design, Experimentation

Keywords

Recommender System, Post-Purchase, Temporal Aspects

1. INTRODUCTION

Large e-commerce sites most often offer millions of products for consumers to choose from. Selecting the right product among this vast collection becomes a challenging task for consumers. Recommender systems have emerged to tackle this challenge and since their inception, they have been essential parts of modern e-commerce sites. Traditional recommender systems often use collaborative filtering [2, 9] based algorithms to analyze user preferences to build a predictive model for future recommendations. While these models are shown to successfully predict future user behavior, most often they ignore user's context information and temporal factors. In different stages of a user session, users' preference of recommendation can change considerably. When a user just logs in, the system needs to know his/her preference and purchase intention. Once the user starts focusing on a particular set of products, the system could recommend similar products for him/her to compare against. As soon as a purchase is made, similar product recommendation becomes less meaningful for the user. Instead, a related product recommendation such as an accessory to the product that was just purchased, would be more interesting and relevant.

We focus on designing the recommender system for the latter stage, denoted as **"post-purchase"** stage, that starts directly after a user purchases some product. The design of such recommender system poses three important challenges: **a)** relevance of recommendation **b)** recommendation coverage, and **c)** time sensitivity. To address the **relevance** challenge, we devise an algorithm that combines the behavioral and content relevance scores. The behavioral relevance is computed by using the user-product purchase data. The content-based relevance is computed by comparing products' key terms. We first estimate the probability of a category being a related category for the active product in the post-purchase stage. In this context, "category" refers to the taxonomy of the products, which is typical in e-commerce sites e.g., **digital camera, lenses, clothing, printers** and so on. The product-level recommendations are further extracted within these top related categories. Moreover, post-purchase recommendations are often time-sensitive, the recommendation for the same user and the same product could change in different time windows. We address this interesting **time sensitivity** challenge by adding a temporal factor into the recommendation relevance score.

In this paper, we present our methodology and experimental results in designing a post-purchase recommender system that addresses the above challenges. To summarize, our contributions include, **a)** algorithms for automatic discovery of related categories for a given product/category from the behavioral relevance in purchase data and design of an enhanced item-based similarity function for post-purchase recommender system and **b)** incorporation of temporal factors in the recommendation process to provide diversified recommendations for different time windows.

RecSys'11, October 23–27, 2011, Chicago, Illinois, USA.

We combine all these enhancements and design a hybrid recommender system for the post-purchase stage. We conduct rigorous experiments on a large scale user data, with purchase history of millions of eBay users. The experimental results suggest that our system significantly improves the recommendation quality and coverage (12% to 30% quality gain and 4% to 37% coverage gain).

2. RELATED WORK

Two major directions of recommender systems are the content-based approach and the collaborative filtering approach. The former utilizes item's content and the latter focuses on using user's behavior history. In content-based approach, the idea is to detect products that are most "similar" to the user's existing profile. Various approaches [7] have been developed, such as cosine similarity with TF-IDF term weight, Bayesian classifiers, clustering, etc. In our research, we observe that the content-based approach is very useful to increase the recommendation coverage. Another direction is to utilize users' behavior history to make recommendations. In the user-item matrix, each row is naturally a user vector and each column is an item vector. The user-based method utilizes the user vector, which are often termed **memory-based** approaches in literature [2, 9]. The item-based method utilizes the item vector, and belongs to **model-based** approaches. Instead of finding the similar user neighbors, this method directly finds the similar item neighbors for items in the user vector. The item-based method significantly improved the recommendation quality and reduced the computation cost, compared with the user-based method [4, 8].

Temporal effect is an essential factor in recommender systems, especially in real world e-commerce applications. In [3], the item-item similarity was weighted by the time distance between two items using an exponential decay function and found that it performed better than pure item-based collaborative filtering. Another important work by Koren [6] thoroughly analyzed the temporal effect in collaborative filtering approaches. Instead of using traditional time-window or instance-decay method, they tracked the time changing behavior through the whole data. It showed significant improvement over the baseline predictor. They reported the best performance on the widely used Netflix movie data [1].

3. DESIGN OF A POST-PURCHASE RECOMMENDER

We focus on designing a recommender system for the post-purchase stage i.e., after a user purchases a product, We term this the **active product**, for which all recommendations are computed. We treat the problem as a ranking problem, which estimates the recommendation score for every product and selects the top ones to recommend. The recommendation score is computed based on the following factors.

3.1 Category-level Relevance Factor $\mathcal{R}_C^B$

The category-level behavioral relevance value reflects the system's own knowledge of recommending a related category for the category of an active product. For instance, the score $\mathcal{R}_C^B$ should enable the system to determine that "cellphone accessory" is a related category for "cellphone" category, but not for "camera" category. Assume that product p_k belongs to category c_k and the active product p_a belongs to category c_a. Then the value $\mathcal{R}_{c_k}^B$ is defined as the probability of category c_k being a valid post-purchase category for the active category c_a. To compute this, we assume that the purchase behavior of each c_j follows a **multinomial** distribution. We denote the probability of category c_j being a related post purchase category for c_i as $\theta_{i,j}$, $i = 1, ..., Q$. The probability of category c_j being a self purchase (not related) category is denoted as $\theta_{0,j}$. Our desired $\mathcal{R}_{c_k}^B$ is denoted by $\theta_{a,k}$. In our training data there are M users and Q categories. The overall data likelihood for purchasing products from category c_j in the user-category matrix $C_{M\times Q}$ can be represented as $L_j = \prod_{i=0}^{Q}[(\theta_{i,j})^{\sum_{m=1}^{m=M} k_{i,j,m}}]$. The maximum likelihood estimation (MLE) for $\theta_{a,k}$ is denoted as $\hat{\theta_{a,k}}$ and we get $\mathcal{R}_{c_k}^B = \hat{\theta_{a,k}} = \frac{\sum_{m=1}^{m=M} k_{a,k,m}}{\sum_{i=0}^{i=Q}\sum_{m=1}^{m=M} k_{i,k,m}}$, where $k_{i,j,m}$ is the number of times category j was purchased after category i by user m. Note here when $i = 0$, $k_{0,j,m}$ is the number of times category j was a self-purchase, which was not related to previous purchases. We rank all categories by their $\hat{\theta_{a,k}}$, and denote the top ones as *top-related post-purchase* categories for the active product/category.

3.2 Product-level Relevance Factor $\mathcal{R}_P^B$

In the next step, we restrict our focus on the categories we discovered in section 3.1 and find relevant recommendations in the product-level. The system learns from the past purchase data and predicts products that are likely to be purchased after the active product given these categories. We term this relevance score as the **product-level behavioral relevance**. In order to increase the recommendation coverage, and alleviate "cold-start" problem, we leverage the content-level relevance to discover good recommendations even if they weren't purchased in history. Related products that share some key content information with the active product could potentially be good recommendations. For instance, "Nikon fabric case(9691) for P90" is a relevant related product for "Nikon Coolpix P90 Camera" irrespective of their co-purchase history. We term this relevance score as the **product-level content relevance**.

3.2.1 *Product-level Behavioral Relevance*

This value reflects the likelihood that users purchase product p_k after the active product p_a. We utilize the item-based similarity using conditional frequency that is widely used in collaborative filtering literature [4]. In addition, we also add some refinements as described below.

If product p_k is recommended, it indicates that p_k was likely to be purchased after p_a in history. Thus the purchase order between p_k and p_a is an essential factor to determine the recommendation probability. We utilize the indicator function $I_{t_{p_a}<t_{p_k}}$ to guarantee that all contribution comes from the product that was purchased after the active product. Also, to ensure stronger similarity, we only consider two products if more than $Thres_u$ users have co-purchased them in history. The indicator function $I_{co_{k,a}>Thres_u}$ implements this idea. It is a common practice in the item-based similarity function [5]. The behavioral relevance is now estimated by equation 1.

$$\mathcal{S}_{(p_k,p_a)}^p = \frac{\sum_{\forall i:r_{i,a}>0} r_{i,k} I_{t_{p_a}<t_{p_k}} I_{co_{k,a}>Thres_u}}{Freq(p_k)^\alpha \times Freq(p_a)} \quad (1)$$

We define $r_{i,k}$ as $\frac{1}{||purchase_i||}$ if user u_i purchased product p_k, where $||purchase_i||$ is the total number of product purchases made by user u_i. With higher threshold $Thres_u$, less recommendation could be learnt from the history data while the recommendations are expected to be more reliable. In order to increase the coverage of recommendations, we also explore the behavioral transitivity to generate more recommendations for a particular product. We utilize users' view/browse data to find similar products for the active product or similar products for the related product. Finally, we combine all factors to compute the product-level behavioral relevance score, $\mathcal{R}_P^B$.

3.2.2 *Product-level Content Relevance*

The content-based relevance element can help to increase the coverage. $\mathcal{R}_P^N$ estimates the recommendation score based on two product's content. In this case, even if users never purchased product p_k after the active product p_a, p_k can still be recommended if its content is related to p_a's content. We use title terms as product content here, which can be easily extended to other content information. Considering the compatibility issue, we utilize the exact match algorithm to compare between p_k and p_a. The key terms of a product typically include its manufacturer or brand, product line, and model name. If product p_k contains all key terms of the active product, we set the $\mathcal{R}_{p_k}^N$ as 1. Otherwise, it is 0.

3.3 Temporal Factor

In order to improve the recommendation relevance and eventually the conversion rate, **recommending the right products at the right time** becomes important. For example, if a user just purchased a camera, the system can recommend compatible lenses directly after this action. Then after a certain time, the system can further recommend some compatible filters for those lenses. If the system recommends filters directly after user purchased a camera without a lens, the user would most likely ignore the recommendation even though camera and lens are related. This suggests that the timing is equally important for recommendation as relevance. We consider the purchase time t_p and the recommendation time t_r; for an active product it is same as considering the time window between these two times. In order to introduce the temporal diversity, we make the behavioral relevance in category level as well as in product level dependent on this time factor. The time window t_w is the interval between purchase time and recommendation time. In this work, we used three time windows, which are **within 1 day, from 1 day to 1 week, from 1 week to 1 month**. It can be easily extended to other time windows. Next we show how we incorporate this temporal factor in our relevance computation.

3.3.1 *Category Level Behavioral Relevance with* t_w

In the MLE computation of section 3.1, if t_w is not considered, the frequency $k_{i,j,m}$ is the number of times that category j was purchased after category i by user m. If t_w is considered, the frequency $k_{i,j,m,t}$ becomes the number of times that category j was purchased after category i by user m, within the given time window t_w.

3.3.2 *Product Level Behavioral Relevance with* t_w

In product level, the contribution of a previous co-purchase behavior is dependent on the time distance. If the time distance between two purchases was within the pre-defined time window t_w, the contribution should be counted. Otherwise, the contribution should be decayed. Thus we extend equation 1 by multiplying every $r_{i,k}$ with the time window decay factor, $Decay(t_d, t_w)$.

$$\mathcal{S}^p_{(p_k,p_a,t_k,t_r)} = \frac{\sum_{\forall i: r_{i,a}>0} r_{i,k} I_{t_{p_a}<t_{p_k}} Decay(t_d,t_w)}{Freq(p_k)^\alpha \times Freq(p_a)}$$
$$t_d = t_r - t_p \qquad (2)$$

Assume that t_d is the time distance between a co-purchase of $< p_a, p_k >$ in history. The time decay factor is defined in equation 3. In this paper we only consider a step function decay factor, the use of other potential *soft* decay functions as used in [3] remains as a future work.

$$Decay(t_d, t_w) = \begin{cases} 1 & t_d \le t_w \\ 0 & t_d > t_w \end{cases} \qquad (3)$$

3.4 Estimating the Recommendation Score

Based on the analysis, we estimate the recommendation score by equation 4. The $\mathcal{R}_C^B$ factor controls that the recommendations come from the top related categories. The $\mathcal{R}_P^B$ and $\mathcal{R}_P^N$ factors estimate the recommendation quality of a product in behavior dimension and content dimension respectively. $\mathcal{R}_{C,t_r,t_p}^B$ and $\mathcal{R}_{P,t_r,t_p}^B$ indicate that these two values are dependent on the temporal factor.

$$\mathcal{R} = \mathcal{R}_{C,t_r,t_p}^B \times (\mathcal{R}_{P,t_r,t_p}^B + \mathcal{R}_P^N) \qquad (4)$$

In this section, we have shown how to estimate the factors to calculate the recommendation value in 4. In the following sections, we discuss our experimental setup and results.

4. EXPERIMENTAL SETUP AND METRICS

4.1 Dataset and Evaluation Metric

In the training data, we collected a sample of 3 million unique eBay users from Jan 1, 2010 to June 30, 2010. These users purchased around 70,000 unique products in 200 product categories. For the test data, we collected a sample of 1 million unique users who purchased in the month of July, 2010. These users purchased around 50,000 unique products in 200 product categories. We report our experimental results for the digital camera category, which has around 4800 unique products in this sample data. We also perform experiments in cellphone and video game categories, which show similar results. We use the following two metrics to evaluate our algorithms.

Recommendation Coverage computes the percentage of products with recommendations. The higher the coverage, the more benefit the system can provide to the user. It is calculated as $\mathcal{C} = \frac{\#\text{ of active products with recommendations}}{\#\text{ of all active products}}$

Purchase Rate is the percentage of times that users purchase from our recommendations. This metric is defined as $\mathcal{P} = \frac{\|\bigcap(p_{purchase}, p_{recommendation})\|}{\|p_{purchase}\|}$, where the numerator is the number of times that users purchase from our recommendation lists and the denominator is the number of times that users have post-purchase behavior.

4.2 Baseline Recommender System

To compare our algorithmic results, we consider a basic item-based collaborative filtering recommender system,

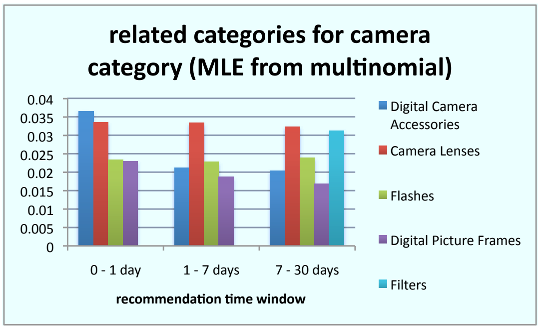

Figure 1: Estimation of top related categories for digital cameras, in the post-purchase stage at three different time windows.

which utilizes the conditional purchase frequency as the similarity function. It calculates the relevance score according to equation 1. It is denoted as the "baseline algorithm". This baseline system uses $\alpha = 0.5$ and $Thres_u = 0$. Every time window has its own baseline performance.

5. EXPERIMENTAL RESULTS ANALYSIS

5.1 Category-level Recommendation Results

In Figure 1 we show the top related categories for digital cameras using the multinomial MLE method described in section 3.1. For instance, in the $0-1$ day time window, MLE method chooses camera accessories, lenses, flashes, picture frames as top related categories. For $7-30$ day, this order changes to lenses, lens filters, flashes, camera accessories and picture frames.

From these observations it can be concluded that by using multinomial distribution we can obtain reasonable *related* categories. It is also evident that the use of the time window t_w has a significant impact over the the distribution of the related categories. For different values of t_w, the distribution of top related categories changes considerably. By incorporating this factor, the recommender system can provide more accurate and targeted recommendations in each time window.

5.2 Product-level Recommendation Results

After determining the top related categories, we look into the performance of our algorithm in the product level. The products in the top related categories are more likely to be recommended to the active product according to equation 4. We put together all our best performing components together and compare our overall post-purchase recommender system with the baseline system described in section 4.2. The results are shown in Table 1. It can be seen from the table that our hybrid recommender system with all factors mentioned in equation 4 achieves 12% to 30% purchase rate improvement while achieving 4% to 37% coverage improvement for different time windows. This happens as the system brings more relevant related product recommendations for the post-purchase stage at the right time. This indicates a significant improvement over our baseline in terms of both metrics.

Method	purchase rate			coverage		
	0-1 day	1-7 day	7-30 day	0-1 day	1-7 day	7-30 day
item-based baseline	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
hybrid system	112.1%	122.5%	130.8%	136.6%	114.8%	104.3%

Table 1: Purchase rate and converage performance of all methods in post-purchase stage. All values are in relative percentages, using the baseline performance for each time window as 100%.

6. CONCLUSION AND FUTURE WORK

In this paper, we present a novel recommender system for the post-purchase stage. This system leverages the historical purchase data to discover related products, which are more likely to be purchased by the user in future. We present a framework to combine users' behavioral relevance as well as products' content relevance to increase the recommendation coverage and quality. In addition, we introduce the temporal factor, which helps us recommend the "right product at the right time". Our final hybrid recommender system achieves 4% to 37% coverage gain while achieving 12% to 30% purchase rate gain for different time windows of recommendation.

In future we plan to extend our system by incorporating the user information to provide personalized recommendations and conduct some online experiments to better understand the implications of our methods. In our current experiments we use a shorter time period of 30 days due to limitations in our test dataset. In future, we plan to look at longer time windows such as 6 months, 1 year, or more. Going from on-the-spot recommendation to a longer time campaign will enable e-commerce sites to achieve increased customer loyalty, which in turn would result in increased revenue.

7. REFERENCES

[1] J. Bennett, S. Lanning, and N. Netflix. The netflix prize. In *In KDD Cup and Workshop, KDD '07*, 2007.

[2] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. pages 43–52. Morgan Kaufmann, 1998.

[3] Y. Ding and X. Li. Time weight collaborative filtering. In *Proc. of CIKM '05*, pages 485–492, New York, NY, USA, 2005. ACM.

[4] G. Karypis. Evaluation of item-based top-n recommendation algorithms. In *Proc. of CIKM '01 conference*, pages 247–254, New York, NY, USA, 2001.

[5] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proc. of ACM KDD '08 conference*, pages 426–434, New York, NY, USA, 2008. ACM.

[6] Y. Koren. Collaborative filtering with temporal dynamics. In *Proc. of KDD '09 conference*, pages 447–456, New York, NY, USA, 2009. ACM.

[7] M. Pazzani, D. Billsus, S. Michalski, and J. Wnek. Learning and revising user profiles: The identification of interesting web sites. In *Machine Learning*, pages 313–331, 1997.

[8] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proc. of WWW '01 conference*, pages 285–295, New York, NY, USA, 2001. ACM.

[9] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Analysis of recommendation algorithms for e-commerce. In *Proc. of EC '00 conference*, pages 158–167, New York, NY, USA, 2000. ACM.