Recommending Complementary Products in E-Commerce Push Notifications with a Mixture Model Approach

Huasha Zhao, Luo Si, Xiaogang Li, Qiong Zhang
Alibaba Group
400 S El Camino Real, #400
San Mateo, California, USA 94402
{huasha.zhao,luo.si,xiaogang,qz.zhang}@alibaba-inc.com

ABSTRACT

Push notification is a key component for E-commerce mobile applications, which has been extensively used for user growth and engagement. The effectiveness of the push notification is generally measured by message open rate. A push message can contain a recommended product, a shopping news and etc., but often only one or two items can be shown in the push message due to the limit of display space. This paper proposes a mixture model approach for predicting push message open rate for a post-purchase complementary product recommendation task. The mixture model is trained to learn latent prediction contexts, which are determined by user and item profiles, and then make open rate predictions accordingly. The item with the highest predicted open rate is then chosen to be included in the push notification message for each user. The parameters of the mixture model are optimized using an EM algorithm. A set of experiments are conducted to evaluate the proposed method live with a popular E-Commerce mobile app. The results show that the proposed method is superior than several existing solutions by a significant margin.

1 INTRODUCTION

Push notification service [1, 8, 11, 13] is a key component of E-commerce mobile applications. It pushes text messages to users to provide a seamless shopping experience. The push messages are usually classified into two categories: transaction related push message which reminds the user of transaction information on existing orders, such as payment processing, logistic updates and etc., and marketing push message which promotes new product or shopping news that the user may be interested in. Some examples of push messages are illustrated in Figure 1. Push message has been extensively used for user growth and user engagement. It initializes the connection with user in an active manner, and can bring users back to the E-commerce mobile app. The effectiveness of the push message is measured by *open rate* which is the percentage of messages clicked/viewed by users.

In marketing push notification, recommending the "right" product is critical. There has been substantial research in recommendation systems [3, 6, 12]. However, push notification has several

Figure 1: Three examples of push messages [1]: The top two are marketing push messages. The top message is a PPR which recommends a personalized sofa mat without making specific connections with user behavior. The middle message is a CPR which recommends Tie Guan Yin tea after the user purchased a teapot. The bottom one is transaction related push message. The message reminds the user that the purchased teapot has been shipped from seller.

key differences in comparison with classic recommendation tasks. First, the text template or the presentation of the push message can be more important in determining the open rate than the recommended product itself. Attractive message slogans sometimes play a decisive role. To increase open rate, the message needs to be created with strong and preferably direct attachment with the user behavior. Furthermore, only one or two items can be shown in the push message due to the limit of display space. Finally, only a given number of messages can be sent to a user during a period of time, to avoid spamming. As a result, the prediction accuracy of open rate becomes critically important due to limited chance of recommendation.

In this paper, we propose a method for recommending post-purchase complementary products through push notification. Post-purchase recommendation has significant advantage in the push setting in comparison with pushing a single personalized product without connecting it to user history. Examples of complementary product recommendation (CPR) and personalized product recommendation (PPR) are illustrated in Figure 1. PPR pushes only one product to the user using some collaborative filtering algorithms, and the message contains only one anchor to the user. In contrast,

---

[1]Translations of the messages in the figure, from top to bottom. 1, Go shopping during the weekend? All-season all-purpose non-slip cloth sofa mat for you. 2, Shopping together: 2016 Anxi fragrant Tie Guan Yin tea with your newly bought teapot. 3, Logistic reminder: your purchased item Xianmingju health teapot has been shipped.

CPR naturally embeds two anchors in the push message template - the purchased product and the recommended product, and it offers stronger connections with the user than PPR. As a result, such messages are more attractive to users.

In addition, a mixture model is presented to predict message open rate. The predicted open rate is in turn used to determine the product to recommend. Mixture model is also known as probabilistic latent class model, and it has been used in user modeling and recommendation systems [4, 5, 7, 14]. A mixture model can automatically learn the underlying structure of a prediction task, and improve the prediction accuracy by training separate sets of forecasting weights for different latent prediction classes.

In our problem, the mixture model is used to represent hidden prediction contexts. A context is determined by a combination of user and product profiles. For example, some users may be more interested in the popularity of the product, while others care more about the complementarity between the recommended product and purchased product. In this case, in the prediction model we need to assign higher weights on product popularity feature to the first group of users while give more weights on complementary score to the second group. A mixture model can be trained to learn such latent contexts for prediction.

The rest of the paper is arranged as follows: Section 2 describes complementary score calculation for CPR, and the score is one of the key features used in the model. The mixture model for open rate prediction is discussed in Section 3. Section 4 presents experiment results along with analysis. And finally Section 5 concludes the paper.

2 COMPLEMENTARY PRODUCT RECOMMENDATION

Complements and substitutes are two types of recommendation products [9, 10]. Complements can be bought in addition to each other. However, substitutes can be bought instead of each other. In the post-purchase recommendation scenario, it is critical to recommend complementary products, but not substitutes. This reason is that users have already made the decision, and may feel being spammed if a product that serves the same purpose is recommended again. Here we present the method we use for complementary product candidate selection. We choose the product pairs with high co-purchase scores and low substitutivity scores at the same time. The scoring method is described in the following.

2.1 Co-Purchasing Graph

Assume A_{ui} is the user-product purchase graph for user $u \in \mathbb{U}$ and product $i \in \mathbb{P}$ over a certain period of time, and $t(A_{ui})$ is the timestamp at which the purchase event happens. The co-purchase score p_{ij} - the score for buying j after buying i is defined as follows,

$$p_{ij} = \frac{\sum_{u \in \mathbb{U}} A_{ui} A_{uj} \mathbf{1}(t(A_{uj}) > t(A_{ui}))}{\sqrt{\sum_{u \in \mathbb{U}} A_{ui}^2 \sum_{u \in \mathbb{U}} A_{uj}^2}}. \tag{1}$$

2.2 View-and-then-Purchase Graph

Further assume B_{ui} is the user-product view graph, and $t(B_{ui})$ is the timestamp at which the view event happens. The substitutivity

score q_{ij} between product i and j is defined as follows,

$$q_{ij} = \frac{\sum_{u \in \mathbb{U}} B_{ui} A_{uj} \mathbf{1}(t(A_{uj}) > t(B_{ui}))}{\sqrt{\sum_{u \in \mathbb{U}} B_{ui}^2 \sum_{u \in \mathbb{U}} A_{uj}^2}}. \tag{2}$$

The final complementary score for product pair i, j is determined by,

$$s_{ij} = p_{ij} - q_{ij}. \tag{3}$$

The score selects product pairs with high complementarity and low substitutivity. We also calculate complementary scores for categories with the same method.

3 MIXTURE MODEL FOR OPEN RATE PREDICTION

In this section, the mixture model for predicting the message open rate for user-product pairs is presented. The motivation comes from the nature of users and products. For instance, some users prefer popular products (product popularity score) regardless of the complementariness between the two products (complementary score). At the same time, different classes of products may also cause different sets of prediction parameters.

At high level, the model is comprised of two parts: an assignment model which maps the inputs to prediction *contexts* and a context-aware prediction model for open rate forecasting. Formally, we assume the following probabilistic model:

$$P(\mathbf{y}|X, \hat{X}, \Theta, \Psi) = \prod_{i=1}^{N} P(y_i|\mathbf{x}_i, \hat{\mathbf{x}}_i, \Theta, \Psi)$$

$$= \prod_{i=1}^{N} \sum_{z_i=1}^{M} P(z_i|\hat{\mathbf{x}}_i, \Theta) P(y_i|\mathbf{x}_i, z_i, \Psi). \tag{4}$$

Here N is the total number of examples in the dataset. There are two sets of features $\hat{\mathbf{x}}_i \in \mathbb{R}^m$, $\mathbf{x}_i \in \mathbb{R}^n$ for each example $i \in \{1, \ldots, N\}$. $\hat{\mathbf{x}}_i$ is the context assignment feature and $\mathbf{x}_i$ is the open rate prediction feature. In practice, features in these two sets may have overlaps. We describe these features in more details in Section 3.1. Furthermore, each example is labeled $y_i \in \{0, 1\}$ to represent whether it is opened or not, and a hidden variable z_i which assigns each example to a predicting context. We further assume M contexts of interest and $z_i \in \{1, \ldots, M\}$. The model is characterized by two sets of parameters, i.e. $\Theta = (\theta_1, \ldots, \theta_M)$ and $\Psi = (\psi_1, \ldots, \psi_M)$. Both θ_k and ψ_k, $k \in \{1, \ldots, M\}$ are of the same dimension as $\hat{\mathbf{x}}_i$ and $\mathbf{x}_i$ respectively, for $i \in \{1, \ldots, N\}$, and they parameterize the assignment model and prediction model of the kth context respectively.

In more details, the assignment process can be modeled with a multi-class logistic model as the following,

$$P(z = k|\hat{\mathbf{x}} = \hat{x}) = \frac{\exp(\theta_k^t \hat{x})}{1 + \sum_{j=1}^{M-1} \exp(\theta_j^t \hat{x})}, \quad k = 1, \ldots, M-1,$$

$$P(z = M|\hat{\mathbf{x}} = \hat{x}) = \frac{1}{1 + \sum_{j=1}^{M-1} \exp(\theta_j^t \hat{x})}. \tag{5}$$

In this case, Θ is only defined up to a multiplicative constant, so that θ_M can be omitted in the model. A binary logistic regression model

is used for prediction, for each scenario $k \in \{1, \ldots, M\}$ respectively,

$$P(y = 0|\mathbf{x} = x, z = k) = \frac{\exp(\psi_k^t x)}{1 + \exp(\psi_k^t x)},$$

$$P(y = 1|\mathbf{x} = x, z = k) = \frac{1}{1 + \exp(\psi_k^t x)}. \qquad (6)$$

The log-likelihood expression for model from the data $(X, \mathbf{y})$ is difficult to optimize because it involves the log of the sum. If the hidden variable z is introduced, however, the likelihood function can be significantly simplified:

$$\log(\mathcal{L}(\Theta, \Psi | X, \hat{X}, \mathbf{y}, \mathbf{z})) = \log(P(X, \hat{X}, \mathbf{y}, \mathbf{z} | \Theta, \Psi)) \qquad (7)$$

$$= \sum_{i=1}^{N} \log(P(\mathbf{x}_i) P(\hat{\mathbf{x}}_i) P(z_i | \hat{\mathbf{x}}_i, \Theta) P(y_i | \mathbf{x}_i, z_i, \Psi)).$$

The above likelihood is intractable for solving optimized parameters analytically. We resort to the EM algorithm to find optimal Θ and Ψ. In the E step, we first derive an expression of the posterior distribution of the unobserved data. Using Bayes's rule and property of conditional independence, we can compute,

$$P(z_i | \hat{\mathbf{x}}_i, \mathbf{x}_i, y_i, \Theta, \Psi) = \frac{P(z_i | \hat{\mathbf{x}}_i, \Theta) P(y_i | \mathbf{x}_i, z_i, \Psi)}{\sum_{k=1}^{M} P(k | \hat{\mathbf{x}}_i, \Theta) P(y_i | \mathbf{x}_i, k, \Psi)}. \qquad (8)$$

An auxiliary Q function [2] of the likelihood can be derived as,

$$Q(\Theta', \Psi' | \Theta, \Psi)$$

$$= E[\log(\mathcal{L}(\Theta', \Psi' | X, \hat{X}, \mathbf{y}, \mathbf{z})) | X, \hat{X}, \mathbf{y}, \Theta, \Psi]$$

$$= \sum_{k=1}^{M} \sum_{i=1}^{N} \log(P(k | \hat{\mathbf{x}}_i, \Theta')) P(k | \hat{\mathbf{x}}_i, \mathbf{x}_i, y_i, \Theta, \Psi)$$

$$+ \sum_{k=1}^{M} \sum_{i=1}^{N} \log(P(y_i | \mathbf{x}_i, k, \Psi')) P(k | \hat{\mathbf{x}}_i, \mathbf{x}_i, y_i, \Theta, \Psi). \qquad (9)$$

Note, $P(\mathbf{x}_i)$ and $P(\hat{\mathbf{x}}_i)$ are removed in the Q function since we assume they both follow uniform distribution. To maximize the the Q function, we can maximize the term containing Θ' and the term containing Ψ' independently since they are not related. We can also solve θ_k' for each scenario independently for the same argument. The M step updates can be derived as the following,

$$\Theta' \propto \arg\max_{\Theta'} \sum_{k=1}^{M} \sum_{i=1}^{N} \log(P(k | \hat{\mathbf{x}}_i, \Theta')) P(k | \hat{\mathbf{x}}_i, \mathbf{x}_i, y_i, \Theta, \Psi),$$

$$(10)$$

$$\psi_k' \propto \arg\max_{\psi_k'} \sum_{i=1}^{N} \log(P(y_i | \mathbf{x}_i, k, \psi_k')) P(k | \hat{\mathbf{x}}_i, \mathbf{x}_i, y_i, \Theta, \psi_k).$$

$$(11)$$

The above updates can be solved using a gradient decent solver.

3.1 Model Features

As shown in Table 1, there are 4 types of features we use in the mixture model. They are user features, product features, user-product features and product-product features.

Besides user demographics, we also use user cluster features and user active scores. User clusters are generated by running k-means on user shopping behaviors (at category level) within certain time

user features	user cluster id
	user active score
	user demographics, e.g. age, income and etc.
product features	product sales in the past 1, 2, 7, 28 days
	product views in the past 1, 2, 7, 28 days
	price and other metadata
user-product features	user-product preference scores in the past 1, 2, 7, 28 days respectively
	user-category preference scores in the past 1, 2, 7, 28 days respectively
product-product features	product complementarity scores defined in equation 3
	category complementarity scores

Table 1: Model features.

period. User active score measures how active the user is on the E-Commerce platform. Furthermore, we use product features and user-product features generated from multiply time intervals, to capture the time dynamic of user and product behaviors. Finally, we also include product-product features: the calculations of which are presented in Section 2.

User features and product features are included in $\hat{\mathbf{x}}_i$, to predict prediction context. While all features except user features are used in the second stage (open rate prediction) of the mixture model (in $\mathbf{x}_i$) - user features are not needed when ranking products for a specific user.

4 EXPERIMENTS

This section presents the experiments for evaluating the mixture model. The model is tested live on a real-world push notification task on a popular E-commerce mobile application with hundreds of millions of active users.

4.1 Experiment Setup

In model training, one month push notification log data is used. The complementary product push task reaches 10 million users daily. There are in total 300 million records for training. L-BFGS solver is applied to compute equations (10) (11) required by the M-step. The threshold of likelihood convergence is set to $1e^{-5}$.

4.2 Number of Contexts Evaluation

The optimal number of hidden contexts k is evaluated in this section. Figure 2 plots the log likelihood at convergence for different k. As we can see, there is no benefit of increasing k beyond $k = 4$. In addition, we also evaluate the model performance for different sets of features ($\hat{\mathbf{x}}_i$) used for context prediction. We compare full features (described in Section 3.1) with user features only and product features only. The full model learns the user and product mixture jointly. As a result, it outperforms both user only and product only models. From Figure 2, we also observe that the model of product features only is much worse. This means there are more opportunities to explore user mixtures than product mixtures. In other words, it will be more beneficial to model open rate at user dimension rather than at product dimension, if we have to choose one.
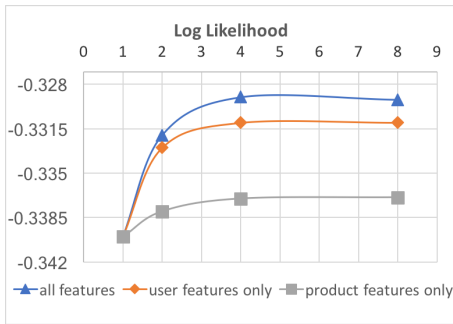
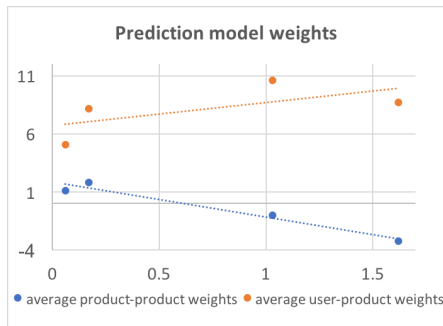**Figure 2: Log-likelihood per example with different feature sets and different number of contexts k.**

Figure 3: Prediction model weights for different user clusters based on user active scores.

4.3 Impact of User Cluster on Model Weights

Experiments are run to illustrate the impact on prediction model weights of different sub-populations in the mixture model. Figure 3 plots the average of product-product feature weights and user-product feature weights in the prediction model against the weight of user active score in the context assignment model. Assignment model with higher user active score weight selects more active users. As shown in the figure, more active users prefer products with higher user preference scores, while less active users prefer more on complementariness between products. Intuitively this can be explained as: sophisticated users know what they want to buy and have strong product preferences, while newbie users rely more on the complementary product recommendation from the platform.

4.4 Online Experiment Results

The CPR with mixture model is deployed live to our production push recommendation system, and A/B tests are conducted. Results are shown in Table 2. The baseline just pushes the most popular products to users without any personalization. In PPR, a state-of-the-art item-based recommender is applied to compute personalized product. The baseline CPR model selects the product for a user with highest user-product score times product complementarity score without considering other features. It is also worth noting that a mixture model (MM) with $k = 1$ is equivalent to a logistic regression model for open rate prediction.

CPR outperforms PPR by a significant margin. This confirms our hypothesis discussed in the introduction. The CPR textual

	Open rate rel. to baseline	P-value
(1) PPR	2.01	.00011*** (vs. baseline)
(2) CPR	2.89	.00036*** (vs. 1)
(3) CPR with MM k=1	3.12	.00073*** (vs. 2)
(4) CPR with MM k= 4	3.22	.0046** (vs. 3)

Table 2: Open rates of different methods and statistical significance tests.

presentations are richer and contain two anchors to the users. And therefore, CPR delivers better results. The mixture model offers additional gains in message open rate. The open rate improvement is around 11%. The figure also shows it is beneficial to model latent user product clusters for open rate prediction. In comparison with a plain logistic regression (mixture model with $k = 1$), the gain is about 3%. Although 3% does not look like a very big number, the impact of the mixture model is significant given the huge user base.

5 CONCLUSION AND FUTURE WORKS

This paper presents a mixture model for post-purchase complementary product recommendation. The model learns different open rate prediction weights for different user-product contexts. Experiment results show that CPR is superior than PPR by a significant margin, and the proposed mixture model gives additional gain in open rates. There are several possibilities to extend the research. Temporal features of user behavior are not included in the model. Furthermore, it will also be beneficial to consider product complementary score at different time scales, so that complementary product can be pushed to users at the most appropriate time.

REFERENCES

[1] Kris M Bell, Darryl N Bleau, and Jeffrey T Davey. 2011. Push notification service. (Nov. 22 2011). US Patent 8,064,896.
[2] Jeff A Bilmes and others. 1998. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. *International Computer Science Institute* 4, 510 (1998), 126.
[3] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. 2013. Recommender systems survey. *Knowledge-based systems* 46 (2013), 109–132.
[4] Suleyman Cetintas, Datong Chen, and Luo Si. 2013. Forecasting user visits for online display advertising. *Information retrieval* 16, 3 (2013), 369–390.
[5] Suleyman Cetintas, Luo Si, Yan Ping Xin, and Ron Tzur. 2013. Probabilistic latent class models for predicting student performance. *CIKM, 2013* (2013), 1513–1516.
[6] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. 1999. An algorithmic framework for performing collaborative filtering. In *SIGIR, 1999*. ACM, 230–237.
[7] Rong Jin, Luo Si, and Chengxiang Zhai. 2006. A study of mixture models for collaborative filtering. *Information Retrieval* 9, 3 (2006), 357–382.
[8] Arvind Kumar and Suchi Johari. 2015. Push notification as a business enhancement technique for e-commerce. In *ICIIP, 2015*. IEEE, 450–454.
[9] Wei Lu, Wei Chen, and Laks VS Lakshmanan. 2015. From competition to complementarity: comparative influence diffusion and maximization. *Proceedings of the VLDB Endowment* 9, 2 (2015), 60–71.
[10] Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring networks of substitutable and complementary products. In *SIGKDD, 2015*. ACM, 785–794.
[11] Veronica S Moertini and Criswanto D Nugroho. 2012. E-commerce mobile marketing model resolving users acceptance criteria. *International Journal of Managing Information Technology* 4, 4 (2012), 23.
[12] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009 (2009), 4.
[13] Luchen Tan, Adam Roegiest, Jimmy Lin, and Charles LA Clarke. 2016. An exploration of evaluation metrics for mobile push notifications. In *SIGIR, 2016*. ACM, 741–744.
[14] Tao Tao and ChengXiang Zhai. 2004. A two-stage mixture model for pseudo feedback. In *SIGIR, 2004*. ACM, 486–487.