Knowledge-Based Systems 113 (2016) 51–60

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

SalesExplorer: Exploring sales opportunities from white-space customers in the enterprise market

Dongsheng Li[a,d,e,*], Yaoping Ruan[b], Qin Lv[c], Li Shang[c,e]

[a] IBM Research - China, Shanghai, 201203, China
[b] IBM T. J. Watson Research Center, Hawthorne, New York, 10532, USA
[c] University of Colorado Boulder, Boulder, Colorado, 80302, USA
[d] Shanghai Key Laboratory of Data Science, Shanghai, 201203, China
[e] Tongji University, Shanghai, 201802, China

ARTICLE INFO

Article history:
Received 18 July 2015
Revised 9 September 2016
Accepted 14 September 2016
Available online 17 September 2016

Keywords:
Recommender system
Collaborative filtering
Logistic regression
White-space customer

ABSTRACT

In commercial sales and services, recommender systems have been widely adopted to predict customers' purchase interests using their prior purchasing behaviors. Cold-start is a known challenge to existing recommendation techniques, e.g., the popular collaborative filtering method is not applicable to predict the interests of "white-space" customers since they have no prior purchasing history in the targeted product categories.

This paper presents SalesExplorer, a new recommendation algorithm to address "white-space" customer issue in the commercial sales and services segment. To predict the interests of customers who are new to a product category, we propose a statistical inference method using customers' existing purchase records from other product categories, a Probabilistic Latent Semantic Analysis (PLSA)-based transfer learning method using customers' business profile content, and a kernel logistic regression-based model which combines these two recommendations to produce the final results with higher accuracy. Experimental study using real-world enterprise sales data demonstrates that, comparing with a baseline and two state-of-the-art methods, the proposed combinatorial algorithm improves recommendation accuracy by 32.14%, 13.13% and 9.85%, respectively.

1. Introduction

In the enterprise market, recommender systems have been widely adopted for sales and services, e.g., cross-selling, upselling, or diversity [5,12,16,20,24]. Using prior purchase records, recommender systems predict the likelihood of the customers' interests in other products in order to optimize customer discovery and conversion. In real-world enterprise market, "white-space" customers, which have no purchase history in the target product domain, will appear in a variety of cases, e.g., recommend products to new customers, recommend products to customers from other product lines, etc. Supporting "white-space" customers is one of the primary challenges faced by existing recommender systems. Collaborative filtering (CF), one of the most widely-adopted recommendation methods [1,7,21], relies on the fact that customers with similar historical purchase records will show similar interests for future products in the same product category. However, for "white-space" customers with no purchase record in the target product category, CF is unable to identify the corresponding customer group with similar purchase interests, which is also known as the "cold-start" problem. For instance, considering a customer with only "hardware" purchase history, it is a challenge for existing CF methods to make an informative decision whether to recommend certain "software" to the target customer or not.

Recently proposed recommendation algorithms, such as cold-start recommendations, aim to address the "white-space customer" issue. Several techniques utilize auxiliary information, such as customer profiles, when no prior purchase record is available for the customers [15,18,25,29]. In essence, they are content-based recommendation methods. Existing works [18,33,34] have shown that content-based recommendation methods are not as accurate as collaborative filtering methods in many scenarios. Therefore, the recommender system will suffer from accuracy issue if they only rely on content-based recommendations for "white-space" customers. Another type of method, namely cross-domain collabo-

* Corresponding author.
E-mail addresses: ldsli@cn.ibm.com, dongshengli@fudan.edu.cn (D. Li), yaoping.ruan@us.ibm.com (Y. Ruan), qin.lv@colorado.edu (Q. Lv), li.shang@colorado.edu (L. Shang).

http://dx.doi.org/10.1016/j.knosys.2016.09.011
0950-7051/

rative filtering (CDCF), adopts customer behaviors from other related product categories to help recommendation in the target category [14,19,23]. However, CDCF typically requires that at least a few records must be available for the target customers in the target product category. Otherwise, CDCF will suffer from the "blind-transfer" issue, i.e., recommendations rely solely on customer behaviors in other product category and as such lead to low recommendation quality [10]. In real-world commercial companies, it is often the case that customers have no purchase records in the target product category, e.g., cross-sell. Therefore, CDCF methods cannot provide high-quality recommendations for completely "white-space" customers. From the above analysis, we can know that collaborative filtering methods that rely on either only content profiles or only customers' purchasing behaviors from other categories cannot provide accurate recommendations for completely "white-space" customers.

This paper presents SalesExplorer, a new recommendation algorithm to address both the accuracy issue in content-based recommendations methods and the "blind-transfer" issue in CDCF methods as described above. It leverages both the customers' content profiles and prior purchase records from other product categories to generate accurate recommendations for completely "white-space" customers. Predictions of customers' behaviors in the target product category are generated by two integrated methods. The first is a statistical inference method, which can address the "blind transfer" issue by analyzing customers' behaviors in the related category and the transitive relationship between the related source category and the target category. Second, predictions are also generated based on customers' content profiles using a PLSA-based transfer learning method, which can accurately characterize the relationship between customer purchasing behaviors and their content profiles through the shared latent topics while alleviating the "noisy" information provided by irrelevant information in the content profiles. Finally, the two types of predictions are combined by a kernel logistic regression model, which further improves the recommendation accuracy. Meanwhile, a suit of optimization strategies are proposed to make the combination method be efficient against large-scale training data in online recommendation. Experimental study on real-world enterprise sales data demonstrates that the proposed method can improve the recommendation accuracy over a baseline method and two state-of-the-art methods substantially. The key contributions of this paper are summarized as follows:

- A statistical inference method, which can recommend products in the target category by analyzing customers' behaviors in the related category and the transitive relationship between the source category and the target category.
- A PLSA-based transfer learning method, which can accurately find shared latent topics between customer purchasing behaviors and their content profiles, and meanwhile alleviate the "noisy" information provided by irrelevant information in customers' content profiles.
- A kernel logistic regression-based combination method as well as a suit of efficiency enhancement strategies, which can combine the recommendation scores from the above two methods to further improve recommendation accuracy and ensure high recommendation efficiency.
- The proposed method is evaluated using real-world enterprise sales data, and the results demonstrate that the proposed method can improve the recommendation accuracy over a baseline method and two state-of-the-art methods by 32.14%, 13.13% and 9.85%, respectively, and can largely improve computation efficiency with negligible loss in recommendation accuracy.

The rest of this paper is organized as follows. Section 2 formulates the "white-space" customer recommendation problem. Section 3 discusses the related work. Section 4 presents the details of the proposed recommendation algorithm. Section 5 presents experimental results. Finally, we conclude this work and discuss future work in Section 6.

2. Problem formulation

This section formulates the "white-space recommendation" problem. Let C be a set of customers, and P, I^s and I^t denote the set of content profiles, the set of items in related (source) product category s and the set of items in the target product category t, respectively. Then, the profile of each customer $c \in C$ is denoted as a 3-tuple $\{p_c, I_c^s, I_c^t\}$, where $p_c \in P$ is the content profile of c and $I_c^s \subseteq I^s$ and $I_c^t \subseteq I^t$ are the purchasing histories of c in related (source) category s and target category t, respectively.

Based on the above notations, the "white-space recommendation" problem is formally defined in Definition 1.

Definition 1. Let $C_{tar} \subset C$ be a set of (target) white-space customers. For each customer $c \in C_{tar}$, c's purchasing history in the target product category is completely unknown, i.e., $I_c^t = \emptyset$. The goal of "white-space recommendation" is to predict the purchasing behaviors of customers in C_{tar} based on the profiles of customers in C.

The scenario of "white-space recommendation" can be regarded as the strict cold-start user scenario in collaborative filtering methods. In this scenario, customer behaviors in the target product category are completely unavailable. Therefore, traditional CF methods, which are based on the idea that customers who have similar interests in the past will have similar interests in the future, are not applicable, because historical interests of customers in the target product category are completely unknown. For instance, if we want to recommend "software" products to customers, but recommendations cannot be generated by CF methods if we have no ratings of these customers on any "software" product. Although we may have the target customers' interests on "hardware" products, these information cannot be directly applied to CF methods, because customers share similar interests on "hardware" products do not guarantee that they share the same level of interests on "software" products.

This work aims to address the "white-space recommendation" problem by combining statistical inference and transfer learning. By statistical inference, customer behaviors in the target category can be inferred by their behaviors in the related (source) category. For instance, if we have customers' purchasing history on "hardware", we can infer their interests on "software" based on correlations such as customers who purchased "RAID disks" have a high probability of purchasing "data backup software". By transfer learning, customer behaviors in the target category can be inferred by their content profiles. For instance, we can recommend "product management systems" to retailers, because "retailer" customers have a high probability of purchasing the "product management systems". And finally, these two methods are combined by a non-linear model to compute the final recommendation scores for the target customers on target products.

3. Related work

Recommender systems have been widely adopted in commercial sales and services [5,12,16,20,24]. And many solutions [12,16,35,36] have been proposed to solve the top-N recommendation problem, which is a common problem in commercial sales and services. However, the above methods only consider the case that customers have historical purchase records in the target product domain, and do not support cold-start recommendation, i.e., those methods cannot deal with "white space" customer problem as defined in this paper.

The cold-start issue is one of the main challenges to existing collaborative filtering (CF)-based recommender systems [1,4,15,18,25,29]. Some recent papers adopt auxiliary information to help improve recommendation accuracy when rating data for cold-start customers are extremely sparse [15,18,22,25,28,29]. Melville et al. [18] proposed a hybrid framework which combines content-based predictor and CF to address the cold-start issue. Schein et al. [25] proposed a two-way aspect model to combine PLSA-based CF method with a person/actor aspect model to make recommendations based on these content features of movies. Zhou et al. [29] developed a decision tree based method to learn user preferences and proposed a functional matrix factorization method to extract user profiles from the interview process on decision trees. Lin et al. [15] adopted the twitter followers of an app as the profile of the new app, and applied Latent Dirichlet Allocation (LDA) to generate latent groups and predict user's interest on apps based on latent groups. Rong et al. [22] adopted random walk method on graph of users and items to simulate the preference propagation processes among users, and then a Monte Carlo algorithm based on random walk was adopted to estimate user similarities for cold-start users. Zhang et al. [28] proposed an ensemble method, which first constructs weak prediction models in different contexts and then adopts a co-training strategy to allow weak model to learn from the other models. The above methods adopt content information to generate recommendations when rating data are not available, in which recommendation accuracies will suffer due to the limitation of content information [18].

Recent work in cross-domain collaborative filtering (CDCF) [13] adopts user rating data from related domains to make recommendations when user rating data in the target domain is scarce. Sahebi et al. [23] empirically studied how cross-domain recommendation performs in different cold-start scenarios. And their results confirmed that cross-domain recommendation can help in the cold-start problem. Li et al. [14] proposed a rating-matrix generative model for cross-domain collaborative filtering, which can fill the missing ratings for both existing and new users by combining rating data from multiple domains. Pan et al. [19] proposed a coordinate system transfer method to address the data sparsity problem in a target domain by transferring knowledge of both users and items from auxiliary data sources. Berkovsky et al. [2] proposed several mediation approaches for aggregating user ratings in different application domains, such that user ratings from different domains will help address the data sparsity issue in one domain. However, most of existing CDCF works focused on improving recommendation quality in the target domain through more comprehensive user features from other domains or address the data sparsity issue in the target domain through user ratings from other domains. Few of them target at the "white-space" recommendation problem. Moreover, as pointed out by Hu et al. [10], many existing CDCF methods will suffer from the "blind-transfer" issue when ratings of target users in target domain are unavailable and recommendations are solely determined by the user data in other domains.

Recently, the work of Hu et al. [10] was proposed to address the "blind transfer" issue in CDCF. However, their method was specially designed for rating prediction problem, so that many of their techniques are not applicable for top-N recommendation (the targeted problem in this work). For instance, conditional probabilities are adopted to model customer rating distributions, which is not applicable in binary dataset because there are only positive samples in binary dataset (all other information should be considered unknown rather than being considered as negative samples). Gantner et al. [6] adopted user/item attributes information and mapped these attributes to the latent features in matrix factorization (MF) model, so that MF models trained on existing users/items can be applied to the new-user/item recommendation. Similar to our work, the relationship between user/item profiles and user interests are modelled. However, content profiles of customers, which are of key values in the recommendation process, are not considered in their work.

Text analysis technique is another related topic to this work. Probabilistic latent semantic analysis was first proposed for text analysis [8] and later adopted in collaborative filtering by Hofmann [9]. Besides PLSA, a variety of other techniques [38–40] have been proposed for text analysis problems recently. Yu et al. [38] proposed text categorization models using both back-propagation neural network (BPNN) and modified back-propagation neural network (MBPNN). And their experiments showed that models using MBPNN can outperform BPNN in news classification. Zhang et al. [39] proposed a hybrid relations analysis approach, which can integrate both semantic relations and co-occurrence relations for topic detection. And their experiments showed that such combination can improve the quality of topic detection. Daud et al. [40] proposed time topic modeling approach, which can exploit semantic structure of words among the authors of research papers and meanwhile utilize time factor to find out dynamic interests of researchers. Different from the above work that targeted on only text analysis, the proposed PLSA-based transfer learning method can discover latent topics shared between customer purchase behaviors and customer content profiles, i.e., the proposed method can transfer knowledge from customer content profiles to their purchase interests. This transfer learning feature makes it possible to recommend products to "white space" customers by knowledge from both their content profiles.

In summary, the proposed method differs from the existing work in the following aspects. (1) The proposed method does not require any rating data for the target customers on target items, an essential property to support "white-space" customers. (2) The proposed method leverages both auxiliary information and user rating data from other related product categories, and effectively improves the recommendation accuracy. (3) Different from existing CDCF methods, which suffer from the "blind-transfer" problem, the proposed method models the relationship between the target category and related category in both statistical inference and PLSA-based transfer learning, so that recommendations will not solely rely on customer behaviors in the related category. (4) The proposed PLSA-based transfer learning method can detect both shared and non-shared latent topics, and recommendations are only generated by shared latent topics, so that "noisy" information provided by irrelevant part of content profiles will not affect the recommendation accuracy.

4. Algorithm design

This section describes the proposed SalesExplorer's recommendation algorithm. As illustrated in Fig. 1, there are three key components in the proposed method: (1) a statistical inference method, which infers customer behaviors in the target category based on their behaviors in the related source category; (2) a transfer learning method based on probabilistic latent semantic analysis, which analyzes the co-occurrence of "words" and "purchase of products" in the target category through shared latent topics between customer purchasing behaviors and their content profiles; and (3) a non-linear combination model based on kernel logistic regression, which efficiently combines the recommendation scores from the above two methods to achieve better recommendation accuracy.

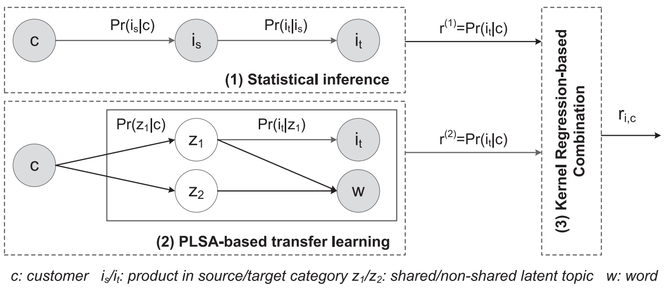

Fig. 1. SalesExplorer: algorithm overview.

4.1. Statistical inference

A common approach to infer customers' interests in the target category is based on their purchasing history in the source category. Methods like cross-domain collaborative filtering (CDCF) have been proposed to achieve this goal. However, two main limitations exist in CDCF methods: (1) CDCF methods will generate recommendations only based on information from the source category, i.e., "blind transfer" [10]; and (2) customers usually have few ratings in many commercial scenarios, and as a result, CDCF models may overfit when only considering similar neighbors in the source category.

To overcome the above limitations, we propose a statistical inference method by adopting customers' behavior in the source category and the statistical relationship between products in the source category and those in the target category. Specifically, we compute two pieces of statistical information: (1) customers' probabilities of purchasing products in I^s ($\Pr(i_s|c)$), i.e., customer behaviors in the source category; and (2) the probability of each product in I^t is purchased given that each product in I^s is purchased ($\Pr(i_t|i_s)$) based on all customer data, i.e., transitive relationship between the source category and the target category. Then, the recommendation score can be jointedly computed by combining the above two pieces of statistical information.

Let $c \in C^t$ be a target customer and $i \in I^t$ be a product in the target category. Then, by assuming that customer purchasing behaviors are independent, the recommendation score based on statistical inference can be jointly computed as follows:

$$r^{(1)} = \Pr(i|c) = \sum_{j \in I^s_c} \Pr(i|j)\Pr(j|c) \quad (1)$$

where I^s_c is the set of products purchased by c in source category s. $\Pr(j|c)$ is the probability of c purchasing product j in the source category. $\Pr(i|j)$ is the probability of customers purchasing product i in the target category given that they have already purchased j in the source category. Therefore, $\Pr(j|c)$ and $\Pr(i|j)$ can be computed as follows:

$$\Pr(j|c) = \frac{n(j,c)+\delta}{\sum_{j' \in I^s}(n(j',c)+\delta)} \quad (2)$$

$$\Pr(i|j) = \frac{n(i,j)+\delta}{\sum_{j' \in I^s}(n(i,j')+\delta)} \quad (3)$$

where $n(i, j)$ is the number of times that product $i \in I^t$ and product $j \in I^s$ have been purchased together by customers in $C - C_{tar}$, and $n(j, c)$ is the number of times that customer c has purchased product j. δ is the Laplacian smoothing parameter to address the "zero-probability" issue in the statistical inference, which is set to 1 in this paper.

Since each $\Pr(i|j)$ can be obtained from all the customer data, recommendations to the target customers are based on the decisions of all other customers. Therefore, the recommendation results will not only depend on customer behaviors in the source category but also depend on the relationship between source category and target category. Therefore, the method will not suffer from the "blind transfer" issue. Moreover, since the recommendation process utilizes all customer data rather than a fixed number of neighbors, the overfitting problem can also be addressed.

In the computation of $r^{(1)}$, the $\Pr(j|c)$ and $\Pr(i|j)$ should be computed in advance, and the computation complexities of which are $O(nm_s)$ and $O(m_s m_t)$, respectively (n is the number of customers in $C - C_{tar}$, m_s (m_t) is the number of products in the source (target) category). Then, the complexity of computing $r^{(1)}$ for all target customers in the target category is $O(n' m_s m_t)$, where n' is the number of target customers.

4.2. PLSA-based transfer learning

Content profiles of customers can also be adopted to determine the latent characteristics of customers, which can further be utilized to generate recommendations for customers by transferring knowledge learned in the content domain to product purchasing behavior. This is especially reliable for enterprise customers, since their content profiles are usually accurate and informative.

Example 1 (Content profile of IBM)**.** "International Business Machines Corporation (IBM) provides information technology (IT) products and services worldwide. The company's Global Technology Services segment provides IT infrastructure and business process services, including ... Its Software segment offers middleware and operating systems software, such as ... The company's Systems and Technology segment provides computing power and storage solutions..."

As presented in Example 1, customer profiles are textual information that describe the characteristics of the customers. For instance, given the content profile of "IBM", we can know that IBM will belong to the latent topic "IT" because the words "infrastructure", "software", "systems", "storage", etc., frequently appear in the description of IBM. Then, by analyzing how other customers that belong to the latent topic "IT" will like products in the target category, we can infer IBM's interest on products in the target category. In the above analysis, the main challenge is how to model the relationship between content profiles and products in the target category. We achieve this by a probabilistic latent semantic analysis-based transfer learning method to transfer the knowledge learned from customers' content profiles to customers' purchasing behaviors in the target product category.

Probabilistic latent semantic analysis [8] is one of the most popular models for analyzing latent topics in the text domain. In the classic PLSA method, the relationship between documents and words can be modeled as follows:

$$\Pr(w|d) = \sum_{z} \Pr(w|z)\Pr(z|d) \quad (4)$$

where d, w and z stand for documents, words and latent topics, respectively. If we view the content profiles of customers as "documents" and products in the target category as "words", then the relationship between customers and products in the target category can be computed as follows:

$$r^{(2)} = \Pr(i|c) = \sum_{z} \Pr(i|z)\Pr(z|c) \quad (5)$$

However, since target customers have no rating data for these target products, $\Pr(i|z)$ and $\Pr(z|c)$ will all be zero if we train the above model as in the classic PLSA model. To address this issue, we adopt a transfer learning method to transfer the knowledge obtained from customers' content profiles to the purchasing interests in the target category, in which a set of shared latent topics are regarded as the "media" of the knowledge transfer.

As illustrated in component (2) of Fig. 1, there are two sets of latent topics. $Z_1 = \{z_1, \ldots, z_{k_1}\}$ is the set of shared latent topics which can transfer knowledge from the content domain to the target product category. $Z_2 = \{z'_1, \ldots, z'_{k_2}\}$ is the set of latent topics which are only related to the content domain. The reason why we adopt Z_2 here is because not all topics from the content domain are related to the target product category. Therefore, the adoption of Z_2 can remove "noisy" topics, which can improve the accuracy of the transfer learning. Based on the above latent topics, the joint likelihood of the whole dataset is:

$$\Pr(C) = \prod_{c \in C} \left[\prod_{w \in V} \left(\sum_{z \in Z} \Pr(w|z) \Pr(z|c) \right)^{n(c,w)} \prod_{i \in I^t} \left(\sum_{z \in Z_1} \Pr(i|z) \Pr(z|c) \right)^{n(c,i)} \right] \tag{6}$$

where $Z = Z_1 \cup Z_2$ is the set of all latent topics and V is the vocabulary. In the above Eq. (6), the likelihood depends on $\Pr(i|z)$, $\Pr(w|z)$ and $\Pr(z|c)$, which are parameters that need to be estimated. Since it is difficult to directly estimate the parameters from Eq. (6), we optimize the lower bound of its log-likelihood as follows:

$$L_c = \sum_{c \in C} \sum_{w \in V} n(c, w) \sum_{z \in Z} \left[\log \left(\Pr(c) \Pr(w|z) \Pr(z|c) \right) \right] + \sum_{c \in C} \sum_{i \in I^t} n(c, i) \sum_{z \in Z_1} \left[\log \left(\Pr(c) \Pr(i|z) \Pr(z|c) \right) \right] \tag{7}$$

By introducing the Lagrange Multipliers, the optimization of L_c can be expressed as follows:

$$H = L_c + \alpha \sum_{z \in Z_1} \left(1 - \sum_{i \in I^t} \Pr(i|z) \right) + \beta \sum_{z \in Z} \left(1 - \sum_{w \in V} (\Pr(w|z)) \right) + \gamma \sum_{c \in C} \left(1 - \sum_{z \in Z} \Pr(z|c) \right) \tag{8}$$

where the last three parts are Lagrange Multipliers to guarantee parameters ranging in [0, 1]. H can be optimized using the standard EM method. In the E-step, we compute the posterior distribution of the hidden variables as follows:

$$\Pr(z|c, w) = \frac{\Pr(w|z) \Pr(z|c)}{\sum_{z' \in Z} \Pr(w|z') \Pr(z'|c)} \quad (z \in Z) \tag{9}$$

$$\Pr(z|c, i) = \frac{\Pr(i|z) \Pr(z|c)}{\sum_{z' \in Z_1} \Pr(i|z') \Pr(z'|c)} \quad (z \in Z_1) \tag{10}$$

And in the M-step, we obtain the new optimal parameters based on the current estimation of the hidden variables. For $\Pr(i|z)$ $(z \in Z_1)$, the new optimal value should satisfy

$$\frac{\partial H}{\partial \Pr(i|z)} = \sum_{c} n(c, i) \frac{\Pr(z|i, c)}{\Pr(i|z)} - \alpha = 0 \tag{11}$$

$$\frac{\partial H}{\partial \alpha} = 1 - \sum_{z} \Pr(i|z) = 0 \tag{12}$$

By solving the above two equations, we have

$$\Pr(i|z) = \frac{\sum_c n(c, i) \Pr(z|i, c)}{\sum_{i'} \left(\sum_c n(c, i') \Pr(z|i', c) \right)} \quad (z \in Z_1) \tag{13}$$

Similarly, for $\Pr(w|z)$ $(z \in Z)$, we have

$$\Pr(w|z) = \frac{\sum_c n(c, w) \Pr(z|w, c)}{\sum_{w'} \left(\sum_c n(c, w') \Pr(z|w', c) \right)} \quad (z \in Z) \tag{14}$$

However, for $\Pr(z|c)$, we should separate Z_1 and Z_2 during the optimization. For $z \in Z_1$, we have

$$\frac{\partial H}{\partial \Pr(z|c)} = \sum_{i} n(c, i) \frac{\Pr(z|i, c)}{\Pr(z|c)} + \sum_{w} n(c, w) \frac{\Pr(z|w, c)}{\Pr(z|c)} - \gamma = 0 \tag{15}$$

For $z \in Z_2$, we have

$$\frac{\partial H}{\partial \Pr(z|c)} = \sum_{w} n(c, w) \frac{\Pr(z|w, c)}{\Pr(z|c)} - \gamma = 0 \tag{16}$$

Again, by integrating $\partial H / \partial \gamma = 0$ and solving Eq. (15) and Eq. (16), we have

$$\Pr(z_1|c) = \frac{\sum_i n(c, i) \Pr(z_1|c, i) + \sum_w n(c, w) \Pr(z_1|c, w)}{N_r} \tag{17}$$

$$\Pr(z_2|c) = \frac{\sum_w n(c, w) \Pr(z_2|c, w)}{N_r} \tag{18}$$

where $z_1 \in Z_1$, $z_2 \in Z_2$, and N_r can be computed as follows:

$$\sum_{z' \in Z} \sum_{w} n(c, w) \Pr(z'|c, w) + \sum_{z' \in Z_1} \sum_{i} n(c, i) \Pr(z'|c, i). \tag{19}$$

In the computation of $r^{(2)}$, the PLSA-based transfer learning model should be trained in advance, and the computation complexity of which is $O(l((k_1 + k_2)n\nu + k_2 nm_t))$ (l is the number of iterations in EM and ν is the number of words in the vocabulary). The training for the PLSA-based model can be performed offline, so that the high computation complexity will not be an issue. Then, after training the PLSA model, the $r^{(2)}$ score can be computed with the complexity of $O(k_1 nm_t)$, which is similar to $r^{(1)}$.

4.3. Kernel logistic regression-based combination method

In order to achieve higher accuracy, we propose a kernel logistic regression-based model to combine the two recommendation scores — $r^{(1)}$ (Eq. (1)) is based on target customers' behavior in the source product category, and $r^{(2)}$ (Eq. (5)) is based on the content profiles of the target customers.

4.3.1. Combination model

The goal of the proposed method is to support enterprise sales, for which it is critical to identify the most relevant products from the target category to enterprise customers, in other words, a top-N recommendation scenario. It is then important to determine the prioritized order of the relevant products tailored to each target customer, instead of providing absolute scores to every single product from the target product category. Therefore, kernel logistic regression (KLR) model is adopted to predict the combined score for a given customer on a given product, because 1) logistic regression is an effective model for binary classification, i.e., predicting whether a customer will be interested in a product or not; 2) predictions with higher scores are more likely to happen in logistic regression, so that we can rank different products to each customer based on the combination scores; and 3) the decision boundary in kernel logistic regression can find non-linear relationships between random variables, so that the non-linear relationship between $r^{(1)}$, $r^{(2)}$ and the final recommendation score can be effectively modeled.

Let f be a function for combining $r^{(1)}$ and $r^{(2)}$, then given a kernel function $K(r_i, r_j)$, f can be computed as follows:

$$r_{i,c} = f(\vec{r}) = f(\langle r^{(1)}, r^{(2)} \rangle) = \sum_{i=1}^{m} \theta_i K(\vec{r}_i, \vec{r}) \tag{20}$$

where m is the number of training samples in the dataset and θ_is are the parameters to estimate. In this paper, we choose the polynomial kernel, in which $K(X, Y) = (aX^T Y + b)^d$ (a is the slope, b is the constant term, and d is the polynomial degree). Note that, values of a, b and d may vary in different datasets, and the optimal values are chosen by cross validation in this paper.

In the above model, the parameters that need to be trained are θ_is. After obtaining θ_is, the recommender systems can integrate $r^{(1)}$ and $r^{(2)}$ scores for target product by computing the kernel values with all training samples.

4.3.2. KLR training

In kernel logistic regression, we need to minimize the following equation [27,30]:

$$H = -\frac{1}{N}\sum_{i=1}^{m} \log s(y_i f(x_i)) + \frac{\lambda}{2}||f||^2_{\mathcal{H}_k} \tag{21}$$

where $y_i \in \{0, 1\}$ indicates whether a customer in $C - C_{tar}$ purchased a specific product or not, $s(x) = 1/(1 + \exp\{-x\})$ and λ is the coefficient for the L2-regularization term.

The gradient descent method can be adopted to minimize the above Eq. (21). However, there may be large and ever-growing size of training examples in enterprise sales segments, so that standard gradient descent method will scale and adapt poorly in such case. Therefore, stochastic gradient descent (SGD) method [3], which can optimize toward gradients using subsets of the training data, is adopted in this paper. Moreover, SGD can update the model parameters in an online fashion, so that the combination model can be updated incrementally when new purchases are made by the customers.

The following Algorithm 1 describes how to obtain $\Theta = \{\theta_1, \ldots, \theta_m\}$ using SGD.

Algorithm 1 SGD(Θ, R, H).

Require: $R = \{\vec{r}_1, \ldots, \vec{r}_m\}$ is the set of recommendation scoresin the training data, H is the optimization goal.

Randomly choose values in Θ;
while Θ has not converged **do**
  for each $\vec{r}_i = \langle r_i^{(1)}, r_i^{(2)} \rangle \in R$ **do**
    for each $j = 1$ to m **do**
      $\theta_j \leftarrow \theta_j - \alpha \frac{\partial H}{\partial \theta_j}$;
    end for
  end for
end while

In Algorithm 1, α is the learning rate for the SGD procedure. And the partial derivative — $\partial H/\partial \theta_j$ for each training sample $\vec{r}_i$ can be computed as follows:

$$\frac{\partial H}{\partial \theta_j} = (1 - s(-y_i f(\vec{r}_i)))y_i K(\vec{r}_i, \vec{r}_j) + \lambda\theta_j K(\vec{r}_j, \vec{r}_j) \tag{22}$$

Note that, the learning parameters — α and λ can be obtained by cross validation.

4.3.3. Efficiency enhancement

After obtaining Θ, the recommender system can compute the final recommendation scores for all target customers based on Eq. (20). However, it should be noted that Θ will be trained across all training samples and the computation complexity for training Θ is $O(m^3)$, so that efficiency issue will arise when m is large. To address this issue, we adopt the feature of SGD that model parameters can be trained from subsets of training samples. Note that, a method named IMV [30] has been proposed to address the similar issue by selecting a subset of training samples to obtain suboptimal model, in which data samples are iteratively selected as the ones that can minimize the optimization function. But during the selection procedure in their method, gradient descent is performed for each data sample, which is computationally expensive. In this work, a random selection method is proposed to address the efficiency issue. To select a subset of training samples from the historical data, the proposed random selection method works as described in Algorithm 2.

Algorithm 2 RandomSelection(M, I, a).

Require: M is the set training data ($|M| = m$), I is the set of all products, and a is the predefined percentage of training samples for selection.

$M^* = \emptyset$;
for each $i \in I$ **do**
  Let M_i be the purchase records for product i in M;
  Randomly select a records from M_i as M_i^*;
  $M = M \cup M_i^*$;
end for
return M;

In Algorithm 2, we randomly select purchase records for each product individually, so that training samples will not be biased for each individual product during the random selection process. Note that, after adopting the proposed random selection method, the computation complexity for training Θ is reduced to $O(a^3m^3)$. Since a can be chosen as a small percentage of training samples, e.g., 20% or 30%, so that the overall computation overhead will largely reduced. Moreover, the proposed random selection method can also benefit the prediction step, because less training samples are considered for prediction.

In the above KLR-based combination method, the computation complexity of training is $O(a^3m^3)$ as analyzed above. And the computation complexity of model prediction is $O(amn'm_t)$. Since $a*m$ is a constant, so that the overall computation complexity is linear to $n'm_t$ (the product of the number of targeted customers and the number of items in the target domain). Therefore, it is applicable even in online cases.

4.4. SalesExplorer: the overall flow

After obtaining $r^{(1)}$ and $r^{(2)}$ for each customer given each product, we can train the kernel logistic regression-based combination model as described previously. Then, the recommender system can compute the final recommendation scores for all target customers, and recommend top N products with highest scores to each customer. The overall flow of the proposed SalesExplorer algorithm is described in Algorithm 3.

Algorithm 3 Top_N_Recommendation(C, P, I^s, I^t).

Require: $C_{tar} \subset C$ is the set of target customers.

for each $c \in C_{tar}$ **do**
  Generate c's profile $\{p_c, I_c^s, I_c^t\}$;
end for
for each $i \in I^t$ and $c \in C$ **do**
  $r_{i,c}^{(1)} = \sum_{j\in I_c^s} \Pr(i|j)\Pr(j|c)$ Eq. (1);
  $r_{i,c}^{(2)} = \sum_{z\in Z_1} \Pr(i|z)\Pr(z|c)$ Eq. (5);
end for
Run Algorithm 2 to select a subset of training samples;
Run Algorithm 1 to train Θ using selected customer data in $C - C_{tar}$;
for each $i \in I^t$ and $c \in C_{tar}$ **do**
  $\vec{x}_{i,c} = \langle r_{i,c}^{(1)}, r_{i,c}^{(2)} \rangle$;
  $r_{i,c} = f(\vec{x}_{i,c}) = \sum_{j=1}^{am} \theta_j K(\vec{r}_j, \vec{x}_{i,c})$;
end for
Recommend the top N products in I^t with highest $r_{i,c}$ to each customer in C_{tar};

Table 1
Dataset description.

#records	#customers	#devices	#services
~ 4 million	9524	2906	3733

5. Experiments

This section evaluates the proposed algorithm using a real-world enterprise sales dataset. We first assess the recommendation accuracy of the proposed algorithm as a function of the sizes of the shared/non-shared latent topics. Next, we compare the recommendation accuracy of the proposed methods with three other ones: (1) popularity-based method (Popularity); (2) aspect model based method (Aspect), which addresses the cold-start user problem by a two-way aspect model [25]; and (3) topic-bridged PLSA method (TPLSA), which can transfer knowledge from other domains when rating data are unavailable for the target domain [26]. The TPLSA method was proposed for cross-domain classification, but the model can also be applied in the "white-space" recommendation scenario as PLSA is one of the popular techniques in collaborative filtering [9]. It should be noted that the compared methods are chosen because they can be applied to the "white-space" recommendation problem discussed in this work. A variety of other cold-start recommendation methods or CDCF methods, which cannot be applied in the same scenario, are compared and discussed in the related section. At last, we analyze the performance of efficiency enhancement in the proposed method.

5.1. Experiment setup

5.1.1. Dataset description

The following experiments adopt a real-world enterprise sales dataset, which contains approximately 4 million multi-year worldwide contract records. In the dataset, customers are enterprise companies or organizations, each of which has a content profile indicating its primary business. For the "white-space" customer setting, we aim to recommend "maintenance services" to customers who have no purchase history of any maintenance services before. A related category, device purchase histories of customers, is adopted to help recommendation in the target category (maintenance service) (Table 1).

There are totally 9524 customers in the dataset, and we randomly select 80% of customers as the training set and keep the remaining 20% of customers as the test set. In the target product category, there are totally 3733 different maintenance services. In the related (source) category, there are totally 2906 different hardware devices. We assume that the purchase histories of test customers in the target category are completely unknown. Therefore, recommendations to test customers are generated based on their content profiles and purchase histories in the related (source) category.

5.1.2. Implementation and evaluation metrics

The proposed method is implemented using Java on a workstation equipped with an eight-core CPU (3.4 GHz) and 16GB memory. All the model parameters in our methods as well as in the compared methods are obtained via 10-fold cross validation. For all the experiments, we run them for 5 times and present the average results.

In this section, we evaluate the accuracy and efficiency of the proposed SalesExplorer recommendation algorithm. For recommendation accuracy, the following evaluation metrics are adopted in the experiments:

- **Precision:** is the fraction of recommended products that are purchased by the customers, which is defined as follows:

$$Precision = \frac{|I_r \cap I_c|}{|I_r|}$$

where I_r and I_c are the set of recommended products and the set of products purchased by customer c, respectively.
- **Receiver Operating Characteristic (ROC) curve:** illustrates the performance of binary recommender systems when the number of recommendations to each customer varies. The x-axis is the false positive rate (FPR) and the y-axis is the true positive rate (TPR), which are defined as follows [17]:

$$FPR = \frac{\#FP}{\#FP + \#TN}, \qquad TPR = \frac{\#TP}{\#TP + \#FN}$$

where *FP, TN, TP, FN*, are "false positive", "true negative", "true positive" and "false negative" recommendations, respectively.
- **Discounted Cumulative Gain (DCG):** can measure ranking quality, in which the DCG value will be higher if relevant products are ranked higher in the recommendation lists. The DCG values can be computed as follows [11,17]:

$$DCG = rel_1 + \sum_{i=2}^{|I_r|} \frac{rel_i}{\log(i)}$$

where i is the position of each recommended product in the recommendation list. rel_i indicates whether product recommended at position i is relevant or not, i.e., $rel_i = 1$ if the customer purchased the i-th product in the recommendation list and $rel_i = 0$ otherwise.

For *Precision* and *DCG*, higher values indicate better recommendation accuracy. And for *ROC* curve, curves on the left top indicate better recommendation accuracy.

5.2. Sensitivity of model parameters

As shown in Section 4, the number of latent topics in the proposed PLSA-based transfer learning method is a key factor for determining the recommendation accuracy of the proposed method. In this experiment, we evaluate how recommendation accuracy changes with the sizes of shared latent topics ($|Z_1| = k_1$) and non-shared latent topics ($|Z_2| = k_2$).

In the proposed method, if k_1 is chosen too small, the transfer learning process will underfit due to insufficient latent topics. On the other hand, if k_1 is chosen too large, the transfer learning process will overfit because customer data are sparse so that the PLSA model will overfit by the few purchasing records. Fig. 2 shows the recommendation accuracy when k_1 changes from 2 to 20 (k_2 and N are fixed to 3 and 10, respectively). We can see from the results that the recommendation accuracy increases when k_1 increase from 2 to 10 and decreases afterwards. Therefore, the optimal k_1 should be around 10 for this dataset. Note that the best k_1 value will differ for different datasets, so k_1 should be carefully chosen through cross validation before applying the proposed method.

Fig. 3 shows the recommendation accuracy when k_2 changes from 1 to 10 (k_1 and N are fixed to 10 and 10, respectively). We can see from the results that the recommendation accuracy increases when k_2 increases from 1 to 3. But when $k_2 > 3$, the recommendation accuracy only slightly changes. This is because more non-shared latent topics will only influence the accuracy of $\Pr(w|z_2)$, not $\Pr(i|z_1)$. But when k_2 is too small, i.e., 1 or 2, model underfitting will occur, which will cause $\Pr(z|c)$ to be inaccurate. Thus, the recommendation accuracy will be affected. Therefore, the optimal k_2 should be around 3 for this dataset.

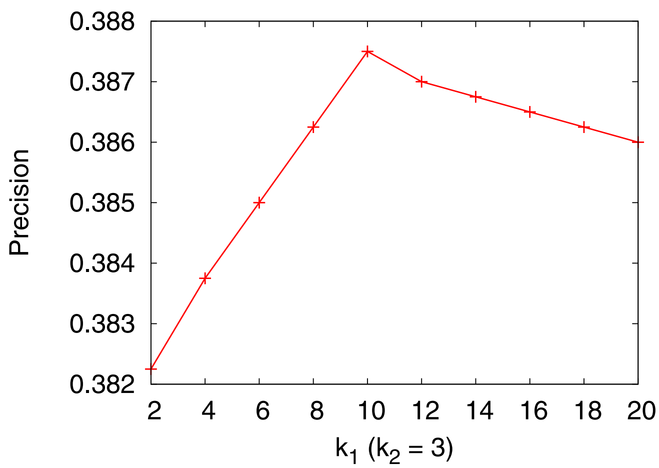

Fig. 2. Recommendation precision variations when k_1 changes from 2 to 20 ($N = 10,\ k_2 = 3$).

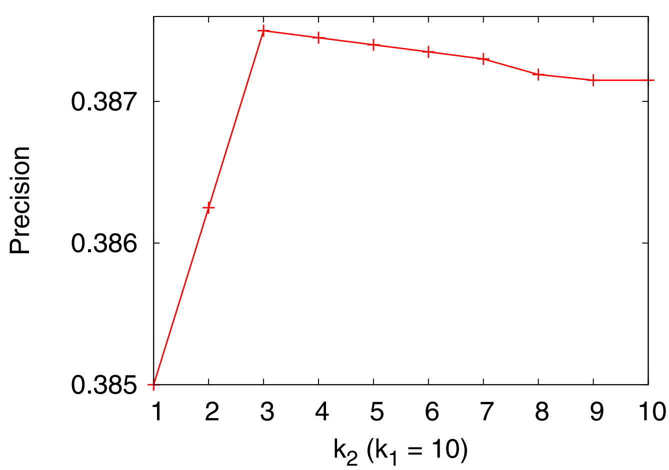

Fig. 3. Recommendation precision variations when k_2 changes from 1 to 10 ($N = 10,\ k_1 = 10$).

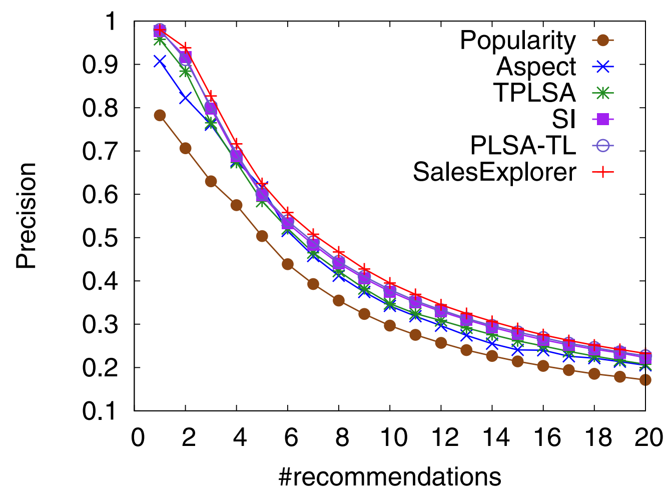

Fig. 4. Precision comparison between the proposed method and three other methods.

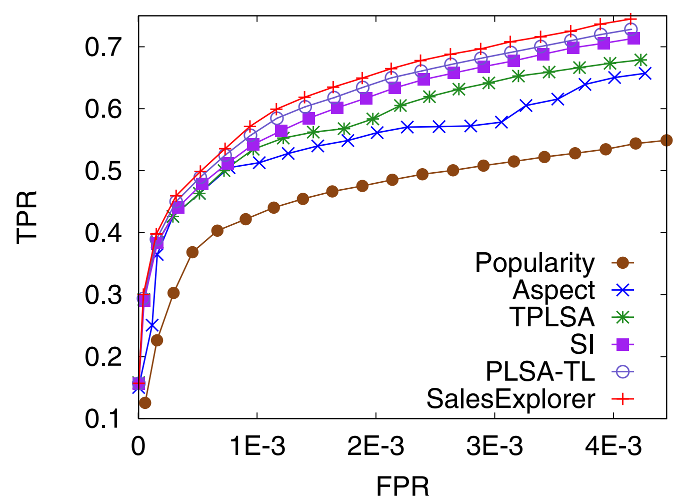

Fig. 5. ROC curve comparison between the proposed method and three other methods.

5.3. Accuracy comparison

Fig. 4 shows the precision comparison between the proposed method and the other three methods. In this comparison, we set $k_1 = 10$ and $k_2 = 3$. For the Aspect method and the TPLSA method, the number of latent topics is set to 10 and both methods achieve near optimal results. As shown in Fig. 4, the Aspect method yields lower recommendation accuracy compared with the TPLSA method and the proposed method. This further confirms that pure content-based recommendation methods are not as accurate as CDCF methods. Between the algorithms which only adopt customer purchasing behavior from the other related category, our statistical inference (SI) method can consistently outperform the TPLSA method, by 5.38% on average. And between algorithms which only adopt content profiles of customers, our PLSA-based Transfer Learning (PLSA-TL) method consistently outperforms the Aspect method, by 9.64% on average. These results indicate that both our statistical and content-based methods outperform the state-of-the-art ones. In addition, when N (the number of recommended products) varies from 1 to 20, our proposed method outperforms the Popularity method, Aspect method and TPLSA method by 32.14%, 13.13% and 9.85%, respectively, on average. These results demonstrate that the SaleExplorer method, which combines the SI method and PLSA-TL method with a kernel logistic regression-based model, can further improve recommendation accuracy.

Figs. 5 and 6 show the ROC and DCG comparisons of the proposed method and other three methods. The proposed method locates at the left top of the ROC curve in Fig. 5, which confirms that the proposed method outperforms the other three methods. Similar phenomenon can be observed from the DCG curves in Fig. 6. The reasons why the proposed method can achieve more accurate recommendations are: (1) the proposed statistical inference method can alleviate the aforementioned overfitting problem even when target customer ratings are few in the source category, because the method adopt the statistical information of all customers; (2) the proposed PLSA-based transfer learning method can find accurate shared latent topics between content domain and product purchasing behavior, which can improve the accuracy compared with other methods that treat all topics as shared topics; and (3) the proposed kernel logistic regression-based combination method can further increase the recommendation accuracy by utilizing both the customers' purchasing behaviors in other category and their content profiles.

5.4. Efficiency analysis

As discussed in Section 4.3.3, the computation complexity of the proposed kernel logistic regression-based combination method is $O(m^3)$, where m is the number of training samples. And the pro-

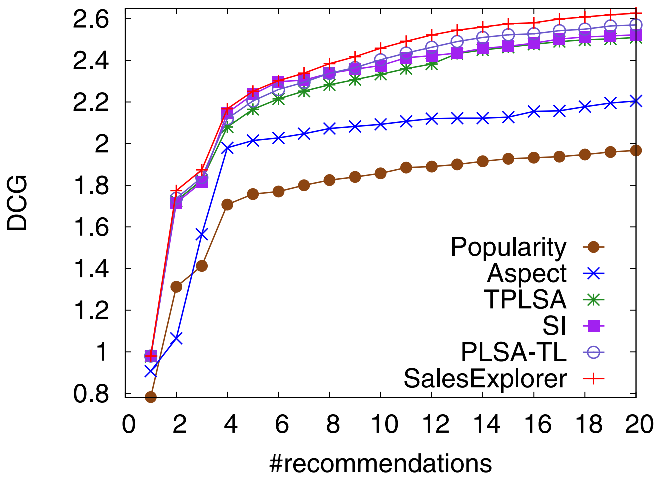

Fig. 6. DCG comparison between the proposed method and three other methods.

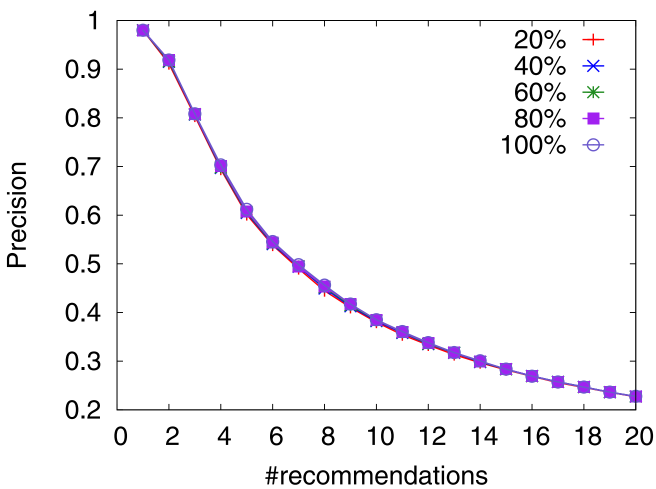

Fig. 7. Tradeoff between accuracy and efficiency.

posed *RandomSelection* method (Algorithm 2) can largely reduce the computation overhead by selecting subset of training samples for modeling training and prediction. In the following experiments, we first analyze the accuracy/efficiency tradeoff with different sizes of training samples.

Fig. 7 shows how the recommendation precision varies when the fraction of training samples — a changes from 20% to 100%. We can see from the results that the precision variations are negligible. Compared with 100% of training samples (optimal case), the recommendation precision only degrades by 1.06% on average even when we only select 20% of training samples. This reflects that the combination model is robust when the number of training samples are sufficient, which is no more than 20% in our dataset. Moreover, the results also indicate that the proposed *RandomSelection* method is effective.

Besides accuracy analysis, we also analyze how the computation efficiency is enhanced when the fraction of training samples — a changes from 20% to 100% in Fig. 8. As shown in the results, the computation time for both model training and prediction increase super-linearly when a increases from 20% to 100%, i.e., computation efficiencies increase super-linearly when a decreases. The above results indicate that the proposed *RandomSelection* method can enhance both model training and prediction efficiencies effectively.

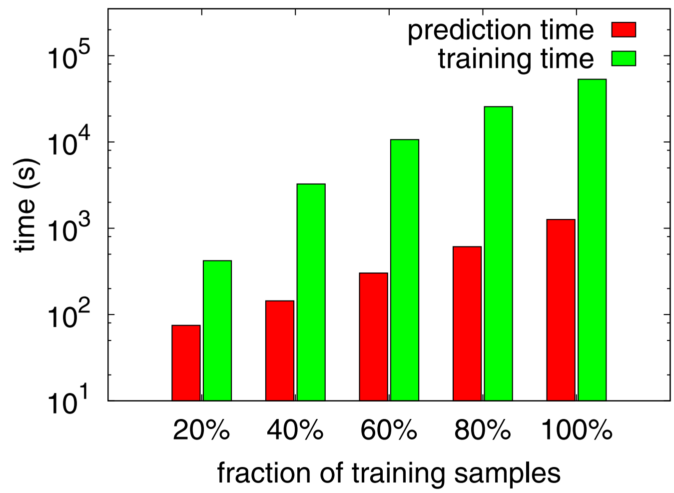

Fig. 8. Model training and predication efficiency vs. fraction of training samples.

6. Conclusion and future work

Recommender systems have been widely adopted to assist sales and services in the enterprise market. Existing CF-based methods suffer from the "white-space" customer problem, a key limitation to effectively support customer discovery and conversion. This paper presents SalesExplorer, a new recommendation algorithm that employs the knowledge from the customers' content profiles as well as their prior purchase records from other product categories. Then, the two kinds of knowledge are integrated via an efficient kernel logistic regression-based combination model, which is applicable in large-scale dataset and online cases through a suit of efficiency enhancement strategies. Experimental study using real-world enterprise sales data demonstrates that the proposed recommendation algorithm outperforms a baseline method and two state-of-the-art methods in recommendation accuracy.

One of the possible extensions to this work is to introduce privacy-preserving mechanism to SalesExplorer, because privacy issue is one of the key challenges in many recommender systems, especially in commercial recommender systems [31,37,41–43]. Existing privacy-preserving techniques, e.g., Homomorphic Encryption [32], can be potentially adopted to protect customer privacy without changing the algorithms in SalesExplorer.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under grant No. 61233016, the Open Project Program of Shanghai Key Laboratory of Data Science under grant No. 201609060004 and the National Science Foundation of USA under grant Nos. 1251257, 1334351 and 1442971.

References

[1] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: asurvey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 734–749.
[2] S. Berkovsky, T. Kuflik, F. Ricci, Cross-domain mediation in collaborative filtering, in: Proceedings of the 11th International Conference on User Modeling, UM '07, 2007, pp. 355–359. Springer.
[3] L. Bottou, Large-scale machine learning with stochastic gradient descent, in: Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT '10), 2010, pp. 177–186.
[4] P. Cremonesi, R. Turrin, Analysis of cold-start recommendations in IPTV systems, in: Proceedings of the 3rd ACM Conference on Recommender Systems, RecSys '09, 2009, pp. 233–236. ACM.
[5] D.M. Fleder, K. Hosanagar, Recommender systems and their impact on sales diversity, in: Proceedings of the 8th ACM Conference on Electronic Commerce, EC '07, 2007, pp. 192–199. ACM.

[6] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, L. Schmidt-Thieme, Learning attribute-to-feature mappings for cold-start recommendations, in: Proceedings of the 2010 IEEE International Conference on Data Mining, ICDM '10, 2010, pp. 176–185. IEEE.

[7] D. Goldberg, D. Nichols, B.M. Oki, D. Terry, Using collaborative filtering to weave an information tapestry, Commun. ACM 35 (12) (1992) 61–70.

[8] T. Hofmann, Probabilistic latent semantic indexing, in: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '99, 1999, pp. 50–57. ACM.

[9] T. Hofmann, Collaborative filtering via Gaussian probabilistic latent semantic analysis, in: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, SIGIR '03, 2003, pp. 259–266.

[10] L. Hu, J. Cao, G. Xu, J. Wang, Z. Gu, L. Cao, Cross-domain collaborative filtering via bilinear multilevel analysis, in: The 23rd International Joint Conference on Artificial Intelligence (IJCAI '13). IJCAI/AAAI, 2013.

[11] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of IR techniques, ACM Trans. Inf. Syst. 20 (4) (2002) 422–446.

[12] B. Kitts, D. Freed, M. Vrieze, Cross-sell: a fast promotion-tunable customer-item recommendation method based on conditionally independent probabilities, in: Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '00, 2000, pp. 437–446. ACM.

[13] B. Li, Cross-domain collaborative filtering: a brief survey, in: Proceedings of the 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence, ICTAI '11, 2011, pp. 1085–1086. IEEE.

[14] B. Li, Q. Yang, X. Xue, Transfer learning for collaborative filtering via a rating-matrix generative model, in: Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09, 2009, pp. 617–624. ACM.

[15] J. Lin, K. Sugiyama, M.-Y. Kan, T.-S. Chua, Addressing cold-start in app recommendation: latent user models constructed from twitter followers, in: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '13, 2013, pp. 283–292. ACM.

[16] G. Linden, B. Smith, J. York, Amazon.com recommendations: item-to-item collaborative filtering, IEEE Internet Comput. 7 (1) (2003) 76–80.

[17] C.D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.

[18] P. Melville, R.J. Mooney, R. Nagarajan, Content-boosted collaborative filtering for improved recommendations, in: 18th National Conference on Artificial Intelligence (AAAI '02), 2002, pp. 187–192. American Association for Artificial Intelligence.

[19] W. Pan, E.W. Xiang, N.N. Liu, Q. Yang, Transfer learning in collaborative filtering for sparsity reduction, in: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI '10, 2010.

[20] B. Pathak, R. Garfinkel, R. Gopal, R. Venkatesan, F. Yin, Empirical analysis of the impact of recommender systems on sales, J. Manage. Inf. Syst. 27 (2) (2010) 159–188.

[21] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, Grouplens: an open architecture for collaborative filtering of netnews, in: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW '94, 1994, pp. 175–186. ACM.

[22] Y. Rong, X. Wen, H. Cheng, A monte carlo algorithm for cold start recommendation, in: Proceedings of the 23rd International Conference on World Wide Web, WWW '14, 2014, pp. 327–336. ACM.

[23] S. Sahebi, P. Brusilovsky, Cross-domain collaborative recommendation in a cold-start context: the impact of user profile size on the quality of recommendation, in: User Modeling, Adaptation, and Personalization, Volume 7899 of Lecture Notes in Computer Science, 2013, pp. 289–295. Springer Berlin Heidelberg.

[24] J.B. Schafer, J.A. Konstan, J. Riedl, E-commerce recommendation applications, Data Min. Knowl. Discovery 5 (1–2) (2001) 115–153.

[25] A.I. Schein, A. Popescul, L.H. Ungar, D.M. Pennock, Methods and metrics for cold-start recommendations, in: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '02, 2002, pp. 253–260. ACM.

[26] G.R. Xue, W. Dai, Q. Yang, Y. Yu, Topic-bridged PLSA for cross-domain text classification, in: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '08, 2008, pp. 627–634. ACM.

[27] L. Zhang, R. Jin, C. Chen, J. Bu, X. He, Efficient online learning for large-scale sparse kernel logistic regression, in: Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI '12), 2012, pp. 1219–1225.

[28] M. Zhang, J. Tang, X. Zhang, X. Xue, Addressing cold start in recommender systems: asemi-supervised co-training algorithm, in: Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, 2014, pp. 73–82. ACM.

[29] K. Zhou, S.-H. Yang, H. Zha, Functional matrix factorizations for cold-start recommendation, in: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '11, 2011, pp. 315–324. ACM.

[30] J. Zhu, T. Hastie, Kernel logistic regression and the import vector machine, J. Computat. Graphical Stat. (2001) 1081–1088. MIT Press.

[31] J.F. Canny, Collaborative filtering with privacy, in: Proceedings of the 2002 IEEE Symposium on Security and Privacy, S&P' 02, 2002, pp. 45–57.

[32] T.E. Gamal, A public key cryptosystem and a signature scheme based on discrete logarithms, in: Proceedings of CRYPTO '84, 1984, pp. 10–18.

[33] R.D. Burke, Hybrid recommender systems: survey and experiments, User Model. User-Adapt. Interact. 12 (4) (2002) 331–370.

[34] M.J. Pazzani, D. Billsus, Content-based recommendation systems, in: The Adaptive Web Lecture Notes in Computer Science, 4321, 2007, pp. 325–341.

[35] D. Li, Q. Lv, X. Xie, L. Shang, H. Xia, T. Lu, N. Gu, Interest-based real-time content recommendation in online social communities, Knowl.-Based Syst. 28 (2012) 1–12.

[36] D. Li, Q. Lv, L. Shang, N. Gu, Item-based top-N recommendation resilient to aggregated information revelation, Knowl.-Based Syst. 67 (2014) 290–304.

[37] D. Li, C. Chen, Q. Lv, L. Shang, Y. Zhao, T. Lu, N. Gu, An algorithm for efficient privacy-preserving item-based collaborative filtering, Future Gener. Comput. Syst. 55 (2016) 311–320.

[38] B. Yu, Z.-b. Xu, C.-h. Li, Latent semantic analysis for text categorization using neural network, Knowl.-Based Syst. 21 (8) (2008) 900–904.

[39] C. Zhang, H. Wang, L. Cao, W. Wang, F. Xu, A hybrid term-term relations analysis approach for topic detection, Knowl.-Based Syst. 93 (2016) 109–120.

[40] A. Daud, Using time topic modeling for semantics-based dynamic research interest finding, Knowl.-Based Syst. 26 (2012) 154–163.

[41] H. Polat, W. Du, Privacy-preserving top-N recommendation on distributed data, J. Am. Soc. Inf. Sci. Technol. 59 (7) (2008) 1093–1108.

[42] D. Li, Q. Lv, H. Xia, L. Shang, T. Lu, N. Gu, Pistis: a privacy-preserving content recommender system for online social communities, in: The 2011 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011, pp. 79–86.

[43] I. Yakut, H. Polat, Estimating NBC-based recommendations on arbitrarily partitioned data with privacy, Knowl.-Based Syst. 36 (2012) 353–362.