International Journal of Research
Available at
https://edupediapublications.org/journals
Special Issue on Conference Papers

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 06
March 2018

Business To Business Communication In Social Recommendations

#1.K. Sai Prasanna B.Tech Student
#2.N.Swathi B. Tech Student
#3.K. Divya B.Tech Student
#4.K. Anusha B.Tech Student
#5.P. Akhila B.Tech Student

Department of CSE

MOTHER THERESSA COLLEGE OF ENGINEERING AND TECHNOLOGY, PEDDAPALLY, TS, INDIA

Abstract—In recent years, the boundaries between e-commerce and social networking have become increasingly blurred. Many e-commerce websites support the mechanism of social login where users can sign on the websites using their social network identities such as their Facebook or Twitter accounts. Users can also post their newly purchased products on microblogs with links to the e-commerce product web pages. In this paper, we propose a novel solution for cross-site cold-start product recommendation, which aims to recommend products from e-commerce websites to users at social networking sites in “cold-start” situations, a problem which has rarely been explored before. A major challenge is how to leverage knowledge extracted from social networking sites for cross-site cold-start product recommendation. We propose to use the linked users across social networking sites and e-commerce websites (users who have social networking accounts and have made purchases on e-commerce websites) as a bridge to map users’ social networking features to another feature representation for product recommendation. In specific, we propose learning both users’ and products’ feature representations (called user embeddings and product embeddings, respectively) from data collected from e-commerce websites using recurrent neural networks and then apply a modified gradient boosting trees method to transform users’ social networking features into user embeddings. We then develop a feature-based matrix factorization approach which can leverage the learnt user embeddings for cold-start product recommendation. Experimental results on a large dataset constructed from the largest Chinese microblogging service Sina Weibo and the largest Chinese B2C e-commerce website JingDong have shown the effectiveness of our proposed framework.

Index Terms—E-commerce, product recommender, product demographic, microblogs, recurrent neural networks

Ç

INTRODUCTION

IN recent years, the boundaries between e-commerce and

commerce websites such as eBay features many of the characteristics of social networks, including real-time status updates and interactions between its buyers and sellers. Some e-commerce websites also support the mechanism of social login, which allows new users to sign in with their existing login information from social networking services such as Facebook, Twitter or Google+. Both Facebook and Twitter have introduced a new feature last year that allow users to buy products directly from their websites by clicking a “buy” button to purchase items in adverts or other posts. In China, the e-commerce company Alibaba has made a strategic investment in SINA WEIBO1 where ALIBABA social networking have become increasingly blurred. E- product adverts can be directly delivered to SINA WEIBO users. With the new trend of conducting e-commerce activities on social networking sites, it is important to leverage knowledge extracted from social networking sites for the development of product recommender systems.

In this paper, we study an interesting problem of recommending products from e-commerce websites to users at social networking sites who do not have historical purchase records, i.e., in “cold-start” situations. We called this problem cross-site cold-start product recommendation. Although online product recommendation has been extensively studied before [1], [2], [3], most studies only focus on constructing solutions within certain e-commerce websites and mainly utilise users’ historical transaction records. To the best of our knowledge, cross-site cold-start product recommendation has been rarely studied before.

In our problem setting here, only the users’ social networking information is available and it is a challenging task to transform the social networking information into latent user features which can be effectively used for product recommendation. To address this challenge, we propose to use the linked users across social networking sites and e-commerce websites (users who have social networking accounts and have made purchases on e-websites) as a

Bridge to map users' social networking features to latent features for product recommendation. In specific, we propose learning both users' and products' feature representations (called user embeddings and product embeddings, respectively) from data collected from e-commerce websites using recurrent neural networks and then apply a modified gradient boosting trees method to transform users' social networking features into user embeddings. We then develop a feature-based matrix factorization approach which can leverage the learnt user embeddings for coldstart product recommendation.

We built our dataset from the largest Chinese microblogging service SINA WEIBO[2] and the largest Chinese B2C e-commerce website JINGDONG,[3] containing a total of 20,638 linked users. The experimental results on the dataset have shown the feasibility and the effectiveness of our proposed framework. Our major contributions are summarised below: We formulate a novel problem of recommending products from an e-commerce website to social networking users in "cold-start" situations. To the best of our knowledge, it has been rarely studied before.

We propose to apply the recurrent neural networks for learning correlated feature representations for both users and products from data collected from an e-commerce website.

We propose a modified gradient boosting trees method to transform users' microblogging attributes to latent feature representation which can be easily incorporated for product recommendation.

We propose and instantiate a feature-based matrix factorization approach by incorporating user and product features for cold-start product recommendation.

2 PROBLEM FORMULATION

Given an e-commerce website, let U denote a set of its users, P a set of products and R a jUjjPj purchase record matrix, each entry $r_{u;p}$ of which is a binary value indicating whether u has purchased product p. Each user u 2U is associated with a set of purchased products with the purchase timestamps. Furthermore, a small subset of users in U can be linked to their microblogging accounts (or other social network accounts), denoted as U^L. As such, each user u $2U^L$ is also associated with their respective microblogging attribute information. Let A denote the set of microblogging features, and each microblogging user has a jAj-dimensional microblogging feature vector a_u, in which each entry $a_{u;i}$ is the attribute value for the ith microblogging attribute feature.

With the notations introduced above, we define our recommendation problem as follows. We consider a cross-site cold-start scenario: a microblogging user u^0 2= U is new to the e-commerce website, who has no historical purchase records. It is easy to see u0 2= UL, too, since we have UL U. We aim to generate a personalised ranking of recommended products for u^0 based on her microblogging attributes a_{u0}.

Due to the heterogeneous nature between these two different data signals, information extracted from microblogging services cannot usually be used directly for product recommendation on e-commerce websites. Therefore, one major challenge is how to transform users' microblogging attribute information a_{u0} into another feature representation v_{u0}, which can be used more effectively for product recommendation. Here, we call a_{u0} the original or microblogging feature representation and v_{u0} the (heterogeneous) transformed feature representation, respectively.

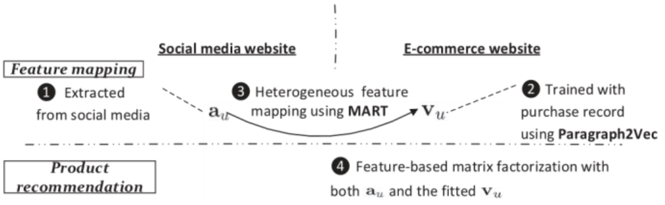
Social media website
E-commerce website
Feature mapping
❶ Extracted from social media
$\mathbf{a}_u$
❸ Heterogeneous feature mapping using MART
$\mathbf{v}_u$
❷ Trained with purchase record using Paragraph2Vec
Product recommendation
❹ Feature-based matrix factorization with both $\mathbf{a}_u$ and the fitted $\mathbf{v}_u$

Fig. 1. The workflow diagram for our presented solution.

Next, we will study how to extract microblogging features and transform them into a distributed feature representation before presenting a feature-based matrix factorization approach, which incorporates the learned distributed feature representations for product recommendation. The entire workflow of our solution is shown in Fig. 1 which consists of four major steps splitting into feature mapping and product recommendation, which will be discussed in Sections 3 and 4 respectively.

3 EXTRACTING AND REPRESENTING MICROBLOGGING ATTRIBUTES

Our solution to microblogging feature learning consists of three steps:

Prepare a list of potentially useful microblogging attributes and construct the microblogging feature vector a_u for each linked user u $2U^L$;

Generate distributed feature representations fv_ug$_{u2U}$ using the information from all the users U on the ecommerce website through deep learning;

Learn the mapping function, fða_uÞ! v_u, which transforms the microblogging attribute information a_u to the distributed feature representations v_u in the second step. It utilises the feature representation pairs fa_u;v_ug of all the linked users u $2U^L$ as training data.

3.1 Microblogging Feature Selection

In this section, we study how to extract rich user information from microblogs to construct a_u for a microblogging user. We consider three groups of attributes.

3.1.1 Demographic Attributes

A demographic profile (often shortened as "a demographic") of a user such as sex, age and education can be used by ecommerce companies to provide better personalised services. We extract users' demographic attributes from their public profiles on SINA WEIBO. Demographic attributes have been shown to be very important in marketing, especially in product adoption for consumers [4]. Following our previous study [5], we identify six major demographic attributes: gender, age, marital status, education, career and interests. To quantitatively measure these attributes, we have further discretized them into different bins following our previously proposed method described in [5].

3.1.2 Text Attributes

Recent studies have revealed that microblogs contain rich commercial intents of users [5], [6]. Also, users' microblogs often reflect their opinions and interests towards certain topics. As such, we expect a potential correlation between text attributes and users' purchase preferences. We perform Chinese word segmentation and stopword removal before extracting two types of text attributes below.

Topic distributions. Seroussi et al. ([7]) proposed to extract topics from user-generated text using the Latent Dirichlet Allocation (LDA) model for recommendation tasks. Follow the same idea, we first aggregate all the microblogs by a user into a document, and then run the standard LDA to obtain the topic distributions for each user. The benefits of topics distributions over keywords are two fold. First, the number of topics is usually set to 50 200 in practice, which largely reduces the number of dimensions to work with. Second, topic models generate condense and meaningful semantic units, which are easier to interpret and understand than keywords.

Word embeddings. Standard topic models assume individual words are exchangeable, which is essentially the same as the bag-of-words model assumption. Word representations or embeddings learned using neural language models help addressing the problem of traditional bag-ofword approaches which fail to capture words' contextual semantics [8], [9]. In word embeddings, each dimension represents a latent feature of the word and semantically similar words are close in the latent space. We employ the Skip-gram model implemented by the tool word2vec[4] to learn distributed representations of words. Finally, we average the word vectors of all the tokens in a user's published document as the user's embedding vector.

3.1.3 Network Attributes

In the online social media space, it is often observed that users connected with each other (e.g., through following links) are likely to share similar interests. As such, we can parse out latent user groups by the users' following patterns assuming that users in the same group share similar purchase preferences.

Latent group preference. Since it is infeasible to consider all users on WEIBO and only keeping the top users with the most followers would potentially miss interesting information, we propose to use topic models to learn latent groups of followings as in [10]. We treat a following user as a token and aggregate all the followings of a user as an individual document. In this way, we can extract latent user groups sharing similar interests (called "following topics"), and we represent each user as a preference distribution over these latent groups.

3.1.4 Temporal Attributes

Temporal activity patterns are also considered since they reflect the living habits and lifestyles of the microblogging users to some extent. As such, there might exist correlations between temporal activities patterns and users' purchase preferences.

TABLE 1
Categorization of the Microblogging Features

Categories	Features
Demographic	Gender (2), Age (6), Marital status (10),
Attributes	Education (7), Career (9), Interests (6)
Text	Topic distributions (50),
Attributes	Word embeddings (50)
Network Attributes	Latent group preference (50)
Temporal	Daily activity distribution (24),
Attributes	Weekly activity distribution (7)

The number of feature dimensions are shown in parentheses.

Temporal activity distributions. We consider two types of temporal activity distributions, namely daily activity distributions and weekly activity distributions. The daily activity distribution of a user is characterised by a distribution of 24 ratios, and the ith ratio indicates the average proportion of tweets published within the ith hour of a day by the user; similarly weekly activity distribution of a user is characterised by a distribution of seven ratios, and the ith ratio indicates the average proportion of tweets published within the ith day of a week by the user.

We summarize all types of features in Table 1.

3.2 Distributed Representation Learning with Recurrent Neutral Networks

In Section 3.1, we have discussed how to construct the microblogging feature vector a_u for a user u. However, it is not straightforward to establish connections between a_u and products. Intuitively, users and products should be represented in the same feature space so that a user is closer to the products that she has purchased compared to those she has not. Inspired by the recently proposed methods in learning word embeddings using recurrent neutral networks [8], [9], we propose to learn user embeddings or distributed representation of user v_u in a similar way.

3.2.1 Learning Product Embeddings

Before presenting how to learn user embeddings, we first discuss how to learn product embeddings. The neural network methods, word2vec, proposed in [8], [9] for word embedding learning can be used to model various types of sequential data. The core idea can be summarised as follows. Given a set of symbol sequences, a fixed-length vector representation for each symbol can be learned in a latent space by exploiting the context information among symbols, in which "similar" symbols will be mapped to nearby positions. If we treat each product ID as a word token, and convert the historical purchase records of a user into a timestamped sequence, we can then use the same methods to

International Journal of Research
Available at
https://edupediapublications.org/journals
Special Issue on Conference Papers
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 06
March 2018

learn product embeddings. Unlike matrix factorization, the order of historical purchases from a user can be naturally captured.

We consider two simple recurrent neutral architectures proposed in [11] to train product embeddings, namely, the Continuous Bag-Of-Words model (CBOW) and the Skipgram model. The major difference between these two architectures lies in the direction of prediction: CBOW predicts the current product using the surrounding context, i.e., Prðp$_t$jcontextÞ, while Skip-gram predicts the context with the current product, i.e., Prðcontextjp$_t$Þ. In our experiments, the context is defined as a window of size 4 surrounding a target product p_t which contains two products purchased before and two after p_t. More formally, each product p_t is modeled as a unique latent embedding vector v_{pt}, and the associated context vector is obtained to average the vectors of the context information as $v_{context}$. For CBOW, the conditional prediction probability is characterized by a softmax function as follows

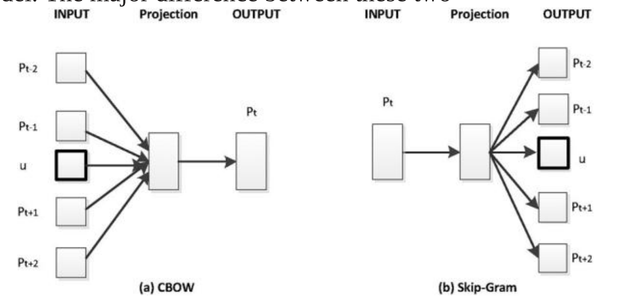

Fig. 2. Two architectures to learn both product and user embeddings. Here u denote a user ID. The major difference between para2vec and word2vec lies in the incorporation of user ID as additional context.

ð tj   Þ¼ expðv$^{>}_{p_t}$ v$_{context}$ Þ Þ

Pr p context:

P$_p$ expðv$_{>p}$ v$_{context}$

To optimize for computing exponential sum probabilities, hierarchical softmax and negative sampling techniques are commonly used to speed up the training process. At each training iteration, we sample a target product together with their context window, and then update the parameters with Stochastic Gradient Descent (SGD) using the gradients derived by backpropogation. Learning for Skip-gram is done in a similar way, which is omitted here.

3.2.2 Learning User Embeddings

Given product embeddings, if we can learn user embeddings in a similar way, then we can explore the correlated representations of a user and products for product recommendation. We borrow the idea from the recently proposed Paragraph Vectorðpara2vecÞ method [9], which learns feature representations from variable-length pieces of texts, including sentences, paragraphs, and documents. We implement a simplified version of para2vec at the sentence level as follows. The purchase history of a user can be considered as a "sentence" consisting of a sequence of product IDs as word tokens. A user ID is placed at the beginning of each sentence, and both user IDs and product IDs are treated as word tokens in a vocabulary in the learning process. During training, for each sentence, the sliding context window will always include the first word (i.e., user ID) in the sentence. In this way, a user ID is essentially always associated with a set of her purchase records (a context window of four products at a time). We can then use the same learning procedure in word2vector for the estimation of Prðcontextjp$_t$Þ and Prðp$_t$jcontextÞ. We present an illustrative example of these two architectures in Fig. 2. After learning, we separate user embeddings from product embeddings and use v_u and v_p to denote the learnt K-dimensional embedding for user u and product p respectively.

The rationales of applying para2vec to model purchase data can be explained below. First, the user embedding representation for each user ID reflects the users' personalized purchase preference; Second, the surrounding context, i.e., product purchases, is used to capture the shared purchase patterns among users. Compared to the traditional matrix factorization , the (window-based) sequential context is additionally modeled in addition to user preference, which is expected to potentially yield better recommendation results.

3.3 Heterogenous Representation Mapping Using Gradient Boosting Regression Trees

We have presented how to construct a microblogging feature vector a_u from a microblogging site and learn a distributed representation v_u from an e-commerce website respectively. In the cross-site cold-start product recommendation problem we considered in this paper (i.e., make a product recommendation to a user u who has never purchased any products from an e-commerce website), we can only obtain the microblogging feature vector a_u for user u. The key idea is to use a small number of linked users across sites as a bridge to learn a function which maps the original feature representation a_u to the distributed representation v_u. Specifically, we can construct a training set consisting of feature vector pairs, fa$_u$;v$_u$g$_{u2UL}$ and cast the feature mapping problem as a supervised regression task: the input is a microblogging feature vector a_u and the output is a distributed feature vector v_u.

Assume that v_u contains K dimensions, we need to learn a set of K functions ff^{ðiÞ}g$^{K}_{i¼1}$, and the ith function f^{ðiÞ} takes the original feature vector of a user u as the input and returns the corresponding ith transformed feature value $v_{u;i}$, i.e., $v_{u;i}$ ¼ f^{ðiÞ}ða^{ðuÞ}Þ. We extend the Multiple Additive Regression Tree (MART) [13] method to learn feature mapping functions since it is powerful to capture higher-order transformation relationship between input and output.

4 Applying The Transformed Features to Cold-Start Product Recommendation

Once the MART learners are built for feature mapping, the original microblogging feature vectors a_u are mapped onto the user embedding v_u. In this section, we study how to incorporate fa$_u$;v$_u$g into the feature-based matrix factorization technique. In specific, we develop our recommendation method based on the recently proposed SVDFeature. Our idea can also be applied to other feature-based recommendation algorithms, such as Factorization Machines (FMs)

4.1.1 Feature Coding with the Side Information

We discuss how to incorporate the user and product information into the SVDFeature framework.

Coding users and products. For users, we reserve the first jUj dimensions in the user input vector. Each user u is coded as a vector of jUj-dimensional vector consists of a "1" in the uth dimension and "0" in other dimensions; Similarly, we can reserve the first jPj dimensions in the product input vector to code the products. Formally, we have

a$_{ðjuÞ}$ ¼10;; jj ¼6¼ uu:;   b$_{ðjpÞ}$ ¼01;; jj ¼6¼ p:p;

Coding microblogging attributes. Given a user u, we use the dimensions from ðjUjþ 1Þth to ðjUjþjAjÞth to code her microblogging attribute vector$_u$ a$_u$. For i ¼ 1 to jAj, we have a$_{ð}$$_{jUjþ}^{Þ}_{i}$ ¼ a$_{u;i}$. Here we follow [20] to directly incorporate microblogging attributes. In practice, a subset of features A^0 can be identified with expertise knowledge instead of using the full set of features in A.

Coding user embeddings. Given a user u, we use the dimensions from ðjUjþjAjþ 1Þth to ðjUjþjAjþ KÞth to code her distributed feature vector (user embedding)$_u$ v$_u$. For k ¼ 1 to

K, we have a$_{ðjUjþÞ\ k}$ ¼ v$_{u;k}$.

5 EXPERIMENTS

We present experimental setup first before discussing our results.

5.1 Experimental Setup

Our task requires data from both an e-commerce website and an online social networking site.

E-commerce data. We used a large e-commerce dataset shared by [6], which contains 138.9 million transaction records from 12 million users on 0.2 million products. Each transaction record consists of a user ID, a product ID and the purchase timestamp. We first group transaction records by user IDs and then obtain a list of purchased products for each user.

Microblogging data. We used our previous data [5] collected from the largest Chinese microblogging site SINA WEIBO, in which we have retrieved a total of 1.7 billion tweets from five million active users within a half-year time span from January 2013 to June 2013.

User linkage. We have found that WEIBO users sometimes shared their purchase record on their microblogs via a system-generated short URL, which links to the corresponding product entry on JINGDONG. By following the URL link, we can obtain the JINGDONG account of the WEIBO user.[6] We

TABLE 3
Performance Comparisons of MAE Results for

——		MART$_{old}$		MART$_{sample}$		MART$_{both}$
	Fitting User Embeddings On D$_{dense}$					
#train #test	CART					
1/1	0.557	0.515	0.515	0.515 1/4	0.557	0.522
	0.521	0.521 1/9	0.564	0.589	0.558	0.529

Smaller is better.

identified 23,917 linked users out of five million active users by scanning tweets in this way. We first filter out 3,279 users with too little information on their WEIBO public profiles. Next, we further divide users into two groups. The first group contains users with more than five product purchases, denote as D$_{dense}$. The second group contains the remaining users, denoted as D$_{sparse}$. The statistics of these linked users are summarized in Table 2. For privacy consideration, all the WEIBO IDs and JINGDONG IDs of all linked users are replaced by anonymized unique IDs, and all their textual information and purchase information is encoded with numeric symbols.

5.2 Evaluation on User Embeddings Fitting

Given a linked user u 2U^L, we have the microblogging feature vector a$_u$ extracted from WEIBO and the user embedding v$_u$ learnt based on her JINGDONG purchase record. We use a regression-based approach to fit v$_u$ with a$_u$ for heterogeneous feature mapping, and the fitted vector is denoted as v^$_u$. To examine the effectiveness of the regression performance, the Mean Absolute Error (MAE) is used as the evaluation metric

$$\underline{\quad} ¼ 1\ XP\underline{_{Kk¼1}jv_{u;k}\ \ v\hat{}_{u;k}j;\quad} \qquad (8)$$

$$\mathrm{MAE}$$

$$jTj \qquad K\ ^{u}{}_{2T}$$

where jTj is the number of test users. We consider three different comparison methods: (1) CART [14]; (2) MART$_{old}$, which is the original implementation as in [13]; (3) MART$_{sample}$, which is our modified implementation with feature sampling; (4) MART$_{both}$, which is our modified implementation with feature sampling and fitting refinement.

For user embedding fitting, we use D$_{dense}$ for evaluation, since the users in D$_{dense}$ have a considerable number of purchases for learning the ground truth user embeddings using our modified

[6] Note that when a user shares a purchase record on her microblog, she will be notified automatically by SINA WEIBO that her JINGDONG account would be exposed to the public.

para2vec method, which are more reliable for evaluation. The dataset D_{dense} is split by users into training set and test set with three different $\frac{\#\ train}{\#\ test}$ ratios, namely 1:1, 1:4 and 1:9. We use a similar evaluation method as N-fold cross validation. Given the $\frac{\#\ train}{\#\ test}$ ratio of 1 : N, each fold will be treated as the training data exactly once and the rest N 1 folds are treated as the test data, the process will be repeated N times and the final results are averaged over N such runs. The number of boosting iterations for all MART variants and the values of m_1 and m_2 for MART_{both} are optimized by N-fold cross validation.

In Table 3, we can see that when the training data is relatively large (ratio 1:1), all the MART variants give similar results and they perform consistently better than the simple CART. Interestingly, when the size of training data becomes smaller, MART_{sample} and MART_{both} outperforms MART_{old}. In specific, the performance gain achieved by MART_{both} over the other two MART variants is more significant with smaller set of training data. These results show that our modifications of feature sampling and fitting refinement are very effective.

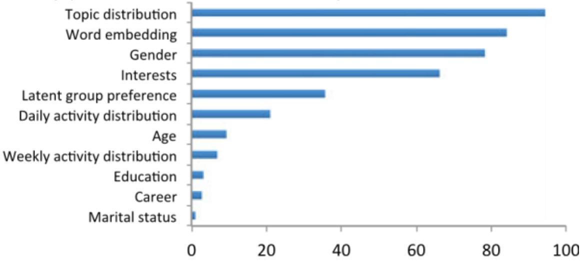

Fig. 3. Relative attribute importance ranking (corresponding to the features in Table 1).

Relative attribute importance. Tree-based methods offer additional feasibility to learn relative importance of each attribute. Inspired by the method we calculate a statistic of the relative importance of each attribute for MART based on the mtrainingl data. Recall that in MART, each feature corresponds to an attribute value. First, we traverse through all the regression trees, and calculate for each feature its contribution to the cost function by adding up the contributions of all the nodes that are split by this feature. Here we define feature contribution to be the reduction of the squared error in the loss function. For each attribute, we can sum up the contributions of all of its possible attribute values as its overall contribution.

The results are shown in Fig. 3. We have the following observations: 1) The text attributes occupy the top two rank positions;[7] 2) Within the demographic category, Gender and Interests are more important than the others. 3) The social based attributes are ranked relatively lower compared to the other two categories. It seems that demographic attributes are less important than text attributes in our dataset. One possible reason is that many demographic attribute values are missing in users' public profiles on WEIBO.[8] Nevertheless, the ranking of relative importance of attributes does not entirely depend on their completeness proportion. For example, Interests is more important than Latent group preference even though the later has a larger completeness proportion. Another possible reason is that the feature dimension for text attributes is much larger than that of demographic attributes, e.g., Topic Distribution has fifty feature dimensions while Gender only has two feature dimensions.

We can also evaluate the importance of each attribute by conducting experiments on the traditional product recommendation task. We use the standard MF approach as a baseline and add attributes one at a time using the SVDFeature framework discussed in Section 4.1, then check the performance improvement yielded by the added attribute. The attribute ranking obtained in this way is similar to the ranking in Fig. 3, but the gap between text attributes and demographic attributes becomes smaller.

5.3 Evaluation on Cold-Start Product Recommendation

For cold-start product recommendation, we aim to recommend products to microblog users without the knowledge of their historical purchase records.

5.3.1 Construction of the Evaluation Set

The evaluation set splits users into training set and test set. For the training set, we sample negative products with a ratio of 1:1 for each user, i.e., we have the same number of negative and positive products. For the test set, we randomly sample negative products with a ratio of 1:50 for each user, i.e., each positive product would involve 50 negative products. All negative products are sampled from the same product category as the corresponding positive one. For example, for "iPhone 6", we can sample "Samsung Galaxy S5" from the "Mobile Phones" category as a negative product. Given a user, we can generate a list of candidate products consisting of both positive and negative products. On average, a user has about 52 positive products and 2,600 negative products in our experimental dataset, which is indeed a challenging task. Similar to the evaluation scenario in Information Retrieval, we would like to examine the performance that a system ranks positive products over negative products.

5.3.2 Methods to Compare

We consider the following methods for performance comparison:

[7] Although both topic distributions and word embeddings are usedto capture the semantic characteristics of user-generated text, they have different focuses. Topic distributions are more suitable to extract topical themes from text based on word co-occurrence patterns (essentially taking the whole document as the context window) while word embeddings are more suitable to capture the semantics between words from local context windows, usually comprising three words before and after the target word. Hence, we keep both types of text features in our approach. It is worth noting that our method is a tree-based approach, which can effectively handle information redundancy, i.e., if a feature contains redundant information given the tree that is being constructed, it will be pushed to a lower rank during attribute selection.

[8] In our dataset, the completeness proportion of demographicattributes are as follows: Gender (100 percent), Interests (65.7 percent), Age (36.7 percent), Education (26.3 percent), Career (12.9 percent) and Marital status (4.6 percent); while for text and network attributes, the proportion of completeness is about 99.1 percent, i.e., most users have published tweets and followed some other users.

Popularity (Pop): products are ranked by their historical sale volumes.

Popularity with Semantic Similarity (Pop++): the ranking score is a combination of two scores: (1) the popularity score S_1; (2) the cosine similarity S_2 between product description and user text information, including profile, tweets and tags. The two scores are combined by logð1 þ S_1Þ logð1 þ S_2Þ.

Embedding Similarities (ES): Similarity scores $v\hat{}^{>}_{u}\, v_p$ between a user embedding $v\hat{}_u$ and a list of product embeddings v_p are used to rank products.

MF with user attributes (MFUA): User attributes (including user profile and topic distributions) are incorporated into the basic matrix factorisation algorithm for product rating prediction [7]. For fairness, we also use the pairwise loss function to train the model.

FM without User Interactions (FMUI): Rendle applied the Factorization Machines for “follow” recommendation in KDDCup 2012. It has been found that similar performance was obtained with or without the interactions of user features. FM without user feature interactions is equivalent to SVDFeature. We reimplement this method in the SVDFeature framework with our extracted microblogging features.

Cold_E: Our proposed approach which uses the fitted user embedding features and product embedding features (Eq. (6)).

$\mathrm{Cold}_{DþE}$: Our proposed approach which uses the microblogging features, the product embedding features and the fitted user embedding features (Eq. (7)). Especially, we only use demographic attributes here, since they have been shown important to product recommendation [5]

$\mathrm{Cold}_{þþ}$: Since the user and product embeddings can be learned for all the users and products respectively in the e-commerce website, we can train Cold_E with all the users in U, not limited to the linked users U^L.

This variant is called $\mathrm{Cold}_{enhanced}$.

We set the regularization coefficient to a 0:004, the iteration number to 50 and the factor number to 32 for all the methods. We use the CBOW architecture to learn the embedding vectors based on the purchase records from all the non-linked users and the partial purchase records from linked users in our training set. The number of dimensions of embedding vectors is set to 50. The user embedding fea-

tures in the test sets for different $^{\#}\underline{\hphantom{xxxxxx}}^{\text{training}}{}_{\#\text{test}}$ settings are set to the values fitted using MART_{both}. For $\mathrm{Cold}_{enhanced}$, we add additional 10,000 randomly selected non-linked users from U into the training set.

5.3.3 Evaluation Metrics for Product Recommendation

Five widely used metrics are used for the evaluation of product recommendation results, including Precision@k, Recall@k, the Mean Average Precision (MAP), the Mean Reciprocal Rank (MRR) and the area under the ROC Curve (AUC).

5.4 Revisiting the Effectiveness of the Distributed Representations of Users and Products

In the previous section, we have shown that the learnt product and users embeddings are effective to improve the recommendation performance. In this section, we give more insights into the effectiveness of the distributed representations.

5.4.1 Insights into Product Embeddings

First, we take the learnt product embeddings to conduct a quantitative similarity analysis in order to find out whether the learned product embeddings can discriminate products from different categories or brands. We compute the average similarity score between product pairs from (1) different categories and brands (DCDB); (2) same category but different brands (SCDB); and (3) same category and same brand (SCSB). As it is infeasible to calculate the similarity scores for all possible product pairs in JingDong, we sample 10 million product pairs randomly for each type of product pairs for computation. The results are as follows: sim_{DCDB} ¼ 0:0217, sim_{SCDB} ¼ 0:2719 and sim_{SCSB} ¼ 0:4406. The average similarity score of $\mathrm{sim}_{SCDB} > \mathrm{sim}_{DCDB}$ indicates the product embeddings learned are indeed very different for products under different categories; while $\mathrm{sim}_{SCSB} > \mathrm{sim}_{SCDB}$ indicates the product embeddings have a good discriminative power for brands.[12]

5.4.2 Insights into User Embeddings

We take the learnt user embeddings to conduct a quantitative similarity analysis in order to find out whether the learned user embeddings can identify users with similar purchase history.

Given a user u, we build two groups of users, denoted by G A B A u G_u G_u and . contains the top K most similar users (a.k.a. K nearest neighbours) of user u, which are identified by the

Jacarrd coefficient in terms of purchase history; $G^B{}_u$ contains K randomly selected users. We would like to examine whether the user embedding vectors can discriminate a user in $G^A{}_u$ from another one in $G^B{}_u$.

Given user u together with $G^A{}_u$ and $G^B{}_u$, we can derive two similarity values $\mathrm{sim}_{ð_{A}\,uÞ}$ and $\mathrm{sim}_{ð_{B}\,uÞ}$, which are the average similarities with the users in $G^A{}_u$ and the users in $G^B{}_u$ respectively for user u. We use the cosine function to compute the similarity between two user embedding vectors. K is set to 30 in our experiments. In this way, we can obtain two arrays of similarity values fsim$_{ðAuÞ}$g$_{u2U}$ and fsim$_{ðBuÞ}$g$_{u2U}$. By constructing the paired t-test, the results have shown that the values in fsim$_{ðAuÞ}$g$_{u2U}$ are significantly larger than those in fsim$_{ðBuÞ}$g$_{u2U}$ at the level of 0.001. The average similarities for fsim$_{ðAuÞ}$g$_{u2U}$ and fsim$_{ðBuÞ}$g$_{u2U}$ are 0.090 and 0.031 respectively.

6 Related Work

Our work is mainly related to three lines of research:

Recommender systems. In recent years, the matrix fatortization approach has received much research interests. With the increasing volume of web data, many studies focus on incorporating auxiliary information [1], into the matrix factorization approach. Two typical frameworks of such studies are the SVDFeature and Factorization Machine.

There has also been a large body of research work focusing specifically on the cold-start recommendation problem. Seroussi et al. [7] proposed to make use of the information from users' public profiles and topics extracted from usergenerated content into a matrix factorization model for new users' rating prediction. Zhang et al. propose a semisupervised ensemble learning algorithm. Schein proposed a method by combining content and collaborative data under a single probabilistic framework. Lin et al. [10] addressed the cold-start problem for App recommendation by using the social information from Twitter. Trevisiol et al. Zhou et al. experimented with eliciting new user preferences using decision trees by querying users' responses progressively through an initial interview process. Moshfeghi et al. proposed a method for combining content features such as semantic and emotion information with ratings information for the recommendation task. Bao and Chang presented an influence-based diffusion model considering user influence in addition to relevance for matching ads. Liu et al. identified representative users whose linear combinations of tastes are able to approximate other users.

Cross-domain recommendation. One of the key techniques for cross-domain recommendation is Transfer Learning and the idea is to learn transferable knowledge from the source domain, and further apply it in a target domain. Singh and Gordon proposed collective matrix factorization to estimate the relations of multiple entities by factorizing several matrices simultaneously while sharing parameters in the latent space. Li attempted to transfer user-item rating patterns from an auxiliary matrix in another domain to the target domain through Codebooks. Hu and Zhao extended transfer learning to triadic factorization and active learning for cross-domain recommendation, respectively.

Social network mining. We follow the early commercial mining studies on social networking websites. Hollerit et al. presented the first work on commercial intent detection in Twitter. Zhao et al. [5] first proposed to route products from e-commerce companies to microblogging users. Our work is also related to studies on automatic user profiling and cross-site linkage inference .

Our work is built upon these studies, especially in the areas of cross-domain and cold-start recommendation. Though sharing some similarities, we are dealing with a very specific task of highly practical value, cold-start product recommendation to microblogging users. To the best of our knowledge, it has not been studied on a large data set before. The most relevant studies are from by connecting users across eBay and Facebook. However, they only focus on brand- or category-level purchase preference based on a trained classifier, which cannot be directly applied to our cross-site coldstart product recommendation task. In addition, their features only include gender, age and Facebook likes, as opposed to a wide range of features explored in our approach. Lastly, they do not consider how to transfer heterogeneous information from social media websites into a form that is ready for use on the e-commerce side, which is the key to address the crosssite cold-start recommendation problem.

7 Conclusions

In this paper, we have studied a novel problem, cross-site cold-start product recommendation, i.e., recommending products from e-commerce websites to microblogging users without historical purchase records. Our main idea is that on the e-commerce websites, users and products can be represented in the same latent feature space through feature learning with the recurrent neural networks. Using a set of linked users across both e-commerce websites and social networking sites as a bridge, we can learn feature mapping functions using a modified gradient boosting trees method, which maps users' attributes extracted from social networking sites onto feature representations learned from e-commerce websites. The mapped user features can be effectively incorporated into a feature-based matrix factorisation approach for cold-start product recommendation. We have constructed a large dataset from Weibo and JingDong. The results show that our proposed framework is indeed effective in addressing the cross-site cold-start product recommendation problem. We believe that our study will have profound impact on both research and industry communities.

Currently, only a simple neutral network architecture has been employed for user and product embeddings learning. In the future, more advanced deep learning models such as Convolutional Neural Networks[13] can be explored for feature learning. We will also consider improving the current feature mapping method through ideas in transferring learning.

References

[1] J. Wang and Y. Zhang, "Opportunity model for E-commerce recommendation: Right product; right time," in Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2013, pp. 303–312.

[2] M. Giering, "Retail sales prediction and item recommendations using customer demographics at store level," SIGKDD Explor. Newsl., vol. 10, no. 2, pp. 84-89, Dec. 2008.

[3] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," IEEE Internet Comput., vol. 7, no. 1, pp. 76–80, Jan./Feb. 2003.

[4] V. A. Zeithaml, "The new demographics and market fragmentation," J. Marketing, vol. 49, pp. 64–75, 1985.

[5] W. X. Zhao, Y. Guo, Y. He, H. Jiang, Y. Wu, and X. Li, "We know what you want to buy: A demographic-based system for product recommendation on microblogs," in Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2014, pp. 1935–1944.

[6] J. Wang, W. X. Zhao, Y. He, and X. Li, "Leveraging product adopter information from online reviews for product recommendation," in Proc. 9th Int. AAAI Conf. Web Social Media, 2015, pp. 464–472.

[7] Y. Seroussi, F. Bohnert, and I. Zukerman, "Personalised rating prediction for new users using latent factor models," in Proc. 22nd ACM Conf. Hypertext Hypermedia, 2011, pp. 47–56.

[8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their

compositionality,” in Proc. Adv. Neural Inf. Process. Syst., 2013, pp. 3111–3119.

[9] Q. V. Le and T. Mikolov, “Distributed representations of sentences and documents,” CoRR, vol. abs/1405.4053, 2014.

[10] J. Lin, K. Sugiyama, M. Kan, and T. Chua, “Addressing cold-start in app recommendation: Latent user models constructed from twitter followers,” in Proc. 36th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2013, pp. 283–292.