Connecting Social Media to E-Commerce: Cold-Start Product Recommendation using Microblogging Information

T. Sampath Reddy\textsuperscript{1} \hspace{1cm} V. Divya vani\textsuperscript{2}

\textsuperscript{1}Assistant Professor in Department of Computer Science & Engineering \hspace{1cm} \textsuperscript{2}M.Tech

123 Nalla Narasimha Reddy Education Society's Group of Institutions

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 SI N A WE I B O and the largest Chinese B2C e-commerce website JI N G DO N G have shown the effectiveness of our proposed framework.

Keywords—e-commerce, product recommender, product demographic, microblogs, recurrent neural network

I. INTRODUCTION

In recent years, the boundaries between e-commerce and social networking have become increasingly blurred. E-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 web sites 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 WEIBO where ALIBABA 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-commerce 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 cold-start product recommendation. We built our dataset from the largest Chinese microblogging service SINA WEIBO2 and the largest Chinese B2C e-commerce website JINGDONG3, 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 network works 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.

II. PROBLEM FORMULATION

Given an e-commerce website, let U denote a set of its users, P a set of products and R a \([U \times P]\) purchase record matrix, each entry \(r_{u,p}\) of which is a binary value indicating whether u has purchased product p. Each user \(u \in U\) 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 \(u \in U_L\) is also associated with their respective microblogging attribute information. Let A denote the set of microblogging features, and each microblogging user has a \(|A|\)-dimensional microblogging feature vector \(a_u\), in which each entry \(a_{ui}\) is the attribute value for the i-th 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 \(u' \in U\) is new to the e-commerce website, who has no historical purchase records. It is easy to see \(u' \in U_L\), too, since we have \(U_L \subset U\). We aim to generate a personalised ranking of recommended products for \(u'\) based on her microblogging attributes \(a_{u'}\).

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 \(a_u\) into another feature representation \(v_u\), which can be used more effectively for product recommendation. Here, we call \(a_u\) the orignal or microblogging feature representation and \(v_u\) the (heterogeneous) transformed feature representation, respectively. 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 Figure 1, which consists of four major steps splitting into feature mapping and product recommendation, which will be discussed in Section 3 and 4 respectively.

III. 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 \(u \in U_L\); • Generate distributed feature representations \(\{v_u\}_{u \in U}\) using the information from all the users U on the e-commerce website through deep learning. Learn the mapping function, \(f(a_u) \rightarrow v_u\), 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 \(\{a_u, v_u\}\) of all the linked users \(u \in U_L\) 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.

Demographic Attributes

A demographic profile (often shortened as— a demographic profile) of a user such as sex, age and education can be used by e-commerce 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].

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 topic 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-of-words 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.

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.

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.

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 i-th ratio indicates the average proportion of tweets published within the i-th hour of a day by the user; similarly weekly activity distribution of a user is characterised by a distribution of seven ratios, and the i-th ratio indicates the average proportion of tweets published within the i-th day of a week by the user. We summarise all types of features in Table 1.

3.2 Distributed Representation Learning With Recurrent Neural Networks

In Section 3.1, we have discussed how to construct the microblogging feature vector au for a user u. However, it is not straightforward to establish connections between au 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 neural networks [8], [9], we propose to learn user embeddings or distributed representation of user vu in a similar way.

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 learn product embeddings. Unlike matrix factorization, the order of historical purchases from a user can be naturally captured.
We consider two simple recurrent neural architectures proposed in [11] to train product embeddings, namely, the Continuous Bag-Of-Words model (CBOW) and the Skip-gram 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., \( P(\text{pt} | \text{context}) \), while Skip-gram predicts the context with the current product, i.e., \( P(\text{context}|\text{pt}) \). In our experiments, the context is defined as a window of size 4 surrounding a target product \( \text{pt} \) which contains two products purchased before and two after \( \text{pt} \). More formally, each product \( \text{pt} \) is modeled as a unique latent embedding vector \( v_{\text{pt}} \), and the associated context vector is obtained to average the vectors of the context information as \( v_{\text{context}} \). For CBOW, the conditional prediction probability is characterized by a softmax function as follows:

\[
P(\text{pt} | \text{context}) = \frac{\exp(v_{\text{context}}^T \cdot v_{\text{pt}})}{\Sigma_p \exp(v_{\text{context}}^T \cdot v_p)}
\]

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 backpropagation. Learning for Skip-gram is done in a similar way, which is omitted here.

**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 Vector (para2vec) 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 4 products at a time). We can then use the same learning procedure in word2vector for the estimation of \( P(\text{context}|\text{pt}) \) and \( P(\text{pt} | \text{context}) \).

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 [12], the (window-based) sequential context is additionally modeled in addition to user preference, which is expected to potentially yield better recommendation results.

**IV. 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 \( \{a_u, v_u\} \) into the feature-based matrix factorization technique. In specific, we develop our recommendation method based on the recently proposed SVDFeature [18]. Our idea can also be applied to other feature-based recommendation algorithms, such as Factorization Machines [19].

**4.1 The General SVDFeature Framework for Product Recommendation**

SVDFeature [18] is built based on the traditional matrix factorization approach, and it considers factorization in three aspects, namely global features (also called as dyadic features), user features and item features. It can be formulated for the task of product recommendation as follows:

\[
\hat{y}_{u,p}(c_u, \beta_p, \gamma_{u,p}) = \sum_j \sum_j \alpha_j \beta_j \gamma_j (\sum_j \alpha_j \beta_j \gamma_j)
\]

where \( \alpha(u) \in RN_u \), \( \beta(p) \in RN_p \) and \( \gamma(u,p) \in RN_{\gamma} \) are the input vectors consisting of the features of user \( u \), the features of product \( p \) and the global features for the pair \( (u, p) \) with the lengths of \( N_u \), \( N_p \) and \( N_{\gamma} \) respectively. Here, \( b(G) \), \( b(U) \) and \( b(P) \) are the global, latent vectors \( x_j \) and \( y_j \) capture the \( j \)-th user feature and the \( j \)-th product feature respectively.
Let \( \{x_j\} \) and \( \{y_j\} \) denote the set of all user features and product features respectively. Note that \( \{x_j\} \) are shared by all the users, \( \{y_j\} \) are shared by all the products, and the global features and bias values do not have any corresponding latent vectors. In summary, a user-product pair corresponds to a feature vector concatenated by global features, user features and product features. The response value to be fitted indicates whether the user has purchased the product or not.

**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 \(|U|\) dimensions in the user input vector. Each user \( u \) is coded as a vector of \(|U|\)-dimensional vector consists of a “1” in the \( u \)th dimension and “0” in other dimensions. Similarly, we can reserve the first \(|P|\) dimensions in the product input vector to code the products. Formally, we have

\[
\alpha^{(u)}(j) = \begin{cases} 
1, & j = u; \\
0, & j = p.
\end{cases}
\]

**Coding microblogging attributes:** Given a user \( u \), we use the dimensions from \((|U| + 1)\)-th to \((|U| + |A|)\)-th to code her microblogging attribute vector \( a_u \). For \( i = 1 \) to \(|A|\), we have \( \alpha(u) = a_u,i \). Here we follow [20] to directly incorporate microblogging attributes. In practice, a subset of features \( A' \) 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 \((|U| + |A| + 1)\)-th to \((|U| + |A| + K)\)-th to code her distributed feature vector (user embedding) \( v_u \). For \( k = 1 \) to \( K \), we have \( \alpha(u)|U| + k = v_u,k \)

**REFERENCES**


