International Journal of Innovative Technologies
WWW.IJITECH.ORG
ISSN 2321-8665
Volume.06, Issue.01,
January-June, 2018,
Pages:0025-0032

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

J. SINDHUJA[1], RAVEENDRA REDDY ENUMULA[2]
[1]PG Scholar, Dept of CSE, St.Mary's Group of Institutions, Guntur, AP, India.
[2]Assistant Professor, Dept of CSE, St.Mary's Group of Institutions, Guntur, AP, 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 "coldstart" 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.

Keywords: E-Commerce, Product Recommender, Product Demographic, Microblogs, Recurrent Neural Networks.

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 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 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 ecommerce 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.

II. PROBLEM FORMULATION

Given an e-commerce website, let U denote a set of its users, P a set of products and **R** a $|\mathcal{U}| \times |\mathcal{P}|$ purchase record matrix, each entry ru;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 $|\mathcal{A}|$-dimensional microblogging feature vector **a**u, in which each entry au;i 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′ $\notin$U is new to the e-commerce website, who has no historical purchase records. It is easy to see u′ $\notin$ 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 as shown in Fig.1. Here, we call **a**u′ the original or microblogging feature representation and **v**u′ the (heterogeneous) transformed feature representation, respectively.

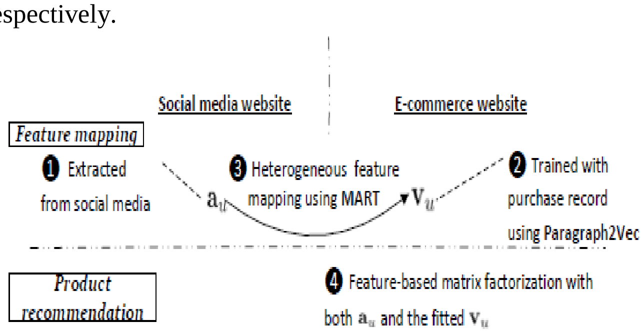

Fig. 1. The workflow diagram for our presented solution.

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 $\{\mathbf{v}_u\}_{u \in \mathcal{U}}$ using the information from all the users U on the e-commerce website through deep learning;
- Learn the mapping function, $f(\mathbf{a}_u) \rightarrow \mathbf{v}_u$, 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 $\{\mathbf{a}_u, \mathbf{v}_u\}$ of all the linked users u $\in$ U^L as training data.

A. 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") 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 stop word 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 twofold. 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-word 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 Skipgram model implemented by the tool word2vec4 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 characterized 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 characterized 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 summarize all types of features in Table1.

B. Distributed Representation Learning With Recurrent Neutral Networks

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.

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 neutral architectures proposed in[11] to train product embeddings, namely, the Continuous Bag-Of-Words model (CBOW) and the Skip-gram model.

TABLE I: Categorisation Of The Microblogging Features. The Number Of Feature Dimensions Are Shown In Parentheses

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

The major difference between these two architectures lies in the direction of prediction: CBOW predicts the current product using the surrounding context, i.e., Pr(pt|context), while Skip-gram predicts the context with the current product, i.e., Pr(context|pt). In our experiments, the context is defined as a window of size 4 surrounding a target product pt which contains two products purchased before and two after pt. More formally, each product pt 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

$$Pr(p_t|\text{context}) = \frac{\exp(\mathbf{v}_{p_t}^{\top} \cdot \mathbf{v}_{context})}{\sum_p \exp(\mathbf{v}_p^{\top} \cdot \mathbf{v}_{context})} \quad \textbf{(1)}$$

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.

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 Pr(context|pt) and Pr(pt|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.

C. 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 as shown in Fig.2.

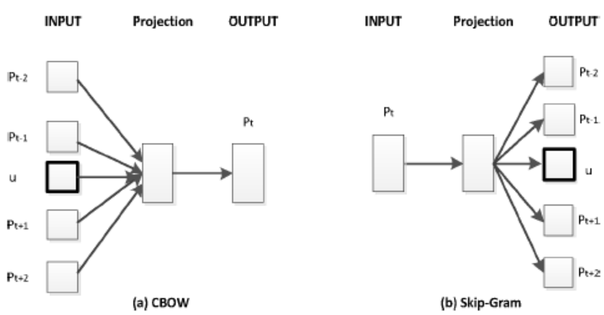

Fig. 2. Two architectures to learn both product and user embeddings.

Here u denote a user ID. The major difference between para2vec and word2vec lies in the incorporation of user ID as additional context. u who has never purchased any products from an ecommerce 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, $\{\mathbf{a}_u, \mathbf{v}_u\}_{u\in\mathcal{U}^L}$ 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 $\{f^{(i)}\}_{i=1}^{K}$, and the i-th function f(i) takes the original feature vector of a user u as the input and returns the corresponding i-th transformed feature value vu;i, i.e., $v_{u,i} = f^{(i)}(\mathbf{a}^{(u)})$. 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.

A brief Introduction of MART: Gradient boosting algorithms aim to produce an ensemble of weak models that together form a strong model in a stage-wise process. Typically, a weak model is a J-terminal node Classification And Regression Tree (CART) [14] and the resulting gradient boosting algorithm is called Multiple Additive Regression Tree (MART) [13]. An input feature vector **x** $\in$ $\mathbf{R}^d$ is mapped to a score F(**x**) $\in$ **R**. The final model is built in a stage-wise process by performing gradient descent in the function space. At the mth boosting,

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \eta\rho_m h_m(\mathbf{x}; \mathbf{a}) \qquad \textbf{(2)}$$

where each $h_m(\cdot)$is a function parameterised by $\mathbf{a}_m$, $\rho_m \in \mathbf{R}$is the weight associated with the mth function, and $0 < \eta \leq 1$ is the learning rate. The learning procedure of gradient boosting consists of two alternative steps in the m-th iteration: first fit a new component function hm by using the steepest descent method, and then minimize the loss function to derive the ensemble weight _m for the learnt learner. At each iteration, we use the regularized squared error function to learn a new CART component: we first derive a set of disjoint regions {Rj} which covers the space of all the joint values of the input feature vector, and then set the region fitting coefficient for Rj to the average of "pseudo responses" of the instances falling in Rj .

Completeness-Based Feature Sampling: An issue about the gradient boosting algorithm is that it tends to overfit the training data. It has been previously shown that the incorporation of randomized feature sampling improves the tree based ensemble methods in Random Forest [15]. Inspired by the idea, we propose to use an attribute-level importance sampling method where each attribute is assigned with an importance score and at each node split in building the MART trees, we only sample a fraction of attributes (empirically set to 2 3) based on each attribute's importance score instead of enumerating all the attributes. Once an attribute is sampled, its corresponding attribute value features will be selected subsequently. The importance score of each attribute is set to the proportion of the attribute values that can be extracted from the users' public profiles on SINA WEIBO. Another benefit of completeness-based sampling is that attributes with a larger proportion of missing values will be more likely to be pushed to the leaf nodes, which alleviates the missing value problem in regression trees.

Fitting Refinement: Here we propose two methods to refine the fitted values. First, the fitting quality relies on the number of available linked users since insufficient trainingdata would hurt the performance of the regression method. Recall that we can learn the user embeddings for all the users on an e-commerce website. We create a super user embedding vector $\mathbf{v}^{(sup)}$ by averaging all available user embeddings. When the training data is limited, we require that the fitted vector should not deviate from $\mathbf{v}^{(sup)}$ too much. Second, we fit each dimension separately with an individual MART model. Based on our data analysis, we found that the values of some dimensions from the same user might be correlated. We compute pairwise Pearson Correlation Coefficient (PCC) for every two dimensions using all the learnt user embeddings from the e-commerce website, and construct the correlation matrix $\mathbf{W}^{K\times K}$, where each entry wi;j indicates the correlation degree between two dimensions. We convert all negative values to zero. We then propose to take into account both methods to refine the initially fitted value **v**(0) u in the following way

$$\min \quad \sum_k (v_{u,k} - v_{u,k}^{(0)})^2 + \mu_1 \sum_k (v_{u,k} - v_{u,k}^{(sup)})^2 + \mu_2 \sum_{k,k',k\neq k'} w_{k,k'}(v_{u,k} - v_{u,k'})^2, \quad \textbf{(3)}$$

where μ1 and μ2 are the tuning parameters. The parameter μ1 is used to "smooth" the data when the number of training instances is small or a user has very little microblogging information. While in other cases, μ1 can be simply set to a small value, e.g., 0:05. For μ2, we have found a value of 0:05 usually gives good performance. By setting the derivative w.r.t. vu;k to 0, we derive an iterative formula as follows

$$v_{u,k} \leftarrow \frac{v_{u,k}^{(0)} + \mu_1 v_{u,k}^{(sup)} + \mu_2 \sum_{k',k'\neq k} w_{k,k'} v_{u,k'}}{1 + \mu_1 + \mu_2 \sum_{k',k'\neq k} w_{k,k'}}. \quad \textbf{(4)}$$

Summary: We have built a single learner for each dimension in the transformed feature representation **v**u using a modified gradient boosting trees model. The reason why we choose MART is that its components are regression trees, and trees are shown to be effective to generate high-order and interpretable knowledge using simple plain features [14], [16], [17]. Note other tree-based ensemble methods can apply here, such as Random Forest (RF)[15]. In our experiments, we have found MART is sightly better than RF, and therefore we adopt MART as the fitting model.

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].

A. The General SVD Feature Framework for Product Recommendation

SVD Feature [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

$$\begin{aligned} & \hat{r}_{u,p}(\boldsymbol{\alpha}^{(u)}, \boldsymbol{\beta}^{(p)}, \boldsymbol{\gamma}^{(u,p)}) \\ = \; & \mu + \sum_j b_j^{(G)} \gamma_j^{(u,p)} + \sum_j b_j^{(U)} \alpha_j^{(u)} + \sum_j b_j^{(P)} \beta_j^{(p)} \\ & + (\sum_j \alpha_j^{(u)} \mathbf{x}_j)^\top (\sum_j \beta_j^{(p)} \mathbf{y}_j), \end{aligned} \quad \textbf{(5)}$$

where $\alpha^{(u)} \in \mathbb{R}^{N_\alpha}$, $\beta^{(p)} \in \mathbb{R}^{N_\beta}$ and $\gamma^{(u,p)} \in \mathbb{R}^{N_\gamma}$ re 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_α, N_β and N_γ respectively. Here, $b_j^{(G)}$, $b_j^{(U)}$ and $b_j^{(P)}$ are the global, user and product bias parameters respectively.

The 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 uth 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

$$\alpha_j^{(u)} = \begin{cases} 1, & j = u; \\ 0, & j \neq u. \end{cases} \qquad \beta_j^{(p)} = \begin{cases} 1, & j = p; \\ 0, & j \neq p. \end{cases}$$

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. Fori = 1 to |A|, we have _(u) |U|+i = au;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 $\alpha_{|\mathcal{U}|+k}^{(u)} = v_{u,k}$.

Coding Product Embeddings: Given a product p, we use the dimensions from (|P|+1)-th to (|P|+K)-th to code the product embedding **v**p. For k = 1 to K, we have $\beta^{(p)}_{|\mathcal{P}|+k} = v_{p,k}$

Coding The Global User-Product Feature: Since we have both user embeddings and product embeddings, we can incorporate a global feature to denote a similarity degree between a user and a product. The idea is that a user is more likely to buy a product which is closer in the unified latent feature space, therefore the corresponding entry should receive a larger global bias value. We define a global feature as follows

$$\gamma_1^{(u,p)} = \text{sim}(\mathbf{v}_u, \mathbf{v}_p), \quad \textbf{(6)}$$

where the cosine similarity is used to implement the function sim(.; .). With these coded features, for a user-product pair (u; p), we have the following factorization formula

$$\begin{aligned} & \hat{r}_{u,p}(\alpha^{(u)}, \beta^{(p)}, \gamma^{(u,p)}) \\ = & \ \mu + b_1^{(G)}\gamma_1^{(u,p)} + \sum_j b_j^{(U)}\alpha_j^{(u)} + \sum_j b_j^{(P)}\beta_j^{(p)} + \\ & \left(\mathbf{x}_u + \sum_{i=1}^{|\mathcal{A}|} a_{u,i}\mathbf{x}_i + \sum_{k=1}^{K} v_{u,k}\mathbf{x}_k\right)^\top \left(\mathbf{y}_p + \sum_{k=1}^{K} v_{p,k}\mathbf{y}_k\right) \end{aligned} \quad \textbf{(7)}$$

We use Θ to denote the parameters to learn,

$$\{\mu, b_1^{(G)}, \{b_j^{(U)}, \mathbf{x}_j\}, \{b_j^{(P)}, \mathbf{y}_j\}\}^5 \quad \textbf{(8)}$$

Parameter Learning: We employ the pairwise ranking model for parameter learning. Given a user u, we generate the positivenegative pairs of products (p; p′) in which u has purchased p (called positive) but not p′ (called negative). The pairwise ranking model assumes that the fitted value for the purchased product is larger than the one that has not been purchased by a user, i.e., Pr(ˆru;p > ˆru;p′). Furthermore, we use the sigmoid function as the loss function

$$Pr(\hat{r}_{u,p} > \hat{r}_{u,p'}) = \frac{1}{1 + e^{-(\hat{r}_{u,p} - \hat{r}_{u,p'})}} \quad \textbf{(9)}$$

Note that for pairwise ranking, we do not need to learn the user bias parameters |b(U)j|. With the above partial-order rank probability function, the overall regularized ranking loss function can be written as follows

$$\begin{aligned} \mathcal{L} = & -\sum_{u\in\mathcal{U}} \sum_{(p,p')\in\mathcal{D}_u} \log \frac{1}{1 + e^{-(\hat{r}_{u,p} - \hat{r}_{u,p'})}} + \sum_j \lambda_1 \| \mathbf{x}_j \|_2^2 \\ & + \sum \lambda_2 \| \mathbf{y}_j \|_2^2 + \lambda_3 \| b_1^{(G)} \|_2^2 + \lambda_4 \sum \| b_j^{(P)} \|_2^2, \end{aligned} \quad \textbf{(10)}$$

where Du denotes the positive-negative pairs for user u, and _s are the coefficients for ridge regularization. By minizing the loss function L, we use the stochastic gradient descent method (SGD) to learn the model parameters. Given a training instance consisting of a user u and a positive-negative pair (p; p′), the derivatives at this instance for updating the model parameters are presented as follows

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial\mathbf{x}_u} &= -e^u_{p>p'}\left\{\Delta\mathbf{y}_{p,p'} + \sum_{k'=1}^{K} \mathbf{y}_{k'}\Delta v_{p,p',k'}\right\} + 2\lambda_1\mathbf{x}_u, \\ \frac{\partial\mathcal{L}}{\partial\mathbf{x}_i} &= -a_{u,i}e^u_{p>p'}\left\{\Delta\mathbf{y}_{p,p'} + \sum_{k'=1}^{K} \mathbf{y}_{k'}\Delta v_{p,p',k'}\right\} + 2\lambda_1\mathbf{x}_i, \\ \frac{\partial\mathcal{L}}{\partial\mathbf{x}_k} &= -v_{u,k}e^u_{p>p'}\left\{\Delta\mathbf{y}_{p,p'} + \sum_{k'=1}^{K} \mathbf{y}_{k'}\Delta v_{p,p',k'}\right\} + 2\lambda_1\mathbf{x}_k, \\ \frac{\partial\mathcal{L}}{\partial\mathbf{y}_p} &= -e^u_{p>p'}\bar{\mathbf{x}}^u + 2\lambda_2\mathbf{y}_p, \\ \frac{\partial\mathcal{L}}{\partial\mathbf{y}_{p'}} &= e^u_{p>p'}\bar{\mathbf{x}}^u + 2\lambda_2\mathbf{y}_{p'}, \\ \frac{\partial\mathcal{L}}{\partial\mathbf{y}_k} &= -e^u_{p>p'}(v_{p,k}\bar{\mathbf{x}}^u - v_{p',k}\bar{\mathbf{x}}^u) + 2\lambda_2\mathbf{y}_k, \\ \frac{\partial\mathcal{L}}{\partial b_1^{(G)}} &= -e^u_{p>p'}(\gamma_1^{(u,p)} - \gamma_1^{(u,p')}) + 2\lambda_3 b_1^{(G)}, \\ \frac{\partial\mathcal{L}}{\partial b_j^{(P)}} &= -e^u_{p>p'}(\beta_j^{(p)} - \beta_j^{(p')}) + 2\lambda_4 b_j^{(P)}, \end{aligned}$$

where $\Delta\mathbf{y}_{p,p'} = \mathbf{y}_p - \mathbf{y}_{p'}$, $\Delta v_{p,p',k'} = v_{p,k'} - v_{p',k'}$, $e^u_{p>p'} = 1 - Pr(\hat{r}_{u,p} > \hat{r}_{u,p'})$, $\bar{\mathbf{x}}^u = \mathbf{x}_u + \sum_{i=1}^{|\mathcal{A}|} a_{u,i}\mathbf{x}_i + \sum_{k=1}^{K} v_{u,k}\mathbf{x}_k$ and $\bar{\mathbf{y}}^p = \mathbf{y}_p + \sum_{k=1}^{K} v_{p,k}\mathbf{y}_k$.

Applications in Cold-Start Product Recommendation: With the learnt models, we can recommend products from e-commerce websites to users in online social networking websites. In this scenario, the only information available is the microblogging features of users, i.e., **a**u. Using MART, we can derive the fitted user embeddings, i.e., ˆ**v**u = f(**a**u). We consider the following variants to rank candidate products with our proposed methods:

- Only with the fitted user embeddings

$$\hat{r}_{u,p} = bias + \left(\sum_{k=1}^{K} \hat{v}_{u,k}\mathbf{x}_k\right)^\top \left(\mathbf{y}_p + \sum_{k=1}^{K} v_{p,k}\mathbf{y}_k\right) \quad \textbf{(11)}$$

- With both the fitted user embeddings and microblogging

feature vectors

$$\hat{r}_{u,p} = bias + \left(\sum_{i=1}^{|\mathcal{A}|} a_{u,i}\mathbf{x}_i + \sum_{k=1}^{K} \hat{v}_{u,k}\mathbf{x}_k\right)^\top \left(\mathbf{y}_p + \sum_{k=1}^{K} v_{p,k}\mathbf{y}_k\right) \quad \textbf{(12)}$$

where $bias = b^{(G)} \cdot \text{sim}_{cos}(\hat{\mathbf{v}}_u, \mathbf{v}_p) + b_p^{(P)}$.

Note that all the above ranking formulae do not use the user latent vector **x**u. In another words, we do not require users made any purchases before recommending products to them. Thus, our proposed recommendation framework can be applied for cold-start recommendation.

V. RESULT

VI. CONCLUSION

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 coldstart 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 Networks13 can be explored for feature learning. We will also consider improving the current feature mapping method through ideas in transferring learning [30].

VII. REFERENCES

[1]J. Wang and Y. Zhang, "Opportunity model for e-commerce recommendation: Right product; right time," in SIGIR, 2013.

[2]M.Giering, "Retail sales prediction and item recommen-dations using customer demographics at store level," SIGKDDExplor. Newsl., vol. 10, no. 2, Dec. 2008.

[3] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," IEEE Internet Computing, vol. 7, no. 1, Jan. 2003.

[4] V. A. Zeithaml, "The new demographics and market fragmentation,"Journal of 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 SIGKDD, 2014.

[6] J. Wang, W. X. Zhao, Y. He, and X. Li, "Leveraging productadopter information from online reviews for product recommendation,"in ICWSM, 2015.

[7] Y. Seroussi, F. Bohnert, and I. Zukerman, "Personalised ratingprediction for new users using latent factor models," in ACMHH, 2011.

[8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in NIPS, 2013.

[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 colds tartin app recommendation: latent user models constructedfrom twitter followers," in SIGIR, 2013.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," CoRR, vol.abs/1301.3781, 2013.

[12] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," Computer, vol. 42, no. 8,pp. 30–37, Aug. 2009.

[13] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," Annals of Statistics, vol. 29, pp. 1189–1232,2000.

[14] L. Breiman, J. Friedman, R. Olshen, and C. Stone, Classification and Regression Trees. Monterey, CA: Wadsworth and Brooks,1984.

[15] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, Oct.2001.

[16] K. Zhou, S. Yang, and H. Zha, "Functional matrix factorizations for cold-start recommendation," in SIGIR, 2011.

[17] T. Chen, H. Li, Q. Yang, and Y. Yu, "General functional matrix factorization using gradient boosting," in ICML, 2013.

[18] T. Chen, W. Zhang, Q. Lu, K. Chen, Z. Zheng, and Y. Yu,"SVD Feature: A toolkit for feature-based collaborative filtering,"
Journal of Machine Learning Research, vol. 13, 2012.

[19] S. Rendle, "Factorization machines with libfm," ACM Trans. Intell. Syst. Technol., vol. 3, no. 3, May 2012.

[20]"Social network and click-through prediction with factorization machines," in KDDCup, 2012. [21] B. Xiao and I. Benbasat, "E-commerce product recommendation agents: Use, characteristics, and impact." MIS Quarterly, vol. 31, pp. 137–209, 2007.

[22] L. Hong, A. S. Doumith, and B. D. Davison, "Co-factorization machines: Modeling user interests and predicting individual decisions in twitter," in WSDM, 2013.

[23] H. Ma, T. C. Zhou, M. R. Lyu, and I. King, "Improving recommender systems by incorporating social contextual information, "ACM Trans. Inf. Syst., vol. 29, no. 2, 2011.

[24] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis," in SIGIR, 2014.

[25] M. Zhang, J. Tang, X. Zhang, and X. Xue, "Addressing cold start in recommender systems: a semi-supervised co-training algorithm," in SIGIR, 2014.

[26] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in SIGIR, 2002.

[27] Y. Moshfeghi, B. Piwowarski, and J. M. Jose, "Handling datasparsity in collaborative filtering using emotion and semanticbased features," in SIGIR, 2011.

[28] H. Bao and E. Y. Chang, "Adheat: an influence-based diffusion model for propagating hints to match ads," in WWW, 2010, pp.71–80.

[29] N. N. Liu, X. Meng, C. Liu, and Q. Yang, "Wisdom of the better few: cold start recommendation via representative based rating elicitation," in ACM RecSys, 2011.

[30] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEETKDE, vol. 22, no. 10, pp. 1345–1359, 2010.

[31] W. Pan, E.W. Xiang, N. N. Liu, and Q. Yang, "Transfer learning in collaborative filtering for sparsity reduction," in AAAI, 2010.

[32] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in SIGKDD, 2008.

[33] B. Li, Q. Yang, and X. Xue, "Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction," in IJCAI, 2009.

[34] L. Hu, J. Cao, G. Xu, L. Cao, Z. Gu, and C. Zhu, "Personalized recommendation via cross-domain triadic factorization," inWWW, 2013.

[35] L. Zhao, S. J. Pan, E. W. Xiang, E. Zhong, Z. Lu, and Q. Yang,“Activetransferlearningforcross-system recommendation, ”in AAAI, 2013.
[36] B. Hollerit, M. Kr¨ oll, and M. Strohmaier, “Towards linking buyers and sellers: Detecting commercial intent on twitter, ”in WWW Companion, 2013.
[37] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel, “You are who you know: Inferring user profiles in online social networks,” in WSDM, 2010.
[38] R. Zafarani and H. Liu, “Connecting corresponding identities across communities,” in ICWSM, 2009.
[39] Y. Zhang and M. Pennacchiotti, “Recommending branded products from social media,” in Seventh ACM Conference on Recommender Systems, RecSys ’13, Hong Kong, China, October12-16, 2013, 2013, pp. 77–84.
[40]“Predicting purchase behaviors from social media,” in22nd International World Wide Web Conference, WWW ’13, Riode Janeiro, Brazil, May 13-17, 2013, 2013, pp. 1521–1532.

Author's Profile:

J. Sindhuja, PG Scholar, Dept of CSE, St.Mary’s Group of Institutions, Guntur, AP, India, Email: sindhusujan@gmail.com.

Mr. Raveendra Reddy Enumula (P.hD), received the Master of Technology degree in Information Technology from the AITAM (Tekkali)-JNTUK. He is currently working as assistant Professor of CSE with St.Mary's Group of Institutions Guntur. His interest subjects are Data Mining And Warehousing, Cloud Computing, Mahine Learning, Bigdata Analytics.