

Embedding Unstructured Side Information in Product Recommendation

Fatemeh Pourgholamali^a, Mohsen Kahani^a, Ebrahim Bagheri^b, Zeinab Noorian^b

^a*Web Technology Lab, Dept. of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran*

^b*Laboratory for Systems, Software and Semantics (LS³), Ryerson University, Toronto, Canada*

Abstract

Various researchers have already engaged in using auxiliary side information within recommender applications to improve the quality and accuracy of recommendations. This side information has either been in the form of structured information such as product specifications and user demographic information or unstructured information such as product reviews. The abundance of unstructured information compared to structured information entices the use of such unstructured information in the recommendation process. Existing works that employ unstructured content have been confined to standard text modeling techniques such as the use of frequency measures or topic modeling techniques. In this paper, we propose to model unstructured content about both products and users through the exploitation of word embedding techniques. More specifically, we propose to learn both user and product representations from any type of unstructured textual contents available in different external information sources using recurrent neural networks. We then apply our learnt product and user representations on two recommendation frameworks based on matrix factorization and link prediction to enhance the recommendation task. Experimental results on four datasets constructed from the Rotten Tomatoes website (movie review aggregator database) have shown the effectiveness of our proposed approach in different real-world situations compared to the state of the art.

Keywords: matrix factorization, user and product embeddings, recurrent neural networks, recommender systems

1. Introduction

In recent years, recommender systems have received much attention from researchers in the e-commerce community due to their role in increasing sale revenues as well as alleviating the information overload problem, by providing users with personalized recommendations about products and services (Dias et al. 2008). Collaborative Filtering (CF) is among the common recommendation paradigms which models users' collaborative behavior reflected in their transactions (i.e., user-rating matrix). Different variations of CF techniques have been proposed in the literature (Xue et al. 2005; Pennock et al. 2000; Koren 2008; Barjasteh et al. 2016). The core idea of these techniques is to incorporate users' previous rating records and interaction history in order to predict future interests. The performance of various CF methods including the basic models such as memory-based and model-based approaches as well as hybrid models have been widely evaluated in different application domains (Koren 2008; Barjasteh et al. 2016). For instance, Matrix Factorization (MF) is a widely used CF technique that exploits user-product rating matrices by mapping them onto a latent feature space, which facilitates the prediction of user ratings. The efficacy of MF algorithms in improving users' satisfaction and providing effective recommendations for a large number of users has already been demonstrated in contexts such as the Netflix competition (Bell and Koren 2007) and KDD Cup 2011 (Dror et al. 2012).

One of the approaches for improving the performance of MF algorithms has been to introduce additional contextual information about the users and products by moving beyond pure user-product interaction information (Esparza, OMahony, and Smyth 2011). Various types of content related to users and products have been utilized in the literature to improve

Email addresses: fatemehjfg@gmail.com (Fatemeh Pourgholamali), kahani@um.ac.ir (Mohsen Kahani), bagheri@ryerson.ca (Ebrahim Bagheri), znoorian@ryerson.ca (Zeinab Noorian)

the recommendation task in the MF-based recommender systems, which can be classified in two main categories: 1) structured information, and 2) unstructured information.

The works that exploit structured information extract content from sources that contain structured information about users and products. For example, authors in (Leung, Chan, and Chung 2008; Leung, Chan, and Chung 2006) exploit external taxonomies or ontologies of products. Other researchers (Beilin and Yi 2013; Chen and Wang 2013; Yang and Kim 2015; Forsati et al. 2014) exploit social relationships of users such as friendship, membership, and trust relationships to augment the accuracy of recommendations. Some works employ product category and/or user demographic information that is inherently structured (Barjasteh et al. 2016; Yu et al. 2017). However, these approaches can face one important limitation, which pertains to structured information. In many cases, relying on structured specification of products is limited to only a few ecommerce websites that provide access to structured information about products. However, the fact of the matter is that many products presented on ecommerce platforms do not have structured information available. For instance, many products on websites such as Kijiji or Craigslist only have snippets of natural language text descriptions. Therefore, relying on structured information, while very effective, is a limiting factor in reality.

To overcome this limitation of structured information, several researchers have proposed to exploit generalizable types of side information, such as unstructured content available for users and/or products (Pourgholamali 2016). The main premise of these approaches is that similarities identified through unstructured content of users or products can enhance the effective recommendation of products. On this basis, existing works in the literature make certain assumptions about the unstructured content that is exploited. For example, most of these approaches assume that additional unstructured contents are derived from user reviews and therefore, inherently contain information such as sentiment words about different product aspects (Musat, Liang, and Faltings 2013; Zhang et al. 2013; Moshfeghi,

Piwowarski, and Jose 2011; Raghavan, Gunasekar, and Ghosh 2012). Our work in this paper falls within the same family of work in that we propose to use additional unstructured content about products and users to improve product recommendation. However, we provide a unique way of modeling unstructured content that does not necessarily restrict its scope to product reviews. Therefore, various types of unstructured product and user related contents can be exploited without having to be concerned with the type of content that is being used, be it user reviews or informal product specifications.

More specifically, we aim to exploit unstructured textual content from external information sources in order to improve the recommendation performance. Our proposed approach is able to capture unstructured textual data, such as informal descriptions of products and review texts provided by users. We propose a generalizable approach to exploit unstructured information to construct a unique representation for each product and user. This representation preserves the distance and geometric properties of products and users and as a result can produce reasonable neighborhoods for products and users that can be incorporated into recommendation frameworks to enhance the performance of the recommendation task. To produce the representation for users and products, we exploit the Recurrent Neural Network (RNN) architecture proposed for developing Word Embeddings (Mikolov et al. 2013b; Mikolov et al. 2013a), which have already shown to be efficient for representing words and preserving their relations in vector space.

The key contributions of our work can be summarized as follows:

1. We propose a technique to formally represent products. We are interested in finding a suitable formal representation of products that can be mined from unstructured product descriptions, user reviews as well as user ratings. Such representation would enable us to measure product similarity and relatedness in an accurate and cost-effective way, which leads to constructing suitable neighborhood models for products.
2. We propose a technique to formally represent users. This representation is constructed

from review texts and purchase history of users by preserving their temporal order; thus, users with similar purchase histories and similar reviews will be similar in their representation.

3. By adopting two recommendation frameworks, namely *i*) a graph based link prediction approach and *ii*) a feature-based matrix factorization approach, we accurately capture the relationships between products and users and exploit them in the recommendation task.
4. We perform extensive experiments to evaluate the performance of the proposed methods on four real-world datasets. The experimental results show how our proposed product and user representation approaches improve the recommendation performance in comparison to the state-of-the-art recommendation algorithms.

2. Related Work

Many state of the art recommender systems have now moved beyond the user-item matrix and include additional information that could be useful for improving the quality of recommendations. While the range and sources of side information on users and items is quite broad, it is still possible to broadly classify them into two main categories: 1) structured side information such as user attributes, i.e., age, gender, hobbies, and product attributes, i.e., properties of products; and 2) unstructured side information such as user-generated content.

The overview of these two types of information sources is illustrated in Figure 1. We provide a detailed description of the recommendation approaches that exploit such side information for the recommendation task in the remainder of this section.

2.1. Structured side information

This category primarily focuses on well-formed information sources that are related to products and users. Examples of this type of external contents for products include tax-

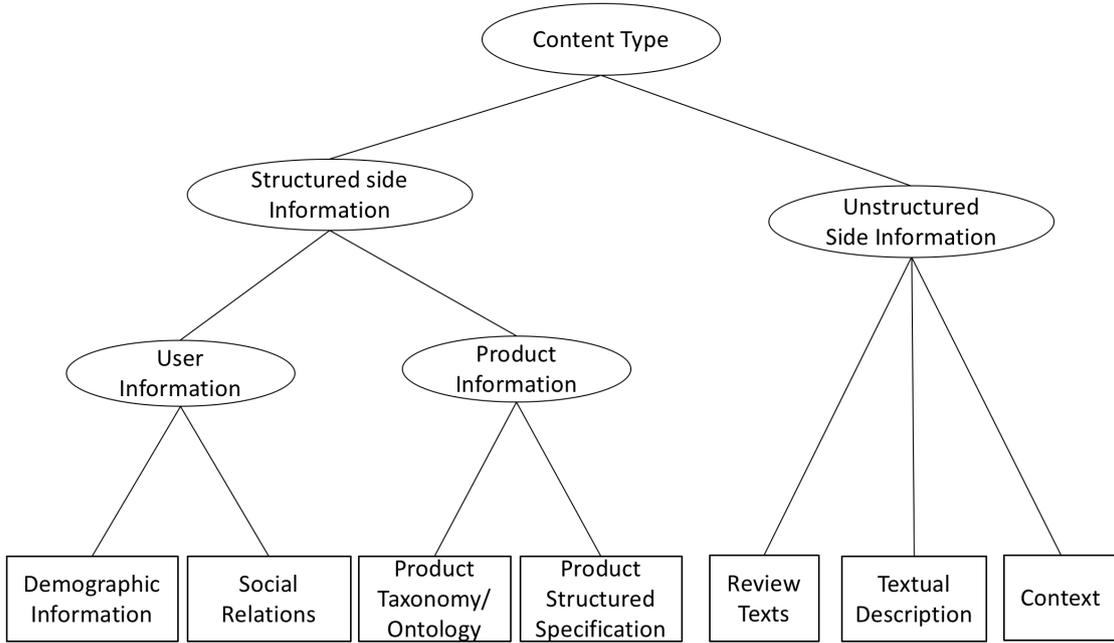


Figure 1: Classification of the types of side information exploited in the literature.

onomies, product categories, external knowledge bases, and structured specifications of products which have already been adopted by several researchers (Barjasteh et al. 2016; Leung, Chan, and Chung 2008; Yu et al. 2017). Moreover, social relations between users such as friendship, membership, trust and distrust (Forsati et al. 2014), as well as their demographic information are some examples of structured information about users (Pereira and Hruschka 2015).

Authors in (Leung, Chan, and Chung 2008) integrate item taxonomy information within the collaborative filtering model. They proposed the Cross-Level Association RuleEs (CLARE) model based on item taxonomies to infer ratings for cold start items. However, since building item taxonomies for various types of products is not a trivial task, this approach might not be easily scalable for different domains. In different work (Pereira and Hruschka 2015), the SCOAL algorithm is proposed, which discovers and classifies groups of users based on their common interests and attributes; and builds predictive models tailored to the attributes of each group. Once a newcomer arrives, the proposed algorithm decides which group a new

user belongs to, and makes a recommendation accordingly.

Authors in (Barjasteh et al. 2016) propose a matrix factorization approach which constructs a similarity matrix based on the structured content of products available on e-commerce websites. First, they filter out cold items from the original matrix and then fill in the missing values for the non-cold items. Afterwards, they transduct the information from non-cold items to cold items by applying spectral clustering on the similarity matrix. However, the performance of this approach depends on user and product structured information which is not always available on all websites.

Several other approaches exploit the social relations of users as side information. In (Forsati et al. 2014), trust and distrust relations between users are incorporated into a factorization based recommendation system. In this work, the model learns latent features of users in such a way that the maximum similarity degree of those users who are distrusted by a particular user does not exceed the minimum similarity degree of the trusted users. The authors conduct experiments on the epinions dataset which allows users to classify other users into trusted and distrusted friends based on the quality of their provided reviews. However, most commercial websites do not support social structures; hence, such an approach is not always applicable to all domains.

2.2. Unstructured side information

Recently, adopting different types of unstructured user-generated information in recommender systems has gained wide popularity. In this section, we review some of the existing work in recommender systems, which incorporate textual reviews and comments in the recommendation task. In particular, existing works can be categorized into three main approaches: 1) review-based approaches; 2) text-based approaches; and 3) context-based approaches.

2.2.1. Review based approaches

One of the sources of unstructured information that has recently increased in importance in recommender systems research is the review texts posted by users about products (Musat, Liang, and Faltings 2013). Many approaches exploit opinionated features of reviews such as sentiment words and strengths (Zhang et al. 2013; Pero and Horvth 2013); review emotions (Moshfeghi, Piwowarski, and Jose 2011); reviews on aspects of products (Musat, Liang, and Faltings 2013; Ganu, Kakodkar, and Marian 2013); and review helpfulness (Raghavan, Gunasekar, and Ghosh 2012).

In (Zhang et al. 2013), the authors exploit sentiment analysis techniques to detect sentiment lexicons in reviews as well as emoticons in order to classify the reviews as positive or negative. Considering the value of 1 for positive reviews and -1 for the negative reviews, they build a user-product rating matrix as the input for the collaborative filtering model. In (Musat, Liang, and Faltings 2013), the researchers extract topics, i.e., aspects, from the reviews posted by each user and build user profiles accordingly. For each product that a user has reviewed, they compute the similarity degree of the extracted topics with the topic profile of the user. Such a similarity degree would further serve as a confidence degree of the associated rating for that particular item.

Despite the efficacy of these approaches in various application domains, their performance is limited in relying solely on opinionated texts and special aspects of reviews. Moreover, in cases that an external sentiment lexicon is necessary, a significant amount of work should be spent to collect and construct the sentiment lexicon.

2.2.2. Text based approaches

In addition to review texts, other types of textual content have also been utilized in the recommendation task. In the news recommendation domain, the researchers in (Saveski and Amin 2014) proposed to build a term-news matrix from the text content of the news on

the one hand, and a user-news matrix from the users' comment on the news on the other hand. The basic assumption of this work is that each news document consists of some topics and therefore those users who have commented on this news article could form a community around such topics. The authors applied collective factorization of both matrices so news and users would be represented in a common latent space. Despite the novelty of this approach in representing users and news, their assumption might not be generalizable beyond the news domain and hence applicable to different application domains (e.g., e-commerce systems).

In (Esparza, OMahony, and Smyth 2011) an index-based approach is proposed, which introduces term based user and product profiles. To build profiles for products, sets of keywords are extracted from their descriptive text; and are weighted by the TF-IDF metric. The same approach would be applied for building user profiles. Products with the most similar profiles are chosen for recommendation.

In a different work (Yanir, Bohnert, and Zukerman 2011), the authors have proposed to incorporate user reviews into the matrix factorization model for the recommendation task. They run LDA on review texts to derive latent attributes for the MF model. The authors further propose a switching strategy to overcome the user cold start problem. As such, in case a user only provides few ratings (less than the minimum value of the support threshold), the attribute-based MF model would be used to make a recommendation; otherwise, the classical biased MF approach would be exploited. In a related approach, exploiting latent topics in review texts has been explored in the work by Bao et al (Bao, Fang, and Zhang 2014). In this work, the authors jointly model user ratings through matrix factorization and review texts through topic modeling. This is done by transforming item and user latent vectors into topic distribution parameters. While such a joint modeling approach can lead to improved performance, it might not perform as well when the order of terms and temporality between user reviews are of importance.

In a more recent work, Wang et al (Wang, Wang, and Yeung 2015) have introduced

the Collaborative Deep Learning (CDL) method that utilizes textual content of products to jointly learn representations based on side information and product ratings. They introduce a deep learning model, called stacked denoising autoencoder (SDAE) as a feature learning component which jointly works with a matrix factorization model to predict the rating of a user for a certain product.

2.2.3. Context (embedding) based approaches

Several recent research works on recommender systems attempt to build representation models for users and products by exploiting the context of products and users. Considering such information as surrounding context, these approaches apply word embedding techniques to make a semantic representation model for users and products (Zhao et al. 2016b; Grbovic et al. 2015; Vasile, Smirnova, and Conneau 2016). The work in (Zhao et al. 2016b) proposes a mechanism to solve the cold user problem in e-commerce websites by enriching the profiles of cold users with their corresponding profiles in social media. The authors designed a mapping function to identify the profile for users with limited activities in e-commerce using their profile in social media. The primary profiles for users and products in e-commerce are constructed from purchase history of users. Considering each product as a word token, they convert the historical purchase records of a user into a timestamped sequence, and adopt Word2vec (Mikolov et al. 2013b; Mikolov et al. 2013a) to learn product embeddings. To build user profiles, they considered the purchase history of users consisting of a sequence of product ids as a sentence with the user id as the sentence id; and then adopt a para2vec (Le and Mikolov 2014) method to learn user embeddings. These embeddings construct the features for items and users and later would be incorporated into a feature based matrix factorization approach. In this work, the embedding approach for representing users and items solely relies on users' purchase history and does not consider textual reviews as side information.

In another approach, the authors in (Vasile, Smirnova, and Conneau 2016) propose a

neural network architecture for representing products such that the purchase history of products as well as their structural descriptions are injected into the model to regularize item embeddings. However, such structured side information is not always available in all domains. In our paper, we aim to propose a generalizable and scalable approach to utilize different types of rich information sources in addition to the user ratings. In our approach, any forms of texts related to users or products can be exploited to construct a semantic representation for users and products. These representation models will be further formally incorporated into the recommendation process using graph-based link prediction and feature based matrix factorization approaches.

3. Approach Overview

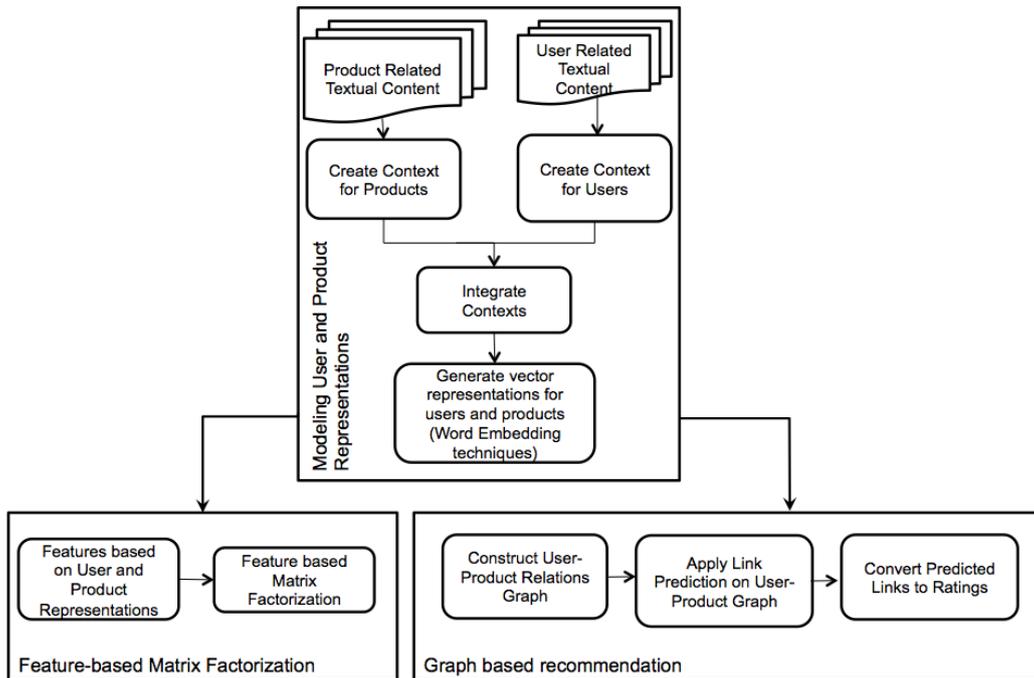


Figure 2: The overview of our proposed approach.

The goal of our research is to propose an approach for recommendation such that various types of unstructured textual information about users or products could be easily in-

corporated in the recommendation process. The approach overview, which is illustrated in Figure 2, consists of three major components:

1. *Modeling users and products based on additional unstructured textual information:* In the first step of our work, we aim to produce a unique representation for users and products by incorporating various types of unstructured side information. Since entities, i.e., products and users, are better represented within their contexts, we model product and user representations using word embedding techniques which preserve the context around entities. The detailed descriptions of the proposed product and user representations, referred to as User Semantic Representation (USR) and Product Semantic Representation (PSR) are described in Sections 4.1 and 4.2.
2. *Graph based recommendation:* We further show how USR and PSR can be incorporated into a graph based link prediction approach for the recommendation task. Considering users and products as nodes, the edges between different nodes would represent the similarity and relationships between different entities, which can be articulated based on the USR and PSR models. After building a weighted graph, we employ different link prediction strategies for the recommendation task.
3. *Feature based matrix factorization:* Given the formal representations for users and products, we combine these information into a feature-based matrix factorization framework. We obtain new sets of features from the proposed user and product representations and incorporate them as extra features into a feature based matrix factorization model. The objective is to incorporate user and product information derived from unstructured textual content into the recommendation process performed through matrix factorization.

4. Products and Users Representation Models

The idea of building product and user representations especially for improving recommendations has already been widely explored in the literature (Pereira and Hruschka 2015; Yanir, Bohnert, and Zukerman 2011; Zhao et al. 2016b; Park et al. 2015). In particular, there have been approaches that build product and user representations to address the cold start problem in recommender systems (Barjasteh et al. 2016; Zhang et al. 2013; Park et al. 2015). However, the majority of these approaches rely on structured information provided by e-commerce websites to model users and products. For instance, they incorporate product categories, tags, and property/values in order to generate the product representation (Park et al. 2015); while users' purchase history, explicit preferences, social relationships, as well as demographic information are exploited within user representation (Forsati et al. 2014; Pereira and Hruschka 2015; Zhao et al. 2016b). Our work; however, focuses on integrating unstructured textual content about or related to products and users into the recommendation process. Our proposed approach is novel in that we do not make any assumptions about the type of the textual content that is being used to build the user and product representations.

To obtain generalizable user and product representations, we exploit word embedding techniques in our work. Word embedding techniques have been extensively applied by researchers for the purpose of semantic text annotation (Sun et al. 2015), content disambiguation (He et al. 2013) and entity linking (Yang and Kim 2015). We propose to adopt word embedding approaches to systematically build user and product representations. In the following subsections, we propose product and user representation models based on word embeddings which are able to capture contextual semantic relations between different products and users.

4.1. Product Semantic Representation (PSR)

Various studies in the literature have shown that structured specification of products could very well represent products, since they are accurate specifications that have been developed by experts in that domain (Park et al. 2015). However, not all products in all product categories come with such detailed structured specifications. We aim to construct a product representation that can be used instead of structured properties of products when such structured information is not available.

The basic idea of our proposed work for PSR is to extract products descriptive contents available on the Web, which are mainly unstructured, and formally represent them based on the word embedding approaches. In PSR, we consider an unstructured text about the product, e.g., the description of a product on an e-commerce website, review texts of the product, wikipedia pages, among others, as input and exploit word embedding methods such as Word2Vec (Mikolov et al. 2013b; Mikolov et al. 2013a) and GloVe (Pennington, Socher, and Manning 2014) to represent the product in the form of a vector representation. The reason that we are interested in word embeddings is that they are distributional representations of words which give collective meaning to the words that are used in similar contexts. Therefore, by exploiting this technique and using unstructured textual contents for products, we can potentially bring products into a semantic representation space which preserves distance and geometric properties. We formalize our proposed product representation model (PSR) as follows:

Given a product p and its associated textual content t including descriptive texts, product reviews, among others, we first perform pre-processing such as removing stop words and stemming. We then replace any mentions of the target product with a unique token, rather than treating it as just a union of words. The reason for this is that the meanings of the exact words within the product name might have individual meanings that are not of interest in our work. Based on this, we then perform interleaving in the following form:

$$t'_p = \text{interleave}(t, ut_p, ilf) \quad (1)$$

where ut_p represents a unique token for product p , and ilf is an interleaving factor that defines the distance between two product tokens in the updated text. The goal of interleaving is to place ut_p in every ilf words of t . We interleave in order to make sure that all relevant words for a product are placed in a visible context to be picked up by the word embedding technique. The details of the interleaving function is as follows:

$$\text{interleave}(t, w, k) = t[1..k] + w + t[k + 1..2k] + w + \dots \quad (2)$$

where t is the textual content, w is a word, k is a constant integer specifying the interleaving window size. The operator $+$ here represents the string concatenation operation.

Given the available word embedding techniques in the literature, (Mikolov et al. 2013b; Pennington, Socher, and Manning 2014), we apply a word embedding method such as Skip-gram on t'_p to produce a model that contains embedding vectors for each word in t'_p . For example, the Skip-gram model which is a recurrent neural architecture proposed in (Mikolov et al. 2013b) predicts the surrounding words given the target product, i.e., $Pr(\text{context}|p)$. Such neural architecture produces an n-dimensional vector representation for each word. Furthermore, given that we have replaced the surface phrase representation of each product with a unique token to represent that product, each product is also represented as an n-dimensional vector within the same space. The added benefit of this is that words and products will share the same vector space and hence can be compared easily. This will place the product vector in the same space as the reviews and hence products can be aligned with each other based on their reviews, descriptions and other unstructured textual artifacts. The architecture of the Skip-gram model is illustrated in Figure 3.

Now given the fact that the word embedding approach produces v_p to represent a unique vector for product p , and by letting v_{context} to denote the associated context vector (which

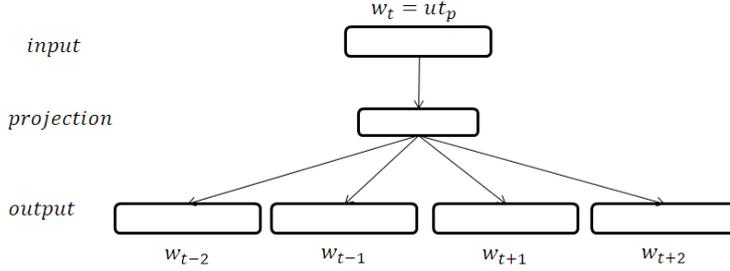


Figure 3: The Skip-gram model architecture for learning product representation. $w_{t-2}..w_{t+2}$ denotes words that surround ut_p include either review words or ut_p , depends on ilf and window size.

is the average of the vectors of the context words), the conditional probability of predicting context based on the product is characterized by a softmax function as follows:

$$Pr(context|p) = \frac{\exp(v_{context}^T \cdot v_p)}{\sum_{w=1}^W \exp(v_w^T \cdot v_p)} \quad (3)$$

where W is the size of the vocabulary.

4.2. User Semantic Representation (USR)

There are various approaches in the literature which have exploited different information sources such as user purchase history, demographic information and social relationships between users to model users. It has been shown that utilizing these information sources can effectively improve recommendation quality (Forsati et al. 2014; Zhao et al. 2016b). Our intention is to benefit from this insight to develop a user representation model that is based on the content that the user has generated. This includes two primary types of information: 1) the reviews that the user has posted for different products; and 2) the sequence of product interactions for each user.

In order to formalize the User Semantic Representation (USR), we adopt a similar approach to that of PSR, where word embedding methods are used to semantically represent users as vectors. The rationale behind our proposed USR model is to map users into a semantic representation space such that those users who have both similar perception of

products and also similar product interaction history would have similar embeddings and therefore have close vectors within the embedding space.

The intuition behind our USR model is that users who use similar terminology to describe similar products and also have similar product interaction histories have a higher likelihood of sharing the same interests. In addition to these two valuable information, we believe that temporal similarity of user interests is also of importance; therefore, we consider timestamps so that the order of user’s interactions and provided reviews is taken into consideration in the user model. We formalize our user Semantic Representation Model (USR) as follows:

Given a user u and her set of product interaction records $P = p_1, p_2, p_3, \dots$ which have been successively ordered, i.e., time preserving order, as well as the reviews posted by user R presented as $R = r_1, r_2, r_3, \dots$, respectively, we introduce context t_u for user u as follows:

$$t_u = p_1 + r_1 + p_2 + r_2 + \dots \quad (4)$$

Here, we consider all operands as strings of words and tokens and ‘+’ to denote the concatenation operator for strings. Similar to the PSR model, we then replace the username of the target user u with a unique token, u_{id} , and then adopt Equation 1 to interleave the u_{id} in t_u . Using the interleaving function, we ensure that all relevant words for a user are placed in a visible context for the user to be picked up by the word embedding technique. We obtain the final descriptive text t'_u for user u as follows:

$$t'_u = \textit{interleave}(t, u_{id}, \textit{ilf}) \quad (5)$$

We can then apply word embedding methods such as Skip-gram and GloVe on t_u to produce a model that contains embedding vectors for each word in t'_u . Applying such methods on t'_u will return the feature vector, v_u , that corresponds to user u . In USR, the ultimate objective is to predict the context (surrounding) words in a text, t'_u , given user

embedding vector v_u .

It is noteworthy to mention that, to learn the embeddings for users and products, we integrate all user and product data (i.e., t'_u, t'_p) and build one neural architecture for the learning process of our proposed USR and PSR models. Once the model is learned, we have vectors for all words in the input data including user tokens and product tokens. We consider user vectors and product vectors as the semantic representations for users and products, respectively, and utilize them in the recommendation process. The main reason why such a joint user and product embedding model is learnt is to place products and users within the same vector space and hence theoretically make them comparable. This way, we are able to not only estimate the similarity of pairs of users or pairs of products, but also predict the similarity between a user and a product given they are both placed within the same vector space.

5. Embedding-based Recommendation Models

After building semantic representation models for users and products, we incorporate these models into the recommendation models. In this paper, we adopt two distinct recommendation frameworks, namely, *i*) link prediction approach for recommendation, and *ii*) feature based matrix factorization approach. The underlying characteristic of these recommendation frameworks is their flexibility to incorporate multiple information sources for large sets of users and products, which makes them suitable frameworks to accommodate our proposed models.

5.1. Recommendation via link prediction (Graph-based)

Graphs provide abstractions for representing interactions between different entities in a network. In this context, link prediction algorithms combine the network structure and node information to predict the links which have not yet been observed between the vertices

of the graph (Li and Chen 2013). Specifically, link prediction can be applied to analyze and solve interesting problems such as predicting outbreak of a disease (Folino and Pizzuti 2012), controlling privacy in networks (Al-Oufi, Kim, and El-Saddik 2011), detecting spam emails (Huang and Zeng 2006), suggesting alternative routes based on the current traffic patterns (Yadav, Singh, and Singh 2015), among others.

In recent years, link prediction approaches have been applied for the recommendation task by several researchers (Li and Chen 2013; Cui et al. 2015; Xie et al. 2015; Chiluka, Andrade, and Pouwelse 2011) for two main reasons: 1) it can overcome the data sparsity problem and compensate for the shortcomings of the traditional transaction based approaches such as collaborating filtering methods; and 2) a graph representation reveals hidden relations between indirectly connected users and products which traditional CF methods are unable to characterize.

Given a set of users U and a set of products P , and user-product ratings as the set of transactions between users and products, a graph can be built based on the connections between the users and the products; thus, the recommendation task can be considered to be a link prediction problem between users and products on this graph. It is worth noting that some link prediction solutions work on unweighted graphs such that they would not account for the strengths of the relationships between nodes, i.e., they do not consider the quality of the transactions between users and products (DeSa and Prudncio 2011). Other graph based approaches for recommendation work on a bipartite graph model, which is limited to presenting the transactions between users and products. We argue that, capturing the weighted relations between pairs of users and pairs of products provides an opportunity to predict hidden relationships between unconnected nodes, which aligns well with the objective of recommender systems.

In this paper, we propose a weighted graph based solution for the recommendation task. Our model augments the bipartite graph by articulating the relationship between users and

products with USR and PSR models. As illustrated in Figure 4, our proposed weighted graph approach endows three different types of relationships: 1) user-user relations which is calculated based on the USR model; 2) product-product relations, which is measured by the PSR model; and 3) user-product relations, which depicts the transaction history of users.

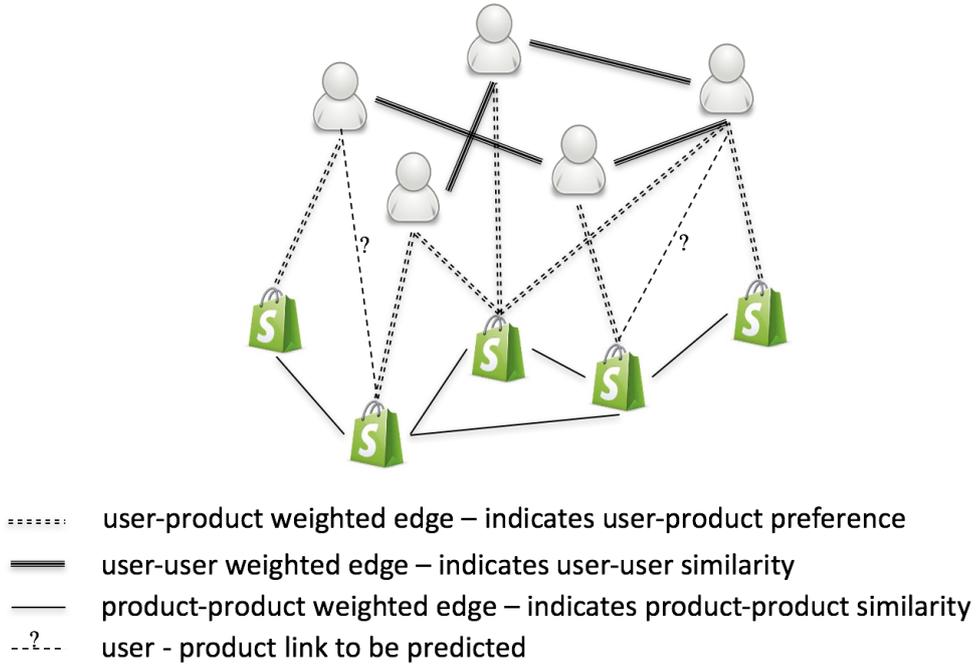


Figure 4: Graph representation model for link prediction-based recommendation

5.1.1. Graph representation

Given a set of users and products denoted by U and P , respectively, we model our problem in the form of graph $G = (G_U \cup G_{UP} \cup G_P)$, which is a graph composed of three subgraphs, G_U , G_{UP} and G_P . $G_U = (V_U, E_U)$ is a weighted graph which represents the relationships between users; $G_{UP} = (V_{UP}, E_{UP})$ denotes interests of users to products populated by the ratings that the users have given to products; and $G_P = (V_P, E_P)$ represents the relationships between products.

The user-user graph, G_U and product-product graph, G_P , are weighted and undirected graphs. In the PSR and USR models, we capture user and product representations, which

are able to depict the distance and geometric properties of users and products in a suitable way. We then apply a proper similarity measure on the corresponding vectors to capture a weighting schema for user-user and product-product links. In our experiments, we apply cosine similarity, which has been adopted by several researchers (Zhao et al. 2016b; Sun et al. 2015) to calculate the similarity between user and product pairs. So, given E_U in G_U , the weight of each user pair edge is equal to the similarity of their corresponding vectors:

$$weight(e_{uu'}) = cosine(v_u, v_{u'}) \quad (6)$$

Similarly, given E_P in G_P , the weight of each product pair edge is equal to the similarity of their corresponding feature vectors:

$$weight(e_{pp'}) = cosine(v_p, v_{p'}) \quad (7)$$

It should be noted that, since the range of ratings r_{up} may differ from one application domain to another, we propose to use a scaling function to normalize rating values so that they can adaptively reside within a specific range. Normalization of weights in link prediction based recommendation has also been adopted in previous studies such as (Cui et al. 2015; Zarrinkalam et al. 2016). Considering the maximum and minimum values of similarities as Sim_{max} and Sim_{min} and the maximum and minimum values of ratings as r_{max} and r_{min} , we adopt the following scaling function:

$$weight(e_{up}) = normalized(r_{up}) = \frac{(r_{up} - r_{min})(Sim_{max} - Sim_{min})}{r_{max} - r_{min}} + Sim_{min} \quad (8)$$

Therefore the weights of edges in G_{UP} are determined based on $weight(e_{up})$.

5.1.2. Recommendation task

After building the graph representation G , our objective is to infer the rating that a user $u \in U$ will give to product $p \in P$ for cases that u has not rated p previously. In other words, we are going to find missing links of G_{UP} by adopting an unsupervised link prediction strategy over observed links in G . There are two main categories of link prediction approaches: *i*) node neighborhood based strategies; and *ii*) path based strategies (LibenNowell and Kleinberg 2007). While both approaches are based on a predictive score function for ranking links that are likely to occur; the idea of node neighborhood strategies is to give high scores to two nodes x and y if they have sufficient number of common neighbors; and the path-based methods consider the ensemble of all paths between two nodes.

Motivated by the work in (Zarrinkalam et al. 2016), we choose the following two metrics, which are compatible with our graph context, to predict user-product links: 1) Jaccard’s Coefficient (JC): this metric gives higher values for pairs of nodes which share a higher amount of common neighbors relative to the their total number of neighbors; and 2) Adamic/Adar (AA): this metric measures the intersection of neighbor-sets of two nodes in the graph, but emphasizes on smaller overlaps. The JC and the AA metrics are applicable only on unweighted graphs, which ignore the weights between participating nodes in the graph. One of the distinguishing aspects of our graph is that it retains the strength of the links between nodes in order to differentiate between highly similar users and products and less similar ones when inferring the recommendations. Inspired by (DeSa and Prudncio 2011), the JC metric can be extended as follows to consider weights:

$$JC(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w(x, z) + w(y, z)}{\sum_{a \in \Gamma(x)} w(a, x) + \sum_{b \in \Gamma(y)} w(b, y)} \quad (9)$$

where $\Gamma(y)$ consists of the neighbors of node y and $w(x, z)$ is the weight of the edge between two nodes x and y in the graph. Also, the AA metric can be extended as follows:

$$AA(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w(x, z) + w(y, z)}{\log(1 + \sum_{c \in \Gamma(z)} w(z, c))} \quad (10)$$

Once the scores are calculated, we apply the scores to estimate the rating that user u would give to product p . We then adopt a normalization approach to map the scores of links to the rating in $[r_{min}, r_{max}]$. A simple strategy is to scale the scores to the predefined rating interval. Suppose that the maximum and minimum values that the link prediction algorithm has calculated are S_{max} and S_{min} , respectively. Considering the fact that S_{ij} is the predicted score of the edge between user i and product j , the normalized value for S_{ij} , denoted as S'_{ij} , can be calculated as follows:

$$S'_{ij} = \frac{(S_{ij} - S_{min})(r_{max} - r_{min})}{S_{max} - S_{min}} + r_{min} \quad (11)$$

With regards to the choice of the link prediction method, it is important to point that it is possible to apply homogeneous link prediction algorithms such as Adamic/Adar and Jaccard's Coefficient in our work despite the fact that the graph consists of two different types of nodes, i.e., products and users. The reason for this is that we have jointly learnt the product and user representation models and therefore the vector representation of users and products are directly comparable, making them suitable for *homogeneous* link prediction.

5.2. Feature-based matrix factorization

Traditional works on recommender systems typically use collaborative filtering to make recommendations based on matching users with similar preferences or interests (Koren 2008; Yu et al. 2012; Salakhutdinov and Mnih 2008). In recent years, with the increasing volume of online data, matrix factorization approaches that are able to incorporate auxiliary information have been widely applied and received much research interests (Barjasteh et al. 2016; Forsati et al. 2014). SVDFeature (Chen et al. 2012) is one of the widely-used frameworks for feature-based collaborative filtering, which has been designed to effectively address this

need. The flexibility in adopting different types of auxiliary information sources, a robust implementation, and having no restriction on the number and meaning of attributes for users and items, have made it an extensive benchmark framework in many studies (Zhao et al. 2016b; Zhao et al. 2016a). In this paper, we adopt this approach for the recommendation task that allows us to build factorization models which accommodate different features such as user and product representation models; making it a suitable framework to evaluate the efficacy of our proposed modeling approaches (i.e., USR and PSR models).

5.2.1. Feature-based Recommendation

The feature based framework allows users to build factorization models by incorporating additional information. It is capable of both rating prediction and collaborative ranking. This framework considers three factors in the factorization process including user features, product features, and global features (features that affect users' preferences over products).

In the following, we will discuss how we can incorporate embedding information so as to optimize the recommendation outcomes. We describe how to encode embedding information such as global, user and product features.

5.2.2. Encoding User Features

One of the important information that we can embed into our recommendation model is additional information about the users. In our work, we benefit from the geometric properties of the user semantic representation model that we have built to extract user features that can be embedded in the matrix factorization model. Based on the geometric properties, it is possible to identify the set of users that have the most similar behavioral patterns and opinions to a given user based on the similarity of their vector representations. Therefore, we select the $top - n$ most similar users to a given user to serve as the required features for representing that user. Let us denote the $top - n$ similar users to user u by $top - n_u = u_1..u_n$,

we formulate the user coding schema as:

$$\alpha_j^{(u)} = \begin{cases} sim(u, j), & j \text{ is in } top - n_u \\ 1, & j = u \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

where $\alpha^{(u)} \in R^{N_\alpha}$ is the input vector consisting of the features of user u and $sim(u, j)$ indicates the similarity of user u to user j , which can be obtained by the cosine similarity of the vector representation of both users. Therefore, the user feature set for each given user is equivalent to the similarity of that user to its $top - n$ most similar related users.

5.2.3. Encoding Product Features

Similar to the users feature set, we consider $top - n$ similar products to the target product p as the candidate to augment the feature set for p . Similarly, let us denote the most similar products to product p by $top - n_p = p_1..p_n$, we propose a product coding schema as follows:

$$\beta_j^{(p)} = \begin{cases} sim(p, j), & j \text{ is in } top - n_p \\ 1, & j = p \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

where $\beta^{(p)} \in R^{N_\beta}$ is the input vector consisting of the features of product p and $sim(p, j)$ indicates the similarity of product p to product j , which can be calculated through the cosine similarity of their vector representations from PSR.

5.2.4. Encoding Global Features

Global features denote features that relate to both users and products. In our proposed approach, users and products are comparable, primarily because they are jointly embedded within the same vector space, as explained earlier. Hence, we can consider the similarity of user and product feature vectors (v_u and v_p) as the global features, which simply states

that the more similar a product vector representation is to a user vector representation, the more likely it will be for that user to be interested in the product.

We introduce the global feature for the user-product (u, p) pair as follows:

$$\gamma_1^{(up)} = \text{cosine}(v_u, v_p) \quad (14)$$

where $\gamma_1^{(up)} \in R$ is the input vector consisting of the global feature for the pair (u, p) . These global features are also embedded into the matrix factorization model.

5.2.5. Rating Prediction

By incorporating all three types of features including user features, product features and global features, we can predict the rating of user u to product p as follows:

$$\hat{r}_{up}(\alpha^{(u)}, \beta^{(p)}, \gamma_1^{(up)}) = \mu + b_1^{(G)} \gamma_1^{(up)} + \sum_{j=1}^{N_\alpha} b_j^{(U)} \alpha_j^{(u)} + \sum_{j=1}^{N_\beta} b_j^{(P)} \beta_j^{(p)} + \left(x_u + \sum_{j=1}^{N_\alpha} \alpha_j^{(u)} x_j \right)^T \left(y_p + \sum_{j=1}^{N_\beta} \beta_j^{(p)} y_j \right) \quad (15)$$

where x_j and y_j are the d dimensional latent factors associated with each feature. Also, N_α, N_β , and 1 are the lengths of the user, product and global feature vectors, respectively.

The set of parameters would need to be optimized using stochastic gradient descent. We use $L2$ regularization for the loss function as:

$$Loss = (r - \hat{r})^2 + \text{regularization} \quad (16)$$

where r denotes the actual rating and \hat{r} denotes the predicted rating. As suggested in (Chen et al. 2012), we adopt the following update rules to update the model:

$$x_i = x_i + \eta \left(\hat{e} \alpha_i \left(\sum_j y_j \beta_j - \lambda_1 x_i \right) \right)$$

$$y_i = y_i + \eta \left(\hat{\epsilon} \beta_i \left(\sum_j x_j \alpha_j - \lambda_2 y_i \right) \right)$$

$$b_1^{(G)} = b_1^{(G)} + \eta (\hat{\epsilon} \gamma_1 - \lambda_3 b_1^{(G)})$$

$$b_i^{(U)} = b_i^{(U)} + \eta (\hat{\epsilon} \alpha_i - \lambda_4 b_i^{(U)})$$

$$b_i^{(P)} = b_i^{(P)} + \eta (\hat{\epsilon} \beta_i - \lambda_5 b_i^{(P)})$$

where $\hat{\epsilon} = r - \hat{r}$ is the difference between true rating and the predicted rating, η is the learning rate, and λ s are regularization parameters which indicate the strength of regularization.

6. Performance Evaluation

In this section, we aim to investigate the impact of the proposed user and product representation models in enhancing the recommendation process. In particular, we are interested to learn how our proposed approach improves the recommendation task in terms of relevant metrics in rating prediction and ranking order.

We conduct two classes of experiments. First, we evaluate the performance of the PSR and USR models in representing products and users and observe how our proposed models outperform other existing representation schemas in terms of rating prediction. Then by exploiting the most accurate product and user representation models, in the second set of experiments, we compare the efficiency of our proposed embedding based recommendation model with other state-of-the-art methods.

6.1. Datasets

Several datasets have been widely used to evaluate the performance of recommendation algorithms, such as MovieLens¹, EachMovie², and Netflix³. Although these datasets contain item-attribute information, none of them include users' reviews and description of products. Since our proposed user and product representation models rely on the availability of auxiliary unstructured textual content, we select a dataset from the Rotten Tomatoes website⁴. Rotten Tomatoes is a movie review aggregator database that contains large amounts of textual content about movies from different types of users (i.e., critics and regular users). We crawled reviews, ratings and other descriptive relevant content from the top rental playing movies that are available on Netflix for the period of January 1, 2000 to May 30, 2016. General statistics about these data collected from Rotten Tomatoes is summarized in Table 1. In this table, the sparsity value indicates the ratio of missing reviews per user for the movies in the dataset. The data consists of four collections. In Dataset 1, only reviews from movie critics are included. According to Rotten Tomatoes, critics are those users who have two years of published reviews available online, and all critics have demonstrated that their reviews have editorial oversight. The second dataset, Dataset 2, is a random sample from regular users. Dataset 3 is a collection from those regular users who have at least 5 ratings/reviews, and Dataset 4 is a collection from those regular users who have less than 5 ratings/reviews on Rotten Tomatoes.

6.2. Evaluation Metrics

We choose two popular metrics: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to measure the recommendation quality of our proposed method in terms of

¹<http://grouplens.org/datasets/movielens/>

²<http://grouplens.org/datasets/eachmovie/>

³<http://archive.ics.uci.edu/ml/datasets/Netflix+Prize>

⁴<https://www.rottentomatoes.com>

Dataset Name	Description	Number of users	Number of products	Number of ratings	Sparsity
Dataset 1	Critics	3,714	2,678	109,153	0.990
Dataset 2	Random sample of regular users	3,000	1,475	124,845	0.970
Dataset 3	Regular users with at least 5 ratings	3,100	1,370	56,788	0.987
Dataset 4	Regular users with less than 5 ratings	3,000	357	4,260	0.996

Table 1: Statistics of the four datasets obtained from rottentomatoes.com

rating prediction for products compared with other recommendation approaches (Salakhutdinov and Mnih 2008; Jiang et al. 2012). Formally,

$$RMSE = \sqrt{\frac{\sum_{(u,p) \in R_{test}} (r_{up} - \hat{r}_{up})^2}{|R_{test}|}} \quad (17)$$

$$MAE = \frac{\sum_{(u,p) \in R_{test}} |r_{up} - \hat{r}_{up}|}{|R_{test}|} \quad (18)$$

where r_{up} and \hat{r}_{up} presents the actual rating and the predicted rating, correspondingly; and R_{test} denotes the total number of predictions generated for all users. The lower the MAE or RMSE values are, the better the quality rating prediction would be.

Another significant measure to evaluate recommendation quality is based on ranking prediction. In real applications, it is desirable to not only predict rating of products for users, but also correctly predict the relative ranking of different products for a given user. Kendall’s tau rank correlation coefficient (Kendall 1983) is a common measure for ranking comparison. Using this metric, we compare the predicted ranking with actual ranking of

products. Let U be the set of users in the test set. The *tau* rank correlation coefficient can be calculated as follows:

$$\tau = \frac{\sum_{u \in U} \frac{cp(u) - dp(u)}{cp(u) + dp(u)}}{|U|} \quad (19)$$

where $cp(u)$ denotes the number of concordant ranked pairs and $dp(u)$ denotes the number of discordant ranked pairs. We introduce concordant ranked pairs and discordant ranked pairs as follows. Let p and p' be the two products that user u has rated with r_{up} and $r_{up'}$, respectively. Considering \hat{r}_{up} and $\hat{r}_{up'}$ as the ratings predicted by the recommender system, the ranking of pair p and p' is concordant if :

$$sign(r_{up} - r_{up'}) - sign(\hat{r}_{up} - \hat{r}_{up'}) > 0 \quad (20)$$

and the ranking of that pair is discordant otherwise.

Furthermore, we also adopt the Mean Reciprocal Rank (MRR) metric to evaluate the efficacy of different recommendation approaches including the favorite products of a user in the recommendation list.

Formally, we compute MRR as follows:

$$MRR = \frac{\sum_{u \in U} (0 + 1_{rank_u \leq K}) \frac{1}{rank_u}}{|U|} \quad (21)$$

where K is the size of the recommendation list. Considering the favorite product of user u to be the one in which the user gave the highest rating, $rank_u$ is the ranking position of u 's favorite product in the proposed recommendation list predicted by the recommender system. $1_{rank_u \leq K}$ is an indicator function which equals to 1 if $rank_u \leq K$ and equals 0, otherwise. Based on the work in (Guanliang and Chen 2014), we set K to 15.

We also adopt the Normalized Discounted Cumulative Gain (NDCG) measure to evaluate the ranking quality of the proposed recommendation model. The basic idea is that, the

recommender system is more efficient if higher rated products are ranked higher in the recommendation list. Given the actual rating of a product p as r_i which is ranked at position i in a recommendation list of size K , we calculate $DCG@K$ as follows,

$$DCG@K = \sum_{i=1}^K \frac{r_i}{\log_2(i+1)} \quad (22)$$

The normalized value of $DCG@K$ for all users in a recommendation model is computed as,

$$NDCG@K = \frac{1}{|U|} \sum_{u=1}^{|U|} \frac{DCG@K}{IDCG@K} \quad (23)$$

where $IDCG@K$ is the ideal ranking of the product determined by users.

6.3. Evaluating PSR and USR representation models

In this section, we first discuss the parameters used for learning the embedding vectors of the proposed representation models. Next, we provide series of experiments to show the effectiveness of different product modeling schemas within a recommendation task.

In our experiments, we have set some parameter values, which we report here for replication purposes. For the interleaving factor, we set *ilf* to 1 based on the empirical results that were obtained and reported in (Pourgholamali 2016). The window size for the word embedding approaches are set to 5, which is the default value suggested by Gensim⁵. We have also chosen a vector size of 200 for the product and user embeddings. We will show that a vector size of 200 exhibits the best performance in our experiments.

6.3.1. Selecting the best representation schemes

The performance of our proposed recommender system heavily relies on the quality of the product and user representation models. In this section, we report on experiments to

⁵<https://radimrehurek.com/gensim/>

identify the best representation approach.

Table 2 depicts several comparative representation schemes for modeling products. The first three rows of this table show the variations of our proposed PSR model while the next models are adopted from the literature. For the implementation of Para2Vec based models, we collected and concatenated the reviews about a given product (by a given user) and created one paragraph to represent that product (user). We then used Para2Vec to learn a vector representation for each of the products that would then represent each product (user). For instance, for the P2V+d+r model, we appended the product description of the product and all of its reviews to form one paragraph for that product. In addition, we have used the word mover’s distance (Kusner et al. 2015) as another benchmark, which leverages the word embeddings of the word2vec model to compute dissimilarity between two documents (product reviews, description or a combination thereof).

In order to evaluate the effectiveness of the PSRs, we rely on the prediction of the average rating using each of the PSRs. To do so, we apply the KNN algorithm to compute the nearest neighbors for each product based on the similarity computed from the adopted PSR. The nearest neighbors are used to predict the average rating for each product. We use the RMSE metric to measure the error of the predictions.

In addition to the product representation models, which are based on unstructured product descriptions and reviews, we also adopt a representation model based on structured product information. It is clear that when structured product information is available, it will be able to produce very efficient product similarities. We adopt the model proposed in (Park et al. 2015) as a method that has shown reasonable performance based on structured information. Given that every product p_i has its own set of specifications denoted by $S_i = \{s_{i,1}, \dots, s_{i,F}\}$, where F is the number of features and $s_{i,k}$ indicates a feature-value pair $(f_k, v_{i,k})$, we compute the similarity of products based on their structured specification as follows:

Method Name	Citation	Year	Description
PSR+d	Our Work	2017	PSR trained based on product description text with Skip-gram as the embedding function.
PSR+r	Our Work	2017	PSR trained based on product review text with Skip-gram as the embedding function.
PSR+d+r	Our Work	2017	PSR trained based on product description text and products' reviews with Skip-gram as the embedding function.
P2V+d	(Le and Mikolov 2014)	2014	Para2Vec method based on product description text with Skip-gram as the embedding function.
P2V+r	(Le and Mikolov 2014)	2014	Para2Vec method based on product review text with Skip-gram as the embedding function.
P2V+d+r	(Le and Mikolov 2014)	2014	Para2Vec method based on product description text and product reviews with Skip-gram as the embedding function.
Structured	(Park et al. 2015)	2015	A technique that primarily and solely relies on structured product information for building product representations.
WMD+d	(Kusner et al. 2015)	2015	The word mover's distance applied on product descriptions.
WMD+r	(Kusner et al. 2015)	2015	The word mover's distance applied on product reviews.
WMD+r+d	(Kusner et al. 2015)	2015	The word mover's distance applied on the collection of product description and reviews.
PH	(Zhao et al. 2016b)	2016	A method which constructs the product context based on purchase history of the users and uses Skip-gram as the embedding function.

Table 2: Baselines for comparing different product representation schemas.

$$sim(p_i, p_j) = \frac{\sum_{k=1}^F w_k sim_f(s_{i,k}, s_{j,k})}{\sum_{k=1}^F w_k} \quad (24)$$

where w_k is a weight for feature f_k . Based on the characteristics of our movie dataset, we have used the following features, namely title, genre, director, and year of release. Furthermore, we weighted genre and director to be twice as important as year and title.

Motivated by (Park et al. 2015), we measure feature similarity, sim_f , as,

$$sim_f(s_{i,k}, s_{j,k}) = \frac{v_{i,k} \cdot v_{j,k}}{\sqrt{\sum_{v \in v_{i,k}} v^2} \sqrt{\sum_{v \in v_{j,k}} v^2}} \quad (25)$$

where $v_{i,k}$ and $v_{j,k}$ are the word vectors of the corresponding features.

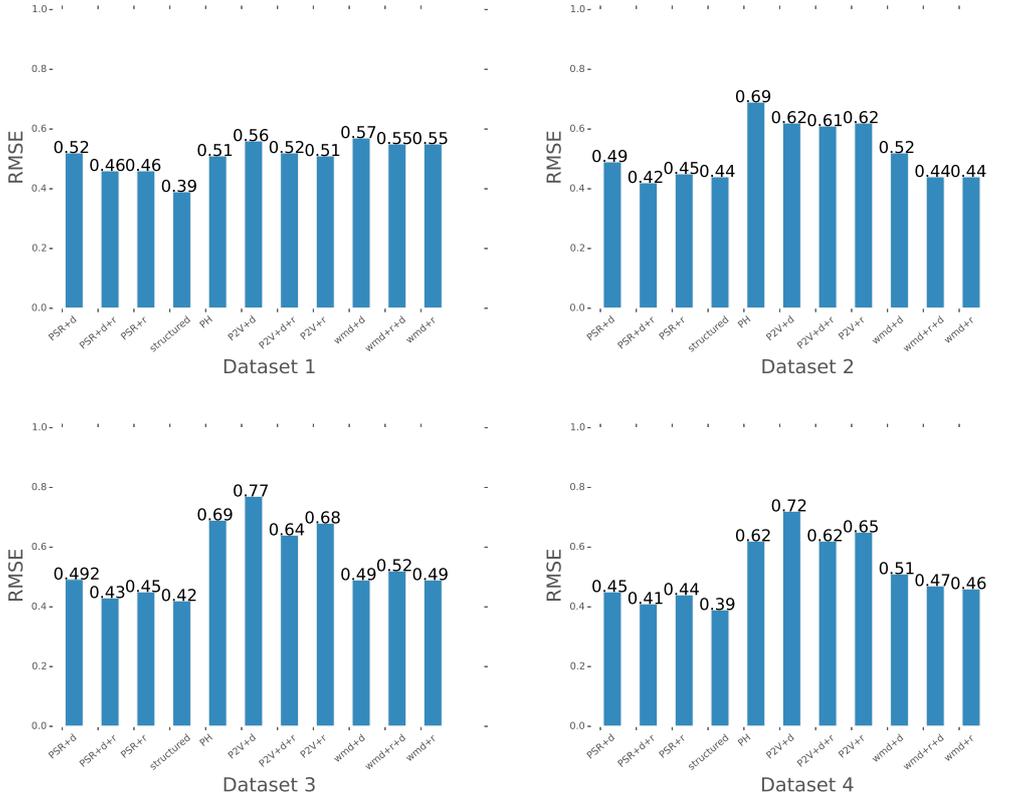


Figure 5: RMSE of predicting average rating for products based on methods introduced in Table 2.

Figure 5 shows the performance of the various product representation models shown in Table 2 over our four datasets. As expected, in most cases the structured product representation model has the least RMSE in predicting the average product rating showing that when structured information are present, they can serve as very efficient representations of the product. However, given our work has been designed for contexts where structured information is not available and there is only access to unstructured textual content, we need to explore how the other product representations that are based on unstructured textual content perform compared to the structured model. As seen from the figure, when only product descriptions (short movie summaries) are used to build the product representations, the RMSE is quite high showing that these product representations are not very accurate. This is due to the fact that short movie summaries are on average 70 words long and are very high level descriptions of the movies. Therefore, a PSR based solely on movie summaries does not yield reasonable results. However, when short movie summaries are augmented with movie reviews, then the performance of the PSRs improves significantly. As can be seen, our proposed PSR model based on both movie short summaries as well as movie reviews (d+r) shows similar performance, even better performance on Dataset 2, to the structured similarity model and outperforms the other state of the art product similarity methods such as PH (Zhao et al. 2016b) and the variations of the WMD method. This is a significant achievement in that product similarities based on structured content are known to provide the best results; therefore, our PSR which provides similar results without considering structured content can be considered to be a strong model.

A similar observation can be made for representations learnt based on Para2Vec models. In Para2Vec models, representations learnt solely based on product descriptions show a weak performance; however, this performance is improved when product descriptions are augmented with product reviews. Also, in comparison with PSR, Para2Vec models show weaker performance on all four datasets. A possible reason for this could be that when

product descriptions and reviews are appended together, they are essentially longer than one typical paragraph and are in effect a collection of many paragraphs (each paragraph representing one review); therefore, leading to a poorer performance by Para2Vec models that expect typical paragraph-length coherent input content. In summary, interleaving product and user tokens within the text and then learning a representation for the products and users on such basis as done in PSR shows to be a more accurate representation compared to when a Para2Vec model is used to learn user and product representations.

Given the fact that we have modeled users and products within the same joint embedding space, and based on the observations from the experiments in this section, we adopt the PSR+d+r model for both products and users in our next set of experiments.

Dataset	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Execution Time	5m 31.387s	6m 36.919s	7m 20.557s	3m 51.375s

Table 3: Time required for training PSR.

6.3.2. Training Time

One of the important considerations in building the product and user representation models is their reusability and time required for learning the representations. Given our product and user representation models are inherently based on word embedding techniques, they can also face similar issues that are faced by word embedding techniques. For instance, the *out of vocabulary* issue is one of the challenges faced by word embedding techniques. This issue refers to cases when a word was not observed in the training phase and hence has not been assigned a vector representation in the embedding space. In the context of our work, the out of vocabulary issue exhibits itself when new users or products are introduced into the ecosystem that were not observed in the user and product representation learning phase. In such case, techniques in the literature resort to either *smoothing* techniques or relearning the embedding space. In the context of product recommendation, smoothing techniques would not be applicable due to the importance of products and users; therefore, the reasonable

approach for addressing this problem is the periodical relearning of the embedding space. Table 3 shows the time required for training the PSR+d+r model on our four datasets⁶. As seen in the table, the required time is quite small, which can even be made to be more efficient if executed on a GPU due to the highly scalable nature of neural embedding methods. Therefore, it is feasible to relearn the product and user representations on a periodic basis.

Dataset 1				
Vector Size	50	100	150	200
PSR+d (Skipgram)	0.538	0.53	0.528	0.524
PSR+r (Skipgram)	0.473	0.475	0.482	0.46
PSR+d+r (Skipgram)	0.46	0.47	0.48	0.462
PSR+d+r(CBOW)	0.46	0.47	0.47	0.465
Dataset 2				
Vector Size	50	100	150	200
PSR+d (Skipgram)	0.51	0.53	0.522	0.49
PSR+r (Skipgram)	0.495	0.45	0.45	0.45
PSR+d+r (Skipgram)	0.477	0.445	0.442	0.425
PSR+d+r(CBOW)	0.49	0.468	0.448	0.427
Dataset 3				
Vector Size	50	100	150	200
PSR+d (Skipgram)	0.517	0.52	0.528	0.492
PSR+r (Skipgram)	0.466	0.462	0.454	0.452
PSR+d+r (Skipgram)	0.464	0.462	0.43	0.434
PSR+d+r(CBOW)	0.468	0.467	0.477	0.45
Dataset 4				
Vector Size	50	100	150	200
PSR+d (Skipgram)	0.481	0.493	0.521	0.45
PSR+r (Skipgram)	0.5	0.53	0.514	0.444
PSR+d+r (Skipgram)	0.42	0.424	0.43	0.412
PSR+d+r(CBOW)	0.456	0.45	0.476	0.47

Table 4: Impact of parameter tuning on PSR measured using RMSE.

⁶We performed our experiments on a machine with Intel(R) Xeon(R) CPU E5-2690 v2 @ 3.00GHz and 128GiB memory.

6.3.3. Impact of Vector Size and Embedding Method

It is possible that the size of the vector used for building product and user representations impacts performance. For this reason, we have extensively evaluated the impact of vector size on the performance of PSR. Our findings showed that the impact of vector size on PSR is not significant; however, a vector size of 200 has showed the best performance over all four datasets and for the different PSR models. Table 4 summarizes the impact of vector size on the RMSE of predicting average rating for products based on the three variants of our method introduced in Table 2. Furthermore, it is possible to learn word embeddings using either Skipgram models or Continuous Bag of Words (CBOW). We trained all our PSR models using both CBOW and Skipgram; the results of which show that the models trained based on Skipgram show slight improvement over CBOW. We report the results of the best CBOW model in Table 6.3.2. Based on these findings, we use the PSR+d+r model trained using Skipgram with a vector size of 200 in the rest of our experiments.

6.4. Evaluation Results

In this section we compare the performance of our recommendation models with existing state of the art models on the four datasets from Rotten Tomatoes. In the experiments, we employed a validation set to find the optimal hyperparameters where needed. Similar to the findings of other work (Wang, Wang, and Yeung 2015), the best performance was obtained when the regularization rate for the users and products are set to 0.004 along with a learning rate of 0.005. In order to evaluate our work, we examine both perspectives of rating prediction accuracy and ranking accuracy. Therefore, for the rating prediction accuracy we adopt RMSE and MAE and for the ranking accuracy, we examined Kendall’s tau rank correlation coefficient, MRR and NDCG. We have selected various state of the art baselines for comparison, which have been clearly introduced in Table 5. Our results are reported on a five-fold cross-validation strategy.

Method Name	Citation	Year	Description
User Mean	-	-	The average rating of a user for prediction.
Product Mean	-	-	The average rating of a product for prediction.
Global Mean	-	-	The average of ratings over all ratings.
PMF	(Salakhutdinov and Mnih 2008)	2008	This method is a basic form of matrix factorization which does not consider any side information over rating matrix. It offers a probabilistic foundation for regularization.
SVD++	(Koren 2008)	2008	This is a matrix factorization model that merges model based and memory based methods and extends the models by exploiting both explicit and implicit feedback of users to provide recommendations.
MFUA	(Yanir, Bohnert, and Zukerman 2011)	2011	It incorporates user attributes (including user profile and topic distributions) into basic matrix factorization algorithm for rating prediction task.
CCD++	(Yu et al. 2012)	2012	This method proposed coordinate descent based methods rules instead of ALS, and SGD, updates rank-one factors one by one.
CLiMF	(Shi et al. 2012)	2012	Collaborative Less-is-More Filtering. The model parameters are learned by directly maximizing the Mean Reciprocal Rank (MRR).
CDL	(Wang, Wang, and Yeung 2015)	2015	Collaborative Deep Learning jointly performs deep representation learning for content information and collaborative filtering for ratings representation.
PH	(Zhao et al. 2016b)	2016	A recommendation algorithm which uses both user and product embeddings to build a global feature set and uses SVDFeature to make recommendations.
EBR+Feature	Variants of Our Work	2017	Our recommendation algorithm which uses our proposed embeddings for constructing features as input for SVDFeature.
EBR+Graph+JC	Variants of Our Work	2017	A graph-based variant of our work which uses Jaccard Coefficient as the link prediction algorithm.
EBR+Graph+AA	Variants of Our Work	2017	A graph-based variant of our work which uses Adamic/Adar method as the link prediction algorithm.

Table 5: Descriptions of comparative approaches.

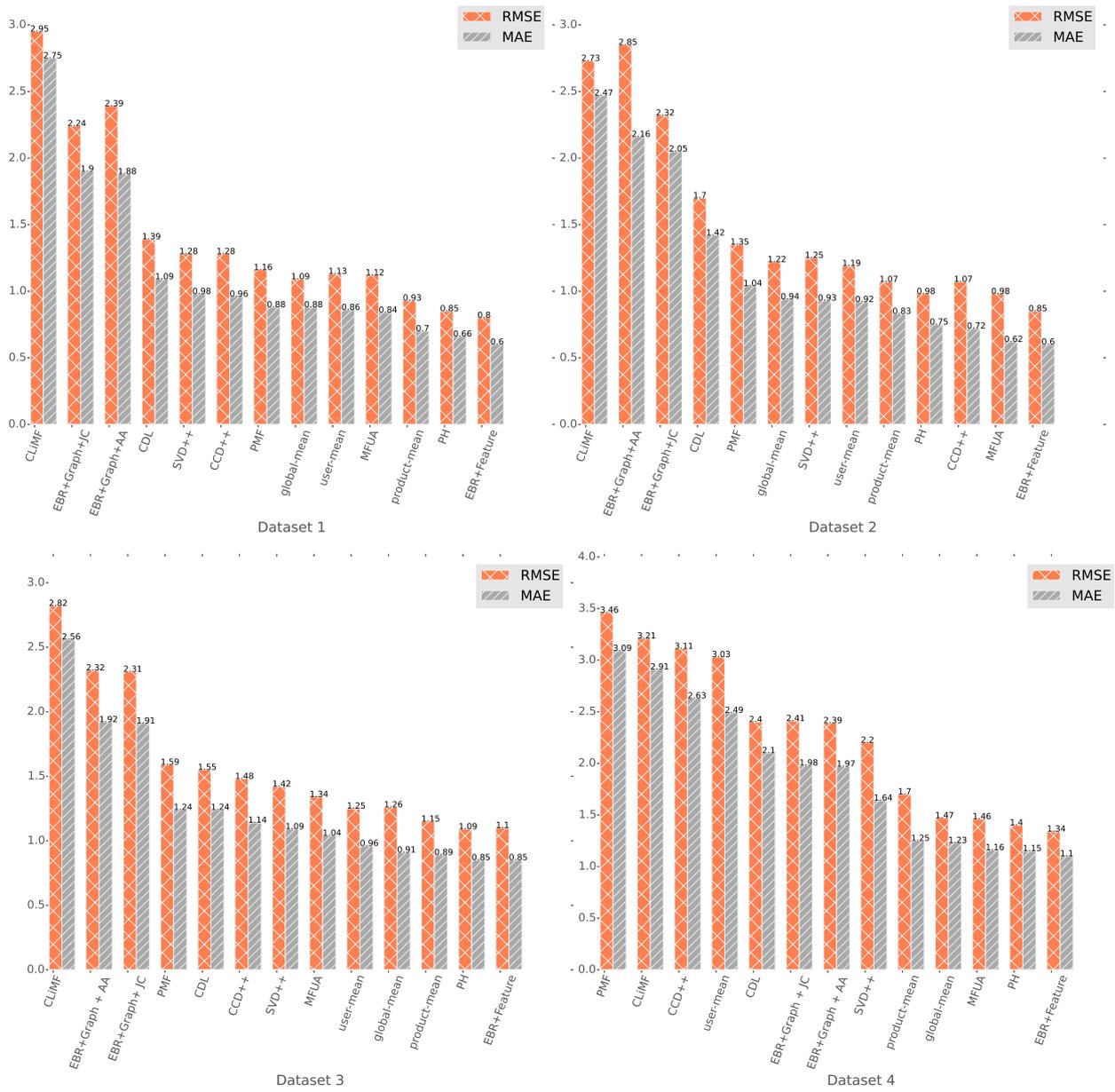


Figure 6: The comparative performance of the methods based on RMSE and MAE. Methods are sorted based on performance on MAE.

Figure 6 depicts the performance of the various baseline methods on the four datasets introduced in Table 1 in terms of RMSE and MAE metrics. The four charts visualize the two rating prediction accuracy metrics, i.e., RMSE and MAE. The lower the value for these two metrics is, the better the performance of the method would be. For the sake of clarity,

the methods have been sorted in descending order based on the MAE metric. As it can be seen, the two most competitive methods in all four datasets are our proposed method (EBR+Feature) as well as PH, which is also built on top of SVDFeature. Dataset 1 represents the opinions of movie critics and therefore includes lengthier reviews compared to the content of the other datasets. As a result, given the fact that our method relies not only on the engagement of users with products but also on the content that are posted by the users about the products, as opposed to PH that only relies on the interaction of the users and products, it produces a better performance in terms of both MAE (0.6 vs 0.66) as well as RMSE (0.8 vs 0.85). In Dataset 2 that consists of a random sample from regular users of the Rotten Tomatoes website, the performance of PH drops significantly compared to our proposed method. PH shows a weaker performance even compared to MFUA and CCD++ methods on this dataset. On this dataset, while MFUA shows a competitive performance with our proposed approach on MAE (0.6 vs 0.61), our method shows much stronger performance on RMSE (0.85 vs 0.98). Dataset 2 is the closest to the average performance of these systems on a real-world situation as it has been randomly sampled from the Rotten Tomatoes user base and movie data. In order to see whether the baseline methods are impacted by the number of reviews posted by users (cold start problem), Datasets 3 and 4 explore different subsets of the Rotten Tomatoes data for users that have five or more ratings and users that have less than 5. The expectation is that a method such as PH that builds user and product representations based on user-product interaction history performs best on Dataset 3 that has information about users with five or more ratings. As seen in the figure, our method is able to perform the same as PH even on Dataset 3 on both MAE (both 0.85) and RMSE (1.09 vs 1.1). It is important that while our method shows competitive performance on Dataset 3 (≥ 5 ratings), it is able to provide more accurate rating predictions compared to PH when applied on Dataset 4, which has users with less than 5 ratings on both MAE (1.1 vs 1.15) and RMSE (1.34 vs 1.4). This is an indication that our method is robust on all

types of datasets where users are randomly sampled, or selected from highly active users or less active (cold start) users. The explanation for this is that in contrast to PH that only focuses on user-product interaction, our proposed method relies on both the interaction and the content of the interaction as exhibited in the reviews.

It is also important to explore the performance of the methods from the perspective of ranking accuracy. There might be cases where a method is not able to accurately predict the value of the rating but is able to preserve and identify the correct rating for the products for a given user. On the other hand, the opposite might also be true for methods that produce low rating prediction error but produce completely incorrect rankings. For this purpose, Figure 7 summarizes the performance of the baselines with regards to Kendall’s Tau, MRR and NDCG. The higher the value of these metrics is, the better the predicted ranking would be. For the sake of representation, the methods have been sorted based on their Kendall’s Tau value. Kendall’s Tau is among the strictest metrics for evaluating ranking conformance. As seen in the figure, PH and our proposed method are the best performing methods in terms of ranking prediction. On the first three datasets, PH and our proposed method show competitive performance where our method has a slightly better performance on all metrics.

However, the most important observations are again on Dataset 4. As seen on this dataset that includes users with less than five reviews and their associated products, the performance of the PH method in terms of ranking accuracy drops significantly, while our proposed method is able to maintain its performance, placing second best among the baselines after the product-mean method. One of the interesting observations is that the two rather simple baselines based on global-mean and product-mean produce quite reasonable ranking prediction results despite their simplicity. This is an indication that when ranking is of importance, methods such as global-mean and product-mean that are quite easy to compute can produce competitive performance specially when dealing with cold start users. From a rating prediction perspective, product-mean seems to produce reasonable results as

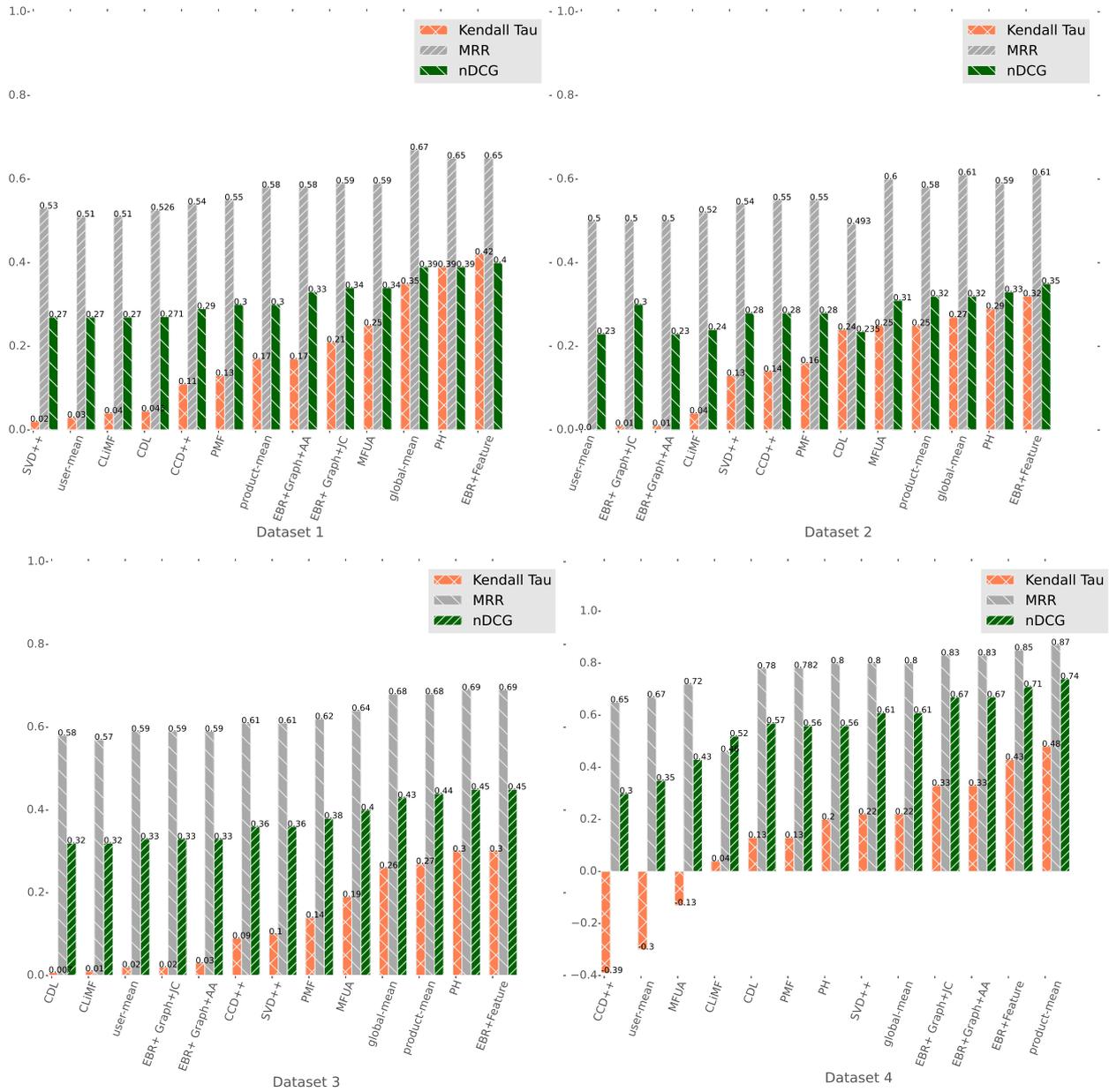


Figure 7: The comparative performance of the methods based on Kendall's Tau, MRR and NDCG. Methods are sorted based on performance on Kendall's Tau.

well on all four datasets, again pointing to the fact that this simple method can generate acceptable results with little computation.

Another important point to mention is that our proposed method shows better perfor-

mance compared to the baselines, especially the stronger baseline PH, when there is a higher sparsity in the dataset. In other words, when fewer reviews are observed, the advantage of our proposed method becomes clearer. When sparsity increases (as for cold start users) such as the case in Dataset 4, our method shows superior performance. This is primarily due to the fact that our method leverages information beyond user-product interactions in the form of user review content for building user and product representations. This way, additional inferences can be made that are beyond what is captured in PH.

It is also important to point to the work by Wang et al (Wang, Wang, and Yeung 2015), which has been reported to show better performance than SVDFeature by the authors. Our experiments showed that CDL does not perform as well as expected on our four datasets⁷. We believe that this can be due to two reasons: 1) CDL only considers product-related side information and does not explicitly build a model for the users. 2) Rating information in CDL are in binary form and do not cover a whole spectrum of rating values. For instance, in their experiments on Netflix data, the authors extract only positive ratings (rating 5) for training and testing and also remove any users with less than 3 positive ratings and movies without a plot. Therefore, while CDL is quite strong for non-cold products and binary ratings, it does not show competitive performance to other methods on our four datasets.

We would like to point that when comparing Kendall’s Tau, MRR and NDCG values in Figure 7 for the different datasets, one can see that the values reported for Dataset 4 is relatively higher than the values for the other three datasets. This difference needs a clear justification as Dataset 4 is the harder dataset that contains cold start items. After looking into the reasons for this, we believe that the higher values for the three metrics in Dataset 4 might be due to the need to rank a smaller number of items. In other words, one possible explanation could be that given this is a cold start dataset; therefore, the number of items

⁷Based on the implementation of CDL provided by the authors. We thank Wang Hao for his invaluable time and help.

that need to be ranked is less compared to the warm datasets, and therefore, the likelihood of a better ranking increases in this dataset and hence higher ranking metric values are observed. This is in contrast to rating metrics. As seen in Figure 6, Dataset 4 shows the weakest results in terms of MAE and RMSE compared to the other three datasets due to the cold start nature of this dataset.

Now, while the feature-based matrix factorization model augmented with the product and user representations performs very well on a range of different datasets, the link prediction models are not as efficient for either rank or rating prediction. There is one exception for this case on predicting rankings on the sparse cold start Dataset 4. The link prediction strategy augmented with user and product information performs quite well for predicting the ranking of cold products and shows a comparable performance to the variation that uses feature-based matrix factorization. This can be explained by the fact that matrix factorization models do not perform too well when the dataset is quite sparse; however, in such cases, the models based on graph-based link prediction show reasonable performance. Our observation of the performance of the graph-based link prediction model can be summarized as follows: *i*) link prediction models are not as efficient in predicting accurate rating values; *ii*) when the dataset of products consists of non-cold start products and users, link prediction methods are not as efficient as the variants of matrix factorization methods for ranking prediction; and *iii*) link prediction methods provide acceptable performance on rank prediction when dealing with sparse and cold start datasets. Even in such cases, their performance is not better than our proposed variant of the feature-based matrix factorization model. The reason for this is that our factorization model uses information beyond user-product interaction data and incorporates user and product unstructured data and hence alleviates the cold start and sparsity issues.

In summary, our experiments show that our proposed user and product representation models, which are based on the incorporation of user reviews into a word embedding mech-

anism, are able to show superior performance compared to the state of the art baselines on both rating and ranking metrics. This is especially more evident and of significantly higher importance because our proposed method is able to show improved performance on a range of user types ranging from randomly selected users, movie critics (expert users), highly active users as well as cold start users. The better performance of our proposed method on the cold start users (Dataset 4) is specially encouraging and noteworthy, pointing to the potential benefit of using unstructured user-generated content such as reviews in building more effective user and product representations.

7. Concluding Remarks

In this paper, we have made progress towards employing unstructured content in recommender systems. In particular, we have proposed a generalizable approach which is able to capture any type of unstructured textual data related to users and products in order to construct unique representations for users and products. We have systematically shown how such representations can be incorporated into two recommendation frameworks based on matrix factorization and link prediction to enhance the recommendation task. We experimentally investigated the potential of such representations of users and products as side information to overcome data sparsity and cold start problems in the state of the art recommender systems. In summary, our results showed that more accurate recommendation in terms of ranking and rating prediction can be obtained especially for the increasingly tougher problem of cold start products and users by incorporating user-generated content, indicating that such information can indeed be beneficial for the recommendation process.

This research opens up exciting directions for future work. It would be interesting to extend our approach with the ability to recommend cold products with absolutely no ratings to users. Given our proposed representation model of products, we will explore the possibility of developing a review selection model to select ratings and reviews of non-cold products

with similar vector representation to the cold start product; and then transfer a subset of them as potential reviews for the cold product. The main challenge in this direction is to select an optimal number of those reviews that reflect the central opinion of users and cover the major aspects of the cold product.

Another possible avenue to extend this research is to enrich the reviews of cold products by collecting more relevant reviews from microblogging services, e.g., Twitter. We intend to investigate a review transfer approach to retrieve those tweets that express user feedback about cold products. We can potentially apply microblogging summarization techniques to summarize the set of tweets to serve as the generated review for the cold products and transfer them to e-commerce websites. The main challenge of this would be related to disparities between users' feedback and the informal language they use in expressing their opinions about products. An interesting approach would be to extend our proposed product/user embedding models by incorporating users' social contextual features, which can help to identify the tweets that are most relevant to the cold products.

References

- Al-Oufi, S., H. Kim, and A. El-Saddik (2011). "Controlling privacy with trust-aware link prediction in online social networks". In: *Third International Conference on Internet Multimedia Computing and Service*, pp. 86–89.
- Bao, Yang, Hui Fang, and Jie Zhang (2014). "TopicMF: Simultaneously Exploiting Ratings and Reviews for Recommendation". In: *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAAI '14. AAAI Press, pp. 2–8.
- Barjasteh, I. et al. (2016). "Cold-Start Recommendation with Provable Guarantees: A Decoupled Approach". In: *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 28.6, pp. 1462–1474.
- Beilin, Liu and Sun Yi (2013). "Survey of personalized recommendation based on society networks analysis". In: *Proceedings of 6th International Conference on Information Management, Innovation Management and Industrial Engineering*. IEEE, pp. 337–340.

- Bell, Robert M and Yehuda Koren (2007). “Lessons from the Netflix prize challenge”. In: *Acm Sigkdd Explorations Newsletter* 9.2, pp. 75–79.
- Chen, L. and F. Wang (2013). “Preference-based clustering reviews for augmenting e-commerce recommendation”. In: *Knowledge Based Systems* 50, pp. 44–59.
- Chen, Tianqi et al. (2012). “SVDFeature: A toolkit for feature-based collaborative filtering”. In: *Journal of Machine Learning Research* 13, pp. 3619–3622.
- Chiluka, Nitin, Nazareno Andrade, and Johan Pouwelse (2011). “A link prediction approach to recommendations in large-scale user-generated content systems”. In: *Proceedings of European Conference on Information Retrieval, ECIR’11*. Springer International Publishing, pp. 189–200.
- Cui, Yiwen et al. (2015). “Heterogeneous Network Linkage-weight Based Link Prediction in Bipartite Graph for Personalized Recommendation”. In: *Procedia Computer Science* 91, pp. 953–958.
- DeSa, H. R. and Ricardo BC Prudncio (2011). “Supervised link prediction in weighted networks”. In: *Proceedings of Neural Networks, The 2011 International Joint Conference on. IJCNN ’11*. IEEE, pp. 2281–2288.
- Dias, M. Benjamin et al. (2008). “The Value of Personalised Recommender Systems to e-Business: A Case Study”. In: *Proceedings of the 2008 ACM Conference on Recommender Systems. RecSys ’08*. New York, NY, USA: ACM, pp. 291–294.
- Dror, Gideon et al. (2012). “The yahoo! music dataset and kdd-cup11”. In: *In KDD Cup*, pp. 3–18.
- Esparza, Sandra Garcia, Michael P OMahony, and Barry Smyth (2011). “Effective product recommendation using the real-time web”. In: *Research and Development in Intelligent Systems XXVII*. Springer, pp. 5–18.
- Folino, F. and C. Pizzuti (2012). “Link prediction approaches for disease networks”. In: *Information Technology in Bio- and Medical Informatics, ITBAM*, pp. 99–108.
- Forsati, Rana et al. (2014). “Matrix factorization with explicit trust and distrust side information for improved social recommendation”. In: *ACM Transactions on Information Systems (TOIS)* 32.4, p. 17.
- Ganu, G., Y. Kakodkar, and A. Marian (2013). “Improving the quality of predictions using textual information in online user reviews”. In: *Information Systems* 38.1, pp. 1–15.
- Grbovic, Mihajlo et al. (2015). “E-commerce in Your Inbox: Product Recommendations at Scale”. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 1809–1818.

- Guanliang, Chen and Li Chen (2014). “Recommendation based on contextual opinions”. In: *Proceedings of the 22nd Conference on User Modeling, Adaptation, and Personalization Denmark, Springer, UMAP '14*, pp. 61–73.
- He, Z. et al. (2013). “Learning entity representation for entity disambiguation”. In: *In Proceedings of Association for Computational Linguistics*. ACL, pp. 30–34.
- Huang, Z. and D.D. Zeng (2006). “A link prediction approach to anomalous email detection”. In: *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 1131–1136.
- Jiang, Meng et al. (2012). “Social contextual recommendation”. In: *Proceedings of the 21st ACM International conference on Information and knowledge management, CIKM '12, New York, NY, USA*. ACM, pp. 45–54.
- Kendall, M.G. (1983). “A new measure of rank correlation”. In: *Biometrika* 30, pp. 81–93.
- Koren, Yehuda (2008). “Factorization meets the neighborhood: a multifaceted collaborative filtering model”. In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 426–434.
- Kusner, Matt et al. (2015). “From word embeddings to document distances”. In: *International Conference on Machine Learning, ICML '15*, pp. 957–966.
- Le, Q.V. and T. Mikolov (2014). “Distributed Representations of Sentences and Documents”. In: *International Conference on Machine Learning, ICML '14*, pp. 1188–1196.
- Leung, C.W.K., S.C.F. Chan, and F.L. Chung (2006). “A collaborative filtering framework based on fuzzy association rules and multiplelevel similarity”. In: *Knowledge and Information Systems* 10, pp. 357–381.
- (2008). “An empirical study of a cross level association rule mining approach to cold start recommendations”. In: *Knowledge Based Systems* 21, pp. 515–529.
- Li, Xin and Hsinchun Chen (2013). “Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach”. In: *Decision Support Systems* 54, pp. 880–890.
- LibenNowell, David and Jon Kleinberg (2007). “The linkprediction problem for social networks.” In: *Journal of the American society for information science and technology* 58, pp. 1019–1031.
- Mikolov, T. et al. (2013a). “Distributed representations of words and phrases and their compositionality”. In: *In Advances in neural information processing systems (NIPS)*, pp. 3111–3119.
- Mikolov, T. et al. (2013b). “Efficient Estimation of Word Representations in Vector Space”. In: *arXiv preprint arXiv:1301.3781*.

- Moshfeghi, Y., B. Piwowarski, and J.M. Jose (2011). “Handling data sparsity in collaborative filtering using emotion and semantic based features”. In: *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 625–634.
- Musat, C.C., Y. Liang, and B. Faltings (2013). “Recommendation using textual opinions”. In: *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI '13*, pp. 2684–2690.
- Park, D. H. et al. (2015). “Retrieval of Relevant Opinion Sentences for New Products”. In: *international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 393–402.
- Pennington, J., R. Socher, and CD. Manning (2014). “Glove: global vectors for word representation”. In: *Proceedings EMNLP'14*, pp. 1532–1543.
- Pennock, David M et al. (2000). “Collaborative filtering by personality diagnosis: A hybrid memory-and model-based approach”. In: *Proceedings of the 6th conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., pp. 473–480.
- Pereira, A. L. V. and E. R. Hruschka (2015). “Simultaneous co-clustering and learning to address the cold start problem in recommender systems”. In: *Knowledge Based Systems* 42.19, pp. 6784–6794.
- Pero, S. and T. Horvth (2013). “Opinion-driven matrix factorization for rating prediction”. In: *Proceedings of the 21st International Conference on User Modeling, Adaptation and Personalization, UMAP '13*, pp. 1–13.
- Pourgholamali, Fatemeh (2016). “Mining Information for the Cold-Item Problem”. In: *Proceedings of the 10th ACM Conference on Recommender Systems*. RecSys '16. New York, NY, USA: ACM, pp. 451–454.
- Raghavan, S., S. Gunasekar, and J. Ghosh (2012). “Review quality aware collaborative filtering”. In: *Proceedings of the 6th ACM Conference on Recommender systems, RecSys '12*, pp. 123–130.
- Salakhutdinov, Ruslan and Andriy Mnih (2008). “Bayesian Probabilistic Matrix Factorization using Markov Chain Monte Carlo”. In: *Proceedings of the 25th international conference on Machine learning, ICML '08*. ACM, pp. 880–887.
- Saveski, M. and M. Amin (2014). “Item cold-start recommendations: learning local collective embeddings”. In: *Proceedings of the 8th ACM Conference on Recommender Systems, RecSys '14*. ACM, pp. 89–96.
- Shi, Yue et al. (2012). “learning to maximize reciprocal rank with collaborative less-is-more filtering”. In: *Proceedings of the Sixth ACM Conference on Recommender Systems, RecSys '12*. ACM, pp. 139–146.
- Sun, Y et al. (2015). “Modeling mention, context and entity with neural networks for entity disambiguation”. In: *International Joint Conference on Artificial Intelligence, IJCAI'15*. Springer, pp. 1333–1339.

- Vasile, Flavian, Elena Smirnova, and Alexis Conneau (2016). “Meta-Prod2Vec-Product Embeddings Using Side-Information for Recommendation”. In: *arXiv preprint arXiv:1607.07326*.
- Wang, Hao, Naiyan Wang, and Dit-Yan Yeung (2015). “Collaborative deep learning for recommender systems”. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 1235–1244.
- Xie, Feng et al. (2015). “A link prediction approach for item recommendation with complex number”. In: *Knowledge-Based Systems* 81, pp. 148–158.
- Xue, Gui Rong et al. (2005). “Scalable collaborative filtering using cluster-based smoothing”. In: *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 114–121.
- Yadav, A., Y.N. Singh, and R.R. Singh (2015). “Improving routing performance in AODV with link prediction in mobile adhoc networks”. In: *Wireless Personal Communications* 83, pp. 603–618.
- Yang, Eun Suk and Yu Seop Kim (2015). “Hallym: Named Entity Recognition on Twitter with Induced Word Representation”. In: *Proceedings of the ACL 2015 Workshop on Noisy User-generated Text*. Association for Computational Linguistics, pp. 72–77.
- Yanir, Seroussi, Fabian Bohnert, and Ingrid Zukerman (2011). “Personalised rating prediction for new users using latent factor models”. In: *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*. ACM, pp. 47–56.
- Yu, Hsiang-Fu et al. (2012). “Scalable Coordinate Descent Approaches to Parallel Matrix Factorization for Recommender Systems”. In: *IEEE International Conference on Data Mining, ICDM’12*. ACM, pp. 765–774.
- Yu, Yonghong et al. (2017). “Attributes coupling based matrix factorization for item recommendation”. In: *Applied Intelligence* 43.3, pp. 521–533.
- Zarrinkalam, Fattane et al. (2016). “Inferring Implicit Topical Interests on Twitter”. In: *Proceedings of European Conference on Information Retrieval, ECIR’16*. Springer International Publishing, pp. 479–491.
- Zhang, W. et al. (2013). “Generating virtual ratings from chinese reviews to augment online recommendations”. In: *ACM Transactions on intelligent systems and technology (TIST)* 4, p. 9.
- Zhao, Wayne Xin et al. (2016a). “Mining Product Adopter Information from Online Reviews for Improving Product Recommendation”. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 10, pp. 953–958.

Zhao, W.X. et al. (2016b). “Connecting Social Media to E-Commerce: Cold-Start Product Recommendation Using Microblogging Information”. In: *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 28 (5), pp. 1147–1159.