

Convolutional Neural Network and Topic Modeling based Hybrid Recommender System

Hira Kanwal¹, Muhammad Assam², Abdul Jabbar³, Salman Khan⁴, Kalimullah⁵

Military College of Signals¹,

National University of Science and Technology (NUST)

Islamabad, Pakistan

Department of Computer Science¹

Khwaja Fareed University of Engineering & Information Technology (KFUEIT)

Rahim Yar Khan, Pakistan

College of Computer Science & Technology^{2,3,5},

Zhejiang University, China

Department of Software Engineering⁴,

University of Science & Technology Bannu, Pakistan

Abstract—In today's personalized business environment, organizations are providing bulk of information regarding their products and services. Recommender system has various accomplishment on exploiting auxiliary information in matrix factorization. To handle data sparsity problem most recommender systems utilized deep learning techniques for in-depth analysis of item content to generate more accurate recommendations. However, these systems still have a research gap on how to handle user reviews effectively. Reviews that were written by users contain a large amount of information that can be utilized for more accurate predictions. This paper proposes a Hybrid Model to address the sparsity problem, convolutional neural network and topic modeling for recommender system, which extract the contextual features of both items and users by utilizing Deep Learning Convolutional Neural Network (CNN) along with Topic Modeling (Lda2vec) technique to generate latent factors of user and item. Topic Modeling is used to capture important topics from side information and deep learning is used to provide contextual information. To demonstrate the effectiveness of the research, an extensive experimental sets were performed on four public datasets (Amazon Instant Video, Kindle store, Health and Personal Care, Automotive). Results demonstrate that the proposed model outperformed the other state of the art approaches.

Keywords—Recommender system; collaborative filtering; Lda2vec; Convolutional Neural Network (CNN); data sparsity problem; user reviews

I. INTRODUCTION

Recommender systems have become the core component of many e-commerce organizations (i.e. movies web sites, e-libraries, articles, news, music, etc.) which avails it to predict the liking and disliking of users. To increase their business revenues companies are widely using these intelligent systems. Recommender systems gained much importance with extensive usage of the internet. It focuses on user needs and providing what they possibly want. Recommender systems are information retrieval system which help users to get their required data from this bulk of information [1]. Both users and organizations can get benefit from recommender systems. It helps user to save plenty of time while searching for online

product and helps in improving the decision-making process of the organization [2].

Various recommendation techniques have been proposed in literature and are classified as collaborative filtering technique (CF), content-based filtering technique (CBF) and hybrid filtering technique (HF) [1]–[4]. Amazon, Google News, Netflix and many other organizations are using Recommender System to target their customers. CF is one of the most widely used techniques in recommender system it can make recommendations on the basis of historical information or previously buying behavior of users to predict which item is liked by the user. This technique is based on user-item rating matrix which utilizes given users rating and predicts unknown ratings in the sparse matrix [5].

CF is further categorized into memory-based collaborative filtering and model-based collaborative filtering. Memory-based collaborative filtering applies the entire database i.e. likes, votes, clicks, etc. for rating prediction. In model-based filtering, deep learning techniques are used to construct models, train machine to learn those models and use those models to predict ratings of unrated items [5]–[7].

Cognitive filtering is another name of content-based filtering. CBF recommend item even if no rating to that item is available in the database. It is based on the content provided by user in the form of rating or reviews. The more information provided by the user more accurate recommendations will be provided [8]. It focuses on machine learning algorithms that capture user and item choices into user and item profiles respectively and recommend items to users having higher similarity with the user profile. To understand what user and item profile are considered movies as items that have different actor, director, genre etc. and user profile has demographic information, user clicks, ratings and selected items i.e. movie [9], [10].

Hybrid filtering is the union of both collaborative and content-based filtering. Both CF and CB filtering have their own pros and cons, the suitability of each approach depends on the situation in which it is used. In order to get benefit from both of them, hybrid filtering is used which suppresses the limitation of one technique alone. The hybrid filtering technique provides

more accurate recommendations than the other two techniques and the performance of recommender systems using a hybrid filtering technique is much more than RS which separately using content or collaborative filtering technique. Different hybrid filtering techniques have been proposed in the literature, however, recommender system still requires some improvements on issues like data sparsity and cold-start problem [2].

Data sparsity problem is one of the most challenging assignments in recommender system. Data sparsity problem occurs when the user rated only a few items or there exist deficient feedback data. Many real-time applications are facing this sparsity issue. CF fail to provide recommendation to user with scarce rating data. Most CF use rating matrix for recommendation task and ignores the important information available in user reviews which improves rating prediction. Various techniques have been proposed in the literature to handle data sparsity problems by combining user ratings and reviews. However, there exists a research gap of data sparsity in recommender system which need improvements and more accurate recommendations [11], [12].

To deal with this data sparsity problem in RS, a hybrid model is proposed which fuses rating data along with the information (reviews) of both user and item. The proposed model is termed as the Hybrid model of Convolutional Neural Network and Topic Modeling (HCNNTM) for recommender system which handles data sparsity problem, model unites CNN + Lda2vec into PMF to achieve latent factors of both the user and item enriched with topic information. PMF is used because it outperforms on sparse, imbalance and large datasets, which provides more efficient and accurate recommendations. Experimental evidence illustrates that using deep learning and topic modeling techniques along with the side information of both user and item content improves the performance of recommender system. The major contribution of this study suggested a Hybrid content Embedding Model for Recommender System, HCNNTM, which combines topic modeling with deep learning techniques to provide topic enriched contextual features of both user and item which will improve accuracy of rating prediction. This research uses two convolutional neural networks for item side information and Lda2vec for user side information and generates latent factors of both user and item. Experimental findings show that the proposed model not only handle data sparsity problem but also enhance rating prediction accuracy when compared with other state-of-art models.

The organization of this paper follows the related work of the RS in Section 2; the proposed model is described in Section 3; experimental findings and Results comparisons and discussion are presented in Section 4. Finally, Section 5 will present the conclusion and future work.

II. RELATED WORK

An immense amount of work on the recommender system using side information has been studied. This section will highlight the researchers' contribution and its relevance to the proposed technique.

Earlier, a recommender system uses either collaborative filtering or content-based filtering for recommendation task. Where, in collaborative filtering method, items are recommended on the basis of users behavior. Some examples of CF are nearest

neighboring modeling [13], matrix factorization, singular value decomposition, non-negative matrix factorization is used to handle data sparsity problem by using item collaborative filtering, user collaborative filtering or both [14].

Probabilistic matrix factorization has been proposed in [15], where the performance is much better than SVD. A variety of techniques have been developed to upgrade the performance of PMF by considering the auxiliary information and introduce Bayesian and Generalized version of PMF [16]–[19].

Trust-aware collaborative filtering is proposed in [8] which utilizes CF to provide recommendation on the basis of trusted users with other trusted users, indoor to handle malicious users in the system which effect the accuracy of recommendation. Eigen taste algorithm is proposed which utilizes global queries to extract user ratings and apply principal component analysis PCA for making recommendations. Content-based filtering is another technique used for recommendation task, it recommends items on the basis of content (side information) of items.

In [20], authors proposed a method known as Meta-product2vec, which utilizes the side information by adding it to previously developed product2vec approach which utilizes synergy of local product information for generating distributed representation and neglect metadata information of item available only on training time in Meta-product2vec method which improves a recommendation performance. Content-based filtering PRES technique is proposed in [21] to recommend small home improvement articles. It basically makes recommendations by comparing user profiles with available document contents. However, they cannot handle data sparsity efficiently. Hybrid approaches have been developed by researchers with unite good features of both collaborative and content-based recommendation to overcome the limitations of CF and CBF [2].

Restricted Boltzmann Machine is used by [22] to find similarity between items and then map them into collaborative filtering. Considering the cross-domain matrix factorization coordinate system transfer method has been introduced in [23]. A cross-domain recommendation has been studied in [24], where no mutual user or item exists among cross domains. The proposed generative model for finding similar group between different domains. A multi-view deep neural network (MVDNN) model has been proposed to solve cold starch problem in [25], which maps user and item view in shared space. User features are obtained from browsing the history of users and recommend movies on the basis of maximum similarity with user.

HFT model is proposed in [12] which uses a topic modeling technique. The proposed model is the combination of latent dimension and latent review topics for interpreting rating dimension of either item or user reviews and results showed that technique outperforms as compared to other models that only used rating or review. Collaborative deionized auto encoder (CDAE) is proposed in [26] to handle/predict top-N recommendation by using user rating matrix, not considering side information such as reviews. A unified model proposed in [11] combines collaborative filtering and content-based filtering to solve cold starch problem. It uses a topic modeling techniques from user reviews to improve recommendation results.

LDA was proposed in [34], which is basically a topic modeling technique to capture latent topics and to address the limitation of word vector which are represented locally using w2v [36]. TWE was proposed in [35] to predict the context of words using topic and allow those words to have different word vector representations having different topics. Lda2vec is proposed by Moody in [33], which solves the limitation of previously mentioned techniques by considering both local and global meaning of documents. It integrates LDA and Word2vec to build context vector during training by utilizing both document vector and topic vector.

Recently, a deep learning technique has gained much importance in artificial intelligence, speech recognition and machine learning domain. These techniques are widely used for developing a recommender system using both collaborative and content-based filtering. A deep learning technique is used in [27] termed as Marginalized De-noising Auto-encoder (MDA), which integrates PMF and Margined Deionized auto-encoder to learn latent representation of items by neglecting randomly corrupted feature and reduces computational cost of training.

In [28], researchers used a deep learning approach called Hybrid-CF and content-based music recommendation which learn latent factors from music content using matrix factorization and apply deep learning techniques to regenerate them for better songs recommending. Deep Conn model is proposed in [29] which uses both user and item embedding as latent model of user and item and model interactions between them by applying factorization machines. It uses two neural networks and maps them into shared layer for making recommendations. The proposed model generates a vector representation of user and item content by utilizes side information along with rating data. It combines deep neural networks (CNN) along with topic modeling (Lda2vec) for the generation of user and item latent factors to improve recommender system accuracy.

III. PROPOSED METHODOLOGY

The proposed model is discussed in this section, which mainly consists of two steps. At first, the probabilistic matrix factorization is used to combine both topic modeling and deep learning technique. PMF unites Lda2vec and CNN for utilizing both rating and review information (user side review and item side review). Furthermore, CNN model was introduced which was used for generating item latent factor and LDa2vec model used for generating user latent factors. The overview of the proposed system is demonstrated in Fig. 1.

A. Probabilistic Matrix Factorization

As model integrates CNN and Lda2vec into PMF, based on which the PMF model explanation is considered first. Probabilistic matrix factorization (PMF) is a matrix factorization technique used to represent both user and items in shared d dimensional latent space where user and item latent vector can be represented as $u_p \in A_{ui}^d$ $i_q \in A_{ui}^d$. In order to predict whether user will like an item or not, dot product of both user and item latent features will be represented as $P_{ui} = (U_p^T \cdot I_q)$. Thus conditional distribution of predicted ratings is given as:

$$P(A_{ui}|U, I, \sigma^2) = \Pi_u \Pi_i N(P_{ui}|u_p^t \cdot i_q, \sigma^2) \quad (1)$$

Where $N(x|\mu, \sigma^2)$ is Gaussian normal distribution, probability density function with mean and variance. The proposed model

can minimize the regularized square error loss by using both user and item latent factors $U = (u_u)_{u=1}^p$ and $I = (i_i)_{i=1}^q$ as shown in equation.

$$\min_{u,i} \sum_{p,q} (A_{ui} - U_p^t \cdot I_q)^2 + \lambda_u \|U_p\|^2 + \lambda_v \|I_q\|^2 \quad (2)$$

Where λ_u and λ_v are the parameters used for regularization, actual rating prediction $A_{ui} > \text{ZERO}$ if user rated an item and it will be zero if user dose'nt rate item. To predict weather user rated an item or not equation is given below:

$$P_{ui} = (U_p^T \cdot I_q) \quad (3)$$

For user latent vector, the contextual topics from Lda2vec, user-contents (Y), and epsilon variable as zero-mean spherical Gaussian noise to optimize latent factor of user are considered.

$$U_p = \text{Lda2vec}(Y) + \epsilon_p \quad (4)$$

$$\epsilon_p \approx N(0, \lambda_u I_d) \quad (5)$$

Consequently, a conditional distribution for user latent factor is represented as:

$$P(U|Y, \lambda_u) = \Pi_u^p N(U_p | \text{Lda2vec}(Y), \lambda_u I_d) \quad (6)$$

Similar to user latent factor, for an item latent factor the weights of CNN (W) along with item-contents (X), epsilon variable and Gaussian noise is used and can be represented as:

$$I_q = \text{CNN}(W, X) + \epsilon_p \quad (7)$$

$$\epsilon_p \approx N(0, \lambda_v I_d) \quad (8)$$

Conditional distribution for item latent factor is represented as:

$$P(I|W, X, \lambda_v) = \Pi_i^q N(I_q | \text{CNN}(W, Y), \lambda_v I_d) \quad (9)$$

For both user-item latent factors, weight of CNN (W) along with user-item-contents document (Y, X) and Gaussian noise is used.

$$P_{ui} \approx N(U_p^T \cdot I_q, c_{pq}) \quad (10)$$

In (10) c_{pq} is a confidence parameter whose larger value represents more accurate rating. If $A_{ui} > 0$ i.e. user p rated an item q, and if $A_{ui} = 0$ i.e. user does not rated any item. The aforementioned details explained how PMF can deal with unknown ratings in the sparse rating matrix.

B. Convolutional Neural Network Model for Content Generation

The objective is to exploit contextual information for regularizing user and item latent factors in matrix factorization for this convolutional neural network (CNN) is utilized to generate content embedding of both users and items. Embedding Layer of CNN can convert unprocessed documents into meaningful numeral matrix based on the dimension/size of words. It is initialized with fast text [31] to generate content embedding of items and convert it into dense matrix which is used as an input document. Item are represented as a sequence of word vectors. Item-content with n words having s dimension can be represented in ID (input document) as:

$$ID = we_1 \oplus we_2 \oplus we_3 \oplus \dots \oplus we_n \quad (11)$$

where we is word embedding, \oplus shows concatenation operation to maintain order or words in input documents these techniques

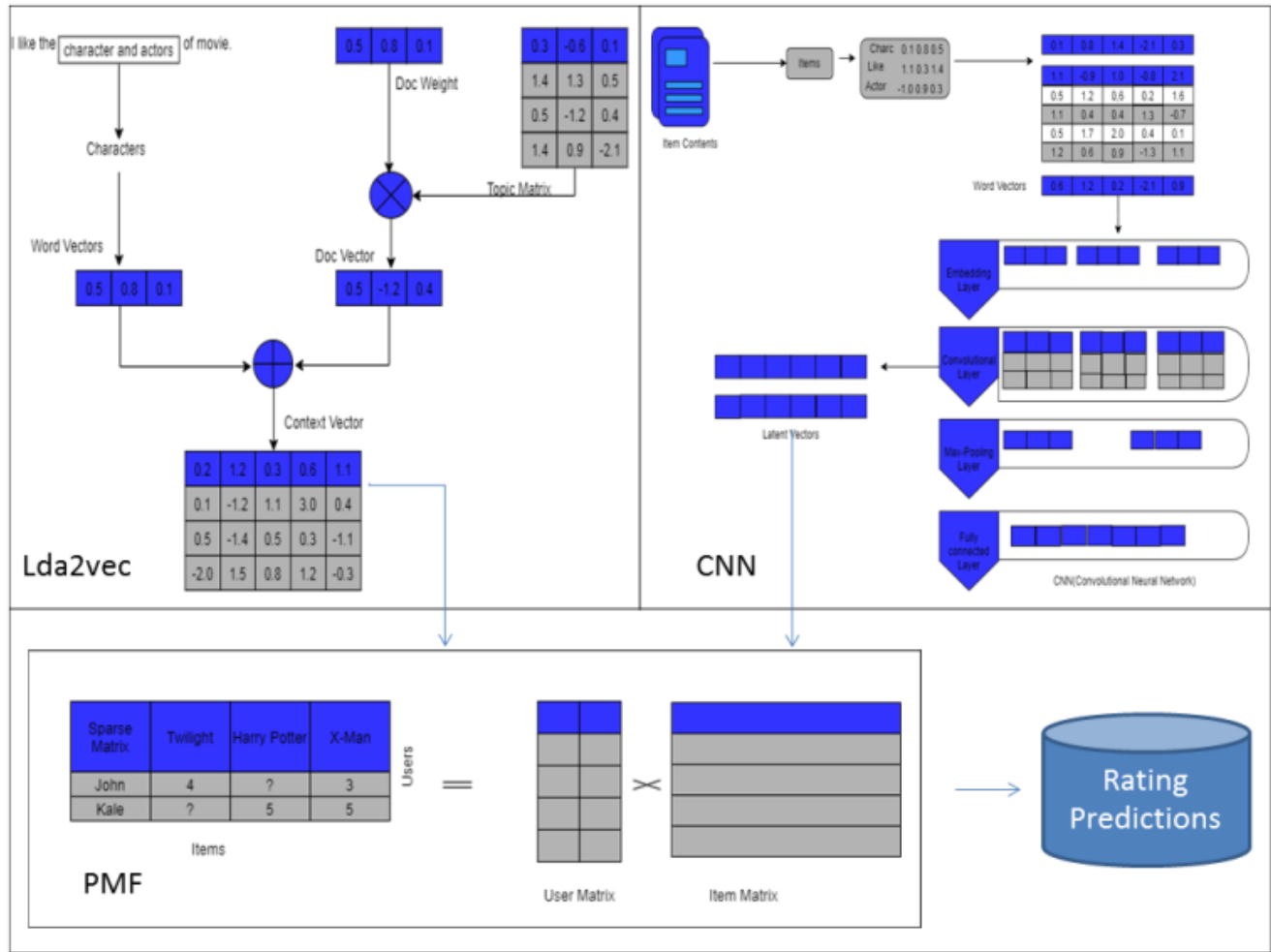


Fig. 1. Proposed Model Diagram

will solve the problems in traditional bags-of-words model. In convolutional Layer contextual information of words present in review document was captured. Various window sizes are used to get content from documents because convolutional kernel determines the number of surrounding words. It consists of m neurons to apply convolutional operation on word vectors of each document ID to get new features. Each neuron m in convolution kernel $L_m \in \mathbb{R}^{s \times t}$ is passed through different window size t and s dimensional word vectors represented in the equation as:

$$H_m = f(ID \otimes L_m + b_m) \quad (12)$$

where H_m represents the feature map in convolutional layer, \otimes represents convolutional operation, b_m represents bias which could be any real number, and f represents the activation function which could be sigmoid, ReLU and tanh. In this work, ReLU is taken as activation function to dodge vanishing gradient problem then (12) will become:

$$H_m = ReLU(ID \otimes L_m + b_m) \quad (13)$$

A *Max - Pooling Operation* is performed on specific kernel H_m to obtain maximum features from convolutional layer. Each feature mapped which has the highest value can be

captured and considered as most important features for rating prediction. It ignores extra contextual features in the convolutional layer and handles the dynamic length of document by providing new fixed-length feature vectors which will improve the performance of system as shown in (14). Only one feature is extracted from one kernel so different filters of various window sizes have been used to get multiple features and the output vector is represented as (16).

$$O_m = \max[H_1, H_2, H_3, \dots, H_{n-t+1}] \quad (14)$$

$$O = [o_1, o_2, o_3 \dots, o_r] \quad (15)$$

where r represent the number of convolutional kernels.

In *Output Layer*, top features from the max-pooling layer ought to be converted in require assignment. These high-level feature O are passed to d - dimensional vector space of user and item by using non-linear projection:

$$E = \tanh(M_2 \times \tanh(M_1 \times +b_1) + b_2) \quad (16)$$

Where M_1 and M_2 are the projection matrix, b_1 and b_2 are bias functions. The aforementioned process can change crude

documents as an input document and return latent vectors of d-dimension as output.

$$I_q = CNN(W, X) \quad (17)$$

C. LDA2VEC for Topic Extracting

Lda2vec is the combination of word2vec model (Natural Language Modeling tool) with most commonly used topic modeling technique LDA (Latent Dirichlet Allocation) to utilize best features from both [33]. Lda2vec model combines “globality” and “locality” idea of word prediction from word2vec and LDA. Lda2vec summed up word embedding with LDA vectors and compare them with latent word topics. Finally, the conditional probability model is applied to predict final topics. Thus model predicts words from not only the given context but also from the probability of coexistence of other words. For local word prediction, (18) is utilized.

$$P = (P_{tar}|P_{piv}) \quad (18)$$

Where P_{tar} is target word and P_{piv} is pivot word probability respectively. While for global word prediction (19) is used.

$$P = (P_{tar}|P_{topic}) \quad (19)$$

Where P_{topic} is topic probability, for which Lda2vec is used that predict both locally and globally as:

$$P = (P_{tar}|P_{piv} + P_{topic} + P_{document}) \quad (20)$$

Where $P_{document}$ is a sparse LDA vector, P_{topic} is contextual topics, P_{piv} is pivot word. Procedure describes above uses Lda2vec model as a function that takes inputs from user-content and returns user latent vector. An obtained latent vector is supplemented with semantics content enriched with topic contents. So the resultant user latent vector can be represented as:

$$U_p = Lda2vec(Topic, Y) \quad (21)$$

D. Hybrid Model of CNN and Topic Modeling

In a proposed model, a description document of both users and items (which have reviews written by user to specific items) is utilized. User contents provide information regarding users interests whether they like or dislike a particular product while item contents provided information about item properties. The proposed model is termed as Hybrid model of convolutional neural network [30] and topic modeling for recommender system as it combines both mentioned techniques into probabilistic matrix factorization to acquire latent factors of both user and item so that a better recommendation can be provided. It concurrently learns the dense word vectors using Dirichlet distributed topic mixture [33]. Union of CNN (deep learning technique) + Lda2vec (topic modeling technique) allows system to get more global features and provides better understanding of the review document. CNN is used to capture contextual information and Lda2vec is used to capture topic information available in review text. These extracted features along with topic details provide good understanding of memory and model-based collaborative filtering. To provide more efficient and accurate recommendations model further utilizes PMF which outperforms on sparse, imbalance and large datasets. Experimental results show the effectiveness of the above proposed model [15].

Lda2vec captures topics both locally and globally, which are further combined with deep learning techniques to improve rating predictions. After getting latent factors of both users and item model initialize PMF which analyze reviews and rating to provide better recommendation to users.

IV. PERFORMANCE EVALUATION

Algorithm proposed in this research is evaluated and compared with already proposed state of the art model. Model exercised a five-fold cross-validation test to check the accuracy of rating prediction.

A. Dataset Preprocessing

In this paper, Amazon dataset¹ is used which has 22 further sub-categories of products. It consists of reviews and product meta-data. In this experiment, four different datasets of Amazon instant video, auto motives, health and personal care and Kindle store (AIV, Auto, HPC, KS) are used to evaluate the performance of proposed model. It has 142.8 million reviews covering May 1996 - July 2014. K-core dataset is used in these experiments, which means each user and item has K reviews. Table I below show the statistic of dataset.

TABLE I. THE STATISTICS OF AMAZON DATASETS

Datasets	No.of Users	No.of Items	No.of Ratings	No.of Reviews	Scale
AIV	5130	1685	37126	37126	[1 - 5]
KS	68223	61934	982618	982618	[1 - 5]
Auto	2928	1835	20473	20473	[1 - 5]
HPC	38609	18534	346355	346355	[1 - 5]

For evaluating performance of the proposed model, the dataset splits into three categories (i.e. training, validation, and testing sets) using ratio of 80%, 10%, and 10%. As it requires reviews of both users and items rating predictions. Real dataset contains reviews of both (user and item) in single item. First user-content and item-content are separated, then every user and item is represented in a sequence of reviews. Furthermore, the data is pre-processed by removing stop words, tf-idf vectorization is used to remove words from document whose frequency is 0.5, fixing the vocabulary size to 8000, setting maximum length of document 1200, and introduce two thresholds $tc = 0.5$ and $tr = 0.8$ to control the number of sequence for each user/item and to control the length of the sentences in each user/item. Items having min-rating is 1 removed along with all other items who have no ratings to get more accurate and precise recommendations. This pre-processed information is then passed to fast text [31] word embedding technique to produce vector representation of users and items. For CNN fast text model is initialized with built-in sentences with min-count = 1, dimension size $d = 200$. Various window size [3-5] is used in convolutional layer of CNN model with dropout ratio to avoid over fitting problem. The number of epochs is fixed to 1 and batch size of 128, 256, and 512. Embedding layer is initialized with these word vectors to get contextual features items. For Lda2vec window size = 1 is used to collect every word surrounds the pivot words. To clean and tokenized user-content the script is removed, preprocess and corpus functions of Lda2vec is applied as pre-trained model.

¹<http://jmcauley.ucsd.edu/data/amazon/>

After getting contextual information in the form of user and item latent vectors PMF is initialized with these latent vectors which play a vital role in improving prediction accuracy.

B. Evaluation Matrices

Different types of evaluation techniques have been used in literature to check the performance of recommender system [32]. Evaluation matrices are divided into three categories:

- 1) Prediction matrices (MAE, RMSE)
- 2) Classification matrices (RECALL, PRECISION)
- 3) Rank Measure (Measuring ordering of items performed by recommender system).

In experiments RMSE is used for rating predictions and can be calculated as:

$$RMSE = \sqrt{\frac{1}{N} \sum_{u,i} (A_{ui} - P_{ui})^2} \quad (22)$$

Where; A_{ui} is actual ratings, P_{ui} is predicted ratings, N represent total number of ratings, u represent user, i represent item. Minimum value of RMSE represent better rating prediction.

Along with RMSE, Precision and Recall metrics are also used as evaluation matrices to measure the quality of recommendation. Two classes namely relevant (have rating ≥ 3) and irrelevant (have rating < 3) are defined to recommend items to users.

$$Precision = \frac{truepositive}{truepositive + falsepositive} \quad (23)$$

$$Recall = \frac{truepositive}{truepositive + falsenegative} \quad (24)$$

Where; *truepositive* means recommend relevant item to user, *falsepositive* means recommend irrelevant item to user, *falsenegative* means relevant items are not recommended to user.

C. Results and Discussions

Experimental environment of this research is Ubuntu platform, with core i7-7700 CPU, 2 GPU, 16 GB RAM, 2 TB Disk. All implementation is done in python for CNN implementation which uses Keras framework with tensor flow at back end. Extensive set of experiments is performed on the Amazon dataset to study the impact of two different parameters latent factors and convolutional kernels. In Fig. 2 and Fig. 3 latent factors varies from 10 to 50 and convolutional kernels vary from 50 to 200 to check sensitivity of the proposed model against those parameters. Fig. 2 and Fig. 3 shows a decreased RMSE which improves rating prediction accuracy.

For top-N recommendation, the precision and recall matrices have been computed, which divides data into two classes as relevant and irrelevant classes. Rating above three or equals to three are considered as relevant items and all remaining items who have ratings less than this threshold value are considered as irrelevant items. Fig. 4 shows the performance of proposed model in terms of top-N recommendation.

The performance of proposed model is evaluated by using different parameter settings and results are shown in Table II which displays RMSE of all datasets and overall performance of the proposed model with other models PMF [15], ConvMF

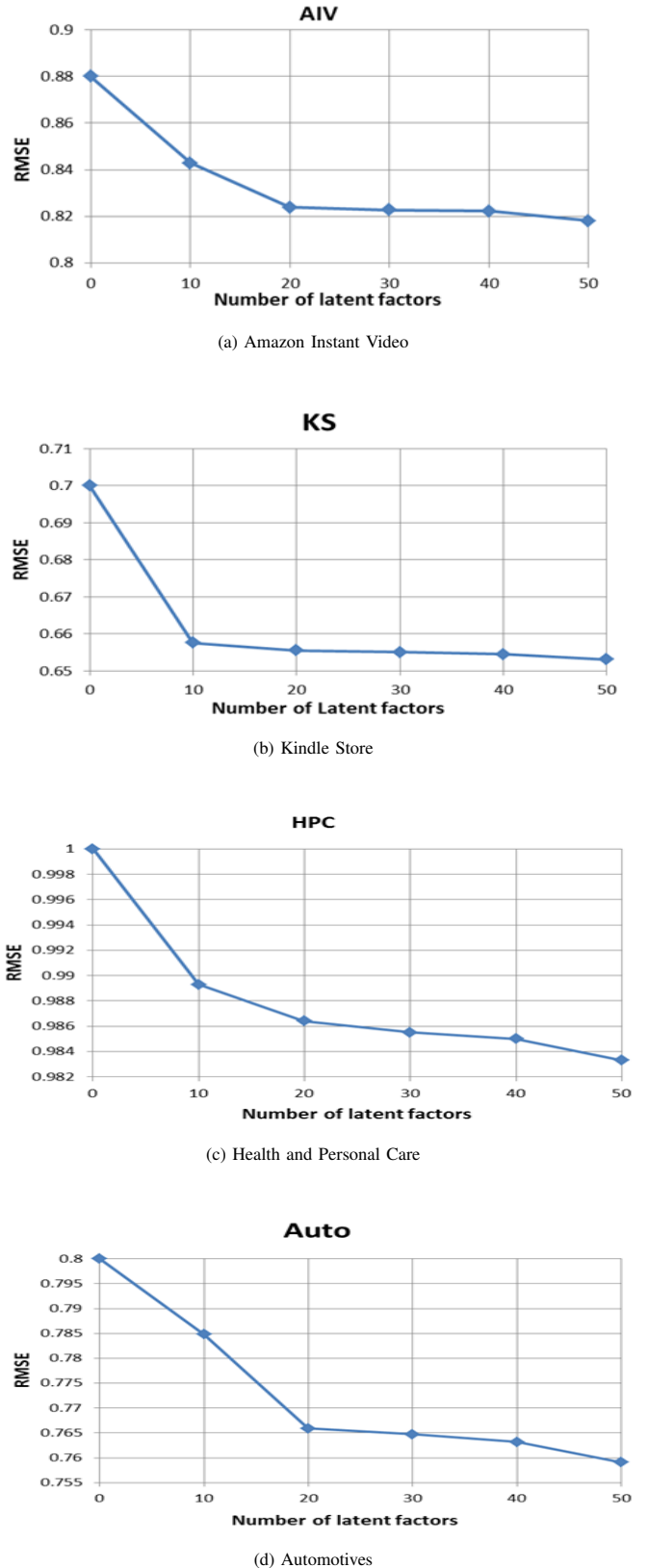
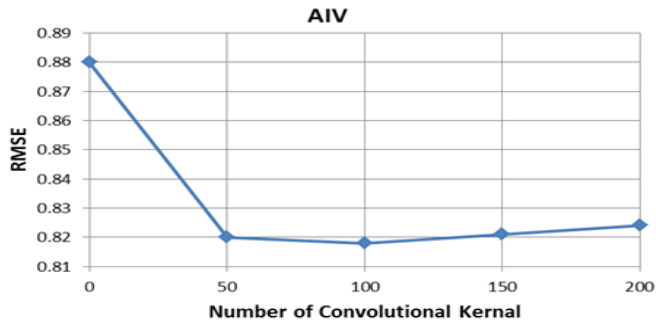
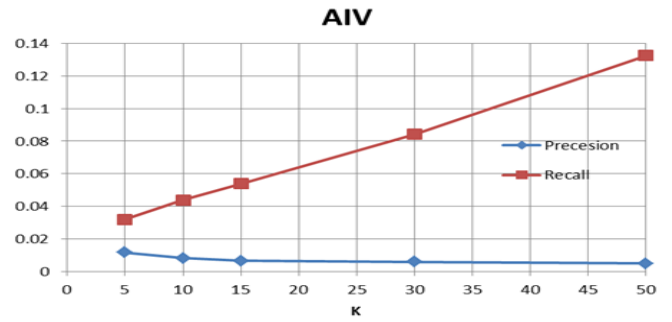


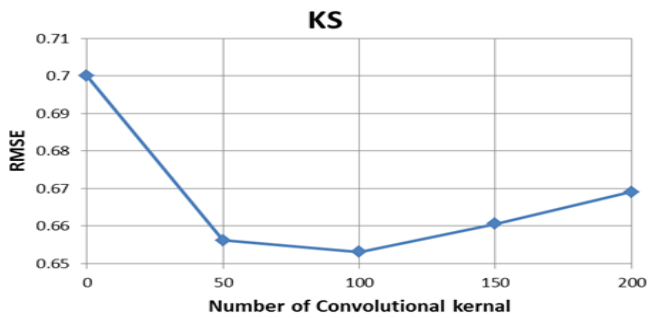
Fig. 2. Performance of Proposed Model on Datasets with different Latent Factors (D)



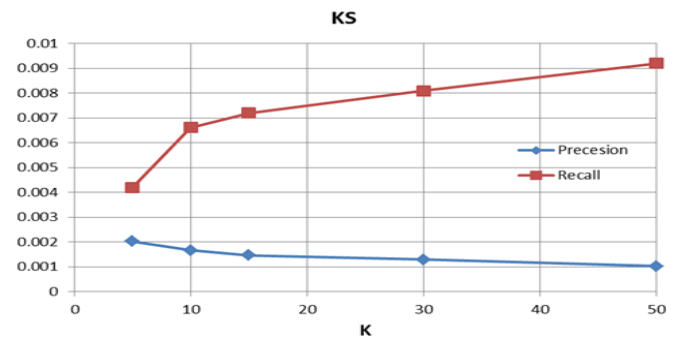
(a) Amazon Instant Video



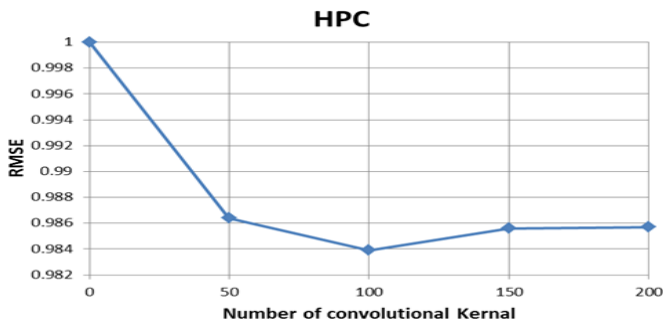
(a) Amazon Instant Video



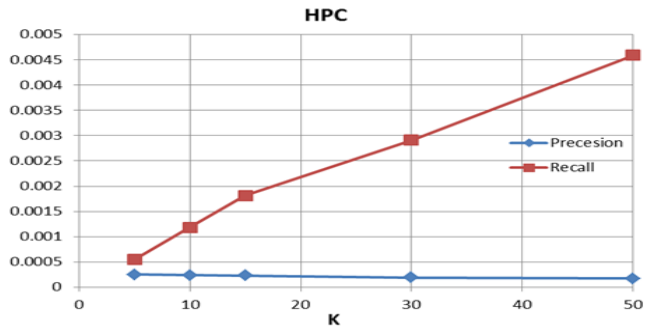
(b) Kindle Store



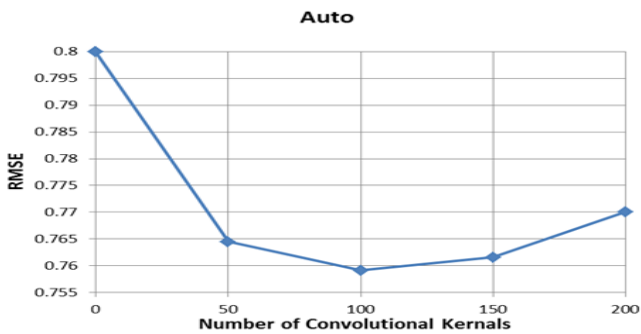
(b) Kindle Store



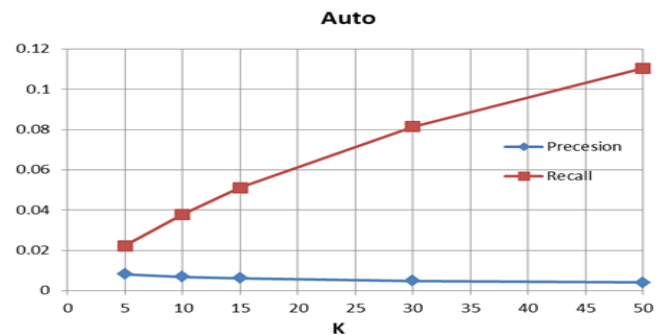
(c) Health and Personal Care



(c) Health and Personal Care



(d) Automotives



(d) Automotives

Fig. 3. Performance of Proposed Model on Datasets with different Convolutional Kernels.

Fig. 4. Precision and Recall

[30] and DeepConn [29]. RMSE of PMF in aiv dataset \approx 1.2098, ConvMF \approx 1.0159(aiv) and in DeepConn \approx 1.0122. This shows that ConvMF gains better performance than PMF it means that by considering side information recommendation accuracy has been improved. PMF uses only rating data. ConvMF used both ratings data along with side information of items but they neglect side information of user side which plays a vital role in rating prediction. Moreover, DeepConn utilizes side information of both user and item contents which bring better results and improve recommendation accuracy.

In Table II result shows that DeepConn-Item, DeepConn-User and DeepConn-Both outperform ConvMF in terms of RMSE. Results depict that taking advantage of both user and item content embedding give more efficient and stronger recommender systems. This might be substantiated by comparing HCCTM-Both with HCCTM-User, HCCTM-Items and in contrast with DeepConn-Items, DeepConn-Users and DeepConn-Both, ConvMF and PMF. HCCTM-Both improves HCCTM-User and HCCTM-Item on AIV dataset by 2.45% and 16% respectively. Similarly, HCCTM-Both outperform DeepConn-Both, ConvMF and PMF by 19.42%, 28.79%, 39.18% on AIV dataset respectively. Similar trend can be seen in remaining datasets.

Proposed paradigm considers information of both user and items content to improve the accuracy of ratings predicting. Finally, evidence shows that proposed model outperform ConvMF, PMF and DeepConn. RMSE has improved significantly which depicts that by using the topic information along with neural networks, provide a more accurate and effective recommender system.

TABLE II. PERFORMANCE ANALYSIS OF OUR PROPOSED MODEL WITH OTHER MODELS

Dataset \rightarrow Models \downarrow	AIV RMSE	KS RMSE	HPC RMSE	Automotive RMSE
PMF	1.2098	0.9657	1.3348	1.1124
ConvMF	1.0159	0.9374	1.2308	0.9178
DeepCoNN – Items	1.0142	0.9184	1.2183	0.9134
DeepCoNN – Users	1.0144	0.9187	1.2185	0.9137
DeepCoNN – Both	1.0122	0.9046	1.2036	0.9113
HCCTM – Items	0.9779	0.8015	1.0892	0.9026
HCCTM – User	0.8425	0.7546	1.0318	0.8234
HCCTM – Both	0.8180	0.6531	0.9833	0.7591

V. CONCLUSION AND FUTURE WORK

In this paper, a hybrid model of convolutional neural network and topic modeling for a recommender system that incorporates CNN + Lda2vec into probabilistic matrix factorization to seize the content information of both items and users is presented. The user and item contents are exploited for rating predictions. The model learned a latent factor of user and item from both ratings and reviews of user. A collaborative filtering technique PMF is used for rating prediction. The suggested model is applicable to other datasets having both user and item contents. The experimental finding shows that the proposed model performs much better when compared with other state of the art model (i.e. PMF and ConvMF). The aforementioned technique can effectively learn latent factors for both user and item, thus provides high accuracy and better performance. But briefly reiterated the facts that there may be room for more

improvement so for further research, examination and incorporation of time factor to generate user and item latent will be evaluated. Also, it would be interesting to eliminate noise problems in the recommender system as it will affect the performance of the recommender system.

VI. ACKNOWLEDGMENT

The authors would like to acknowledge ORIC Khwaja Fared University of Engineering and Information Technology (KFUEIT), College of Computer Science Zhejiang University, China and Military College of Signals National University of Science and Technology (NUST) for supporting this work.

REFERENCES

- [1] Y. Bao, H. Fang, and J. Zhang. Topicmf: Simultaneously exploiting ratings and reviews for recommendation. AAAI, pages 28. AAAI Press, 2014.
- [2] F.O. Isinkaye a,*, Y.O. Folajimi b, B.A. Ojokoh. Recommendation systems: Principles, methods and evaluation. Egyptian Informatics Journal, (2015) 16, 261273.
- [3] S. Bharadwaj, T. I. Dhamecha, M. Vatsa and R. Singh, Computationally efficient face spoofing detection with motion magnification, IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 105–110, 2013.
- [4] G. Krell, M. Glodek, A. Panning, I. Siegert, B. Michaelis, A. Wendemuth and F. Schwenker, Fusion of fragmentary classifier decisions for affective state recognition, International Conference on Multimodal Pattern Recognition of Social Signals in Human-Computer-Interaction, vol. 7742, pp. 116–130, 2012.
- [5] AM. Acilar, A. Arslan. A collaborative filtering method based on Artificial Immune Network. Exp Syst Appl, 2009;36(4):832432.
- [6] Aggarwal, Charu C. Neighborhood-Based Collaborative Filtering Recommender Systems: The Textbook, Springer International Publishing Switzerland 2016.
- [7] Kai Yu, Anton Schwaighofer, Volker Tresp, Xiaowei Xu, Hans-Peter Kriegel. Probabilistic Memory-Based Collaborative Filtering. IEEE Transactions on Knowledge and Data Engineering, 21 July 2013.
- [8] P. Massa, Paolo Avesani. Trust-aware Collaborative Filtering for Recommender Systems Via Sommarive 14 - I-38050 Povo (TN) - Italy.
- [9] X. Su and T. M. Khoshgoftaar. A Survey of Collaborative Filtering Techniques. Advances in Artificial Intelligence, Volume 2009, Article ID 421425, 19 pages.
- [10] J.S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, 1998.
- [11] Guang Ling, Michael R. Lyu and Irwin King, Ratings Meet Reviews, a Combined Approach to Recommendation, Computer, October 610, 2014.
- [12] J. McAuley and J. Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In Proceedings of the 7th ACM conference on Recommender systems, pages 165172. ACM, 2013.
- [13] Robert M Bell and Yehuda Koren. Improved neighborhood-based collaborative filtering. In KDD13 CUP, 2007.
- [14] G. Guo, J. Zhang, D. Thalmann, Merging trust in collaborative filtering to alleviate data sparsity and cold start, Knowl.-Based Syst. 57 (2014) 5768.
- [15] A. Mnih, R.R. Salakhutdinov, Probabilistic matrix factorization, in Adv. Neural Information Processing Systems, 2008, pp. 12571264.
- [16] Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In ICML, pages 880887, 2008.
- [17] Ryan Prescott Adams, George E. Dahl, and Iain Murray. Incorporating side information in probabilistic matrix factorization with gaussian processes. In UAI, pages 19, 2010.
- [18] H. Shan, A. Banerjee, Generalized probabilistic matrix factorizations for collaborative filtering, 2010 IEEE 10th International Conference on Data Mining (ICDM) (2010) 10251030.

- [19] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature, vol. 521, no. 7553, pp. 436444, 2015.
- [20] Robin van Meteren and Maarten van Someren, Using Content-Based Filtering for Recommendation. NetlinQ Group, Gerard Brandtstraat 26-28, 1054 JK, Amsterdam.
- [21] Flavian Vasile, Elena Smirnova, Alexis Conneau, Meta-Prod2Vec: Product Embeddings Using Side-Information for Recommendation. September, 2016. In RecSys '16 Proceedings of the 10th ACM Conference on Recommender Systems, Pages 225-232.
- [22] Asela Gunawardana and Christopher Meek. Tied boltzmann machines for cold start recommendations. In RECSYS08, pages 1926.
- [23] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. Transfer learning in collaborative filtering for sparsity reduction. In AAAI10, 2010.
- [24] Bin Li, Qiang Yang, and Xiangyang Xue. Transfer learning for collaborative filtering via a rating-matrix generative model. In ICML09, pages 61762.
- [25] A. M. Elkahky, Y. Song, and X. He. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In Proceedings of the 24th International Conference on World Wide Web, pages 278288. International World Wide Web Conferences Steering Committee, 2015.
- [26] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester. Collaborative denoising auto-encoders for top-n recommender systems.
- [27] S. Li, J. Kawale, and Y. Fu. Deep collaborative filtering via marginalized denoising auto-encoder. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pages 811820. ACM, 2015.
- [28] X. Wang and Y. Wang. Improving content-based and hybrid music recommendation using deep learning. In Proceedings of the ACM International Conference on Multimedia, pages 627636. ACM, 2014.
- [29] Lei Zheng, Vahid Noroozi and Philip S. Yu, Joint deep modeling of users and items using reviews for recommendation. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM17) (pp. 425434). New York, NY, USA: ACM.
- [30] D. Kim. Convolutional Matrix Factorization for Document Context-Aware Recommendation ,978-1-4503-4035-9 (2016).
- [31] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov .Bag of Tricks for Efficient Text Classification, Facebook AI Research, 2016.
- [32] A. Davoudi M. Chatterjee. Modeling trust for rating prediction in recommender systems. In SIAM Workshop on Machine Learning Methods for Recommender Systems, SIAM, pages 18, 2016.
- [33] C. Moody.: 2016. Mixing Dirichlet Topic Models and Word Embeddings to Make lda2vec, arXiv:1605.02019v1 [cs.CL] 6 May 2016.
- [34] Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent Dirichlet Allocation. J. Mach. Learn. Res. 2003, 3, 9931022.
- [35] Liu, Y.; Liu, Z.; Chua, T.; Sun, M. Topical Word Embeddings. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 2529 January 2015. Available online: <https://pdfs.semanticscholar.org/9a0a/f9e48aad89512ce3e24b6a1853ed3d5d9142.pdf> (accessed on 4 September 2018).
- [36] Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. Available online: <https://arxiv.org/pdf/1301.3781.pdf> (accessed on 4 September 2008).