

Hybrid Recommender System Leveraging Stacked Convolutional Networks

Naresh Nelaturi* and G. Lavanya Devi

Dpt of Computer Science and Systems Engineering, College of Engineering, Andhra University, Visakhapatnam 530 003, Andhra Pradesh, India

Received 24 February 2018; Accepted 5 June 2018

Abstract

Recommendation Systems has emerged as an essential component in web-based systems, as their ability to analyze customers' behavior and generate recommendations seeking customers' satisfaction is successfully accomplished. However, the success of these systems depends on amount of customers' personal preference data and content (items) metadata available for harnessing. Therefore, data sparsity poses a major challenge here. To alleviate this problem, data and models from other domains can be leveraged to gain good insight about customers' preferences and content similarities. In specific, this paper proposes the idea of extracting knowledge for transfer learning leveraging pre-trained deep neural networks. Knowledge from pre-trained models is used to efficiently identify similarity and capture customers' preference among the contents. To attain the objective, this paper presented an approach, for generating efficient top-n recommendations using a hybrid recommender model. Performance analysis is performed on the proposed approach and results obtained are promising. Furthermore, extensions for this work are also discussed

Keywords: Recommender, Convolutional neural networks, item based filtering

1. Introduction

With the dawn of the web 2.0 era, the World Wide Web has grown enormously facilitating large repositories of information. These advances indeed created an opportunity for development of ecommerce, social media networking and many other information management systems. A decade later as the web users increased extensively, all the repositories are abundantly loaded with wide range of information. This posed a great challenge for the web users to effectively retrieve the information from the repositories. Lack of software systems to handle information overload made the web users to experience difficulties in interaction and forced to spend ample time in searching. There is an increasing concern over need of a mechanism to aid web user in making decisions to retrieve information efficiently. This motivated to the development of system software to assist users in their information needs known as Recommendation system. Broadly speaking, main objective of a recommender system is assisting users to combat information overload by generating personalized suggestions. This systems application achieved tremendous success in a wide range of domains across the web such as movies, music, news articles, advertisements, books and retailer products. In the last decade, along with industry it has also attracted interest of research and academic communities. As a result, numerous technologies and algorithms are developed to improve their performance.

Recommendation system employs several computational techniques to perform its major tasks such as rating prediction and ranking items. Recommendation techniques can be classified into content based filtering, collaborative

filtering and hybrid. These techniques may use features, attributes of customer or items; similarly, they may also consume customer-item interaction data in the form a rating matrix. The later one that uses only rating matrix for filtering is called collaborative filtering. In contrast, content based filtering harness metadata of items along with rating matrix. In the Rating matrix each row represents a customer 'c' and each column represents an item 'i' available in company's repository. Conceptually, the matrix represents a customer's interactions and preference on items. Interactions refer to each customer purchase or visited history on items. Preferences are collected either from explicit feedback or implicit feedback. In content based filtering methodology, *content* indicating the descriptive information of the items and customers are utilized by prediction models for rating and subsequently to rank the items. Both collaborative and content based filtering has been state of art models for first generation recommendation system. However, traditional recommender approaches do have limitations such as sparsity, scalability, cold start and long tail. To improve the quality of the recommendations, these traditional techniques are combined to form a unified recommendation system referred as hybrid approaches. A considerable amount of literature has been published in developing various learning models for traditional and hybrid recommendation system. In the domains such as electronic commerce, goal of recommender system is to increase the items sales and generate profitable revenues. Therefore, Recommender systems are supposed to grab the attention of the customers by generating recommendations having relevance, diversity, novelty and serendipity.

In the recent years, evolution of deep learning caused the world to witness a potential drift in processing extremely complex tasks. Their application to recommendation system is raising new aspirations for researcher and practitioners of industry in improving recommendation performance. In the

*E-mail address: naresh855@gmail.com

ISSN: 1791-2377 © 2018 Eastern Macedonia and Thrace Institute of Technology. All rights reserved.

doi:10.25103/jestr.113.12

last few years, deep learning based recommender model produced remarkable performance for YouTube, Google play, Yahoo news and Netflix. Although, a few researchers carried out systematic research in this area, extensive research has not been carried to unleash full potential of deep learning applications in recommendation systems. On the other hand, field of machine vision achieved state of art results in visual interpretations with convolutional neural network (CNN) technique of deep learning family. Despite of this, very few studies have investigated the application of CNN in recommendation systems. The aim of this paper is to provide an approach to generate recommendations by harnessing the metadata (images) with the CNNs, as most of the domains provide image of items. With insights from visual recognition techniques, this work examines models to identify the patterns in the image of items category. These patterns can provide information to have a deeper understanding of customer's preferences and subsequently can aid the recommender model to generate relevant recommendations. A variety of CNN architectures are available in literature for the task of extracting features from images. However, initialization of hyper-parameters and learning of parameters for these CNN models is a subtle task involving larger training period and even may result in inefficient performance. Furthermore, information extracted must be transformed properly to integrate with preference data. Consequently, this paper attempts to show how to leverage pre-trained CNN models to extract features for item images. This further reduces the training period of the model and number of parameters to be learned. Stacking the CNN network with a fully connected layers can generate information that can be integrated with other information with ease. So far numerous filtering paradigms are developed in various domains for generating recommendations. However, less attention has been paid regarding the context that repository having more homogenous items belonging to various categories. To address this issue the proposed approach utilizes item based filtering as it can capture customer preferences by exploiting axillary information about items. In contrast to the other popular methods, item based filtering offers real time performance and scalability.

This paper contributes to existing knowledge of designing recommender models by providing following contributions through conceptualizing deep learning techniques for recommendation task:

- A novel recommendation algorithm that generates Top-n recommendations to customer by learning his preferences through information extracted from the purchased items images using CNN.
- The output of the CNN model providing information about unexpressed customer preferences has been seamlessly integrated with interaction data (rating matrix) in the proposed item-based recommendation algorithm.
- Experimental evaluation is performed by using crawled data from the source *The Movie database*. Results show that proposed approach produced promising performance.

The paper is organized as follows: Section II describes the background theory, concepts of deep learning and its applications in research of recommendation systems. Section III presents a framework for integrating convolutional neural network with a hybrid recommender. Section IV discusses

the experimentation and evaluation of the proposed approach. Finally, Section V summarizes the work and gives an outlook on future directions.

2. Background Theory and Related Work

2.1. Deep Learning

Deep learning successfully turned to be a new terrain for machines, to attain cognitive abilities such as vision and natural language processing by performing complex and difficult tasks. The most admired models of deep learning family are *Convolution neural networks* (CNNs), *Recurrent neural networks* (RNNs), *Deep belief nets* (DBNs) and *Auto encoders*. These models produced state-of-the-art performance in fields of machine vision and natural language processing (NLP) [1, 2, 3, 4]. Convolution Neural Networks (CNNs) evolved as a powerful and prominent artificial neural network algorithm in the field of machine vision for the tasks of object, action recognition. CNNs are unorthodox neural networks (converse to fully connect) that can process and preserve spatial information in the data for performing desired tasks. In theory, CNNs architecture is biologically inspired from human visual cortex layered processing. In humans, visual perception is done through multiple visual cortex levels. Initially, simple single points (pixels) are recognized and from them geometric structures are identified. And then, from these structures known elements (real world things like humans, electronic devise, and animals) are pursued by the cortex [5, 6]. This is reason for adapting CNNs to perform complex tasks of machine vision.

Computationally, CNNs perform *convolutional* operation in some of the layers of network instead of conventional matrix operation. This reduces the overhead of learning all parameters of the network as in the case of fully connected network. During training the CNNs attains information about object's spatial features irrespective of it position in the image, and therefore they have the ability to recognize object in varying spatial positions in an image. Essential elements of CNNs are convolution layer, pooling layer and fully connected layer. Three distinct layers are arranged in hierarchical to form a full CNN architecture. First layer, in any CNN architecture is the convolution layer. This layer takes images represented in the form of matrix of pixels, and produces an output known as activation map (feature map) for each filter considered. Each neuron in the convolutional layer acts as a filter. First convolution layer extract low-level image features (edges) while, the subsequent convolution layers extract complex features/structures (objects). Pooling layer is placed after successive convolutional layers in the CNN architecture to consolidate the feature maps resulted from the convolutional layer. They also produce a feature map that represents semantics of a particular region in the image. Fully connected layers are the final layers in CNN architecture, it represents a regular neural network that has connections to every node in the previous layer. This layer produces a feature vector, where every element in the vector represents a class (label) corresponding to high level features summarized by the convolutional and pooling layers [8, 9].

Competitions and challenges influenced and contributed to the development of state of art models for visual recognition. ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is front runner in organizing such challenges, and also a benchmark in large scale object detection and classification. It has been instrumental since 2010, seeking attention of researchers and practitioners

worldwide to the area of machine/computer vision. The other notable challenges in this area are MNIST, CIFAR, COCO [10, 11, 12]. The most popular architectures resulted from

these competitions that grabbed attention of researchers and industry practitioners are outlined in Table 1.

Table 1. Pre-Trained CNN Architectures Table Type Styles

Challenge	CNN Architecture	Contributor	Description/Summary
ILSVRC 2010	LeNet	Yann LeCun et al. [13]	This shallow network was trained to identify hand written digits. The network was made up of 2 conv layers and 2 fully connected layers, having filters of size 5x5 with stride 1. Number of parameters to be trained 60,000.
ILSVRC 2012	AlexNet	Alex Krizhevsky et al. [14]	Model was trained on 15 million images with 1,000 categories. This network comprises of a total of 8 layers. 5 convolutional layers and 3 fully connected layers each with 4096 units. It is deeper than LeNet with more filters per layer, number of parameters to be learned are around 60 Million
ILSVRC 2013	ZF Net	Zeiler& Fergus [15]	Model is a 8 layered network trained on 1.3 million images. Layers are stacked in similar way as AlexNet, major difference being the size of the filters and number of filters employed in each layer.
ILSVRC 2014	GoogLeNet	Christian Szegedy et al. [16]	Model was trained on 15 million images with 1,000 categories. The network was made up of 22 layers .Number of parameters to be learned are around 4 Million, as it employs Average pooling at the top layers..
ILSVRC 2014	VGGNet	Karen Simonyan and Andrew Zisserman [17]	Model was trained on 15 million images with 1,000 categories. The network was made up of 19 layers - 5 conv layers, max-pooling layers, dropout layers, and 3 fully connected layers. Number of parameters to be learned is around 140 Million, as it employs Average pooling at the top.
ILSVRC 2015	ResNet. Residual Network	Kaiming He et al. [18]	Model was trained on 15 million images with 1,000 categories. The network was made up of 152 layers .

2.2. Recommendations without and with Deep Learning

Traditional Recommendation systems are adopting the paradigms of content based filtering, collaborative filtering or a combination of both. A considerable amount of literature can be obtained on memory based models for collaborative filtering [19, 20]. Neighborhood methods have been state of art for collaborative filtering paradigm, produced numerous algorithms utilizing wide range of similarity metrics [21, 22]. These memory based models an avatar of instance based learning approach are used to develop model for the task of rating prediction on demand for a specific customer. Data sparsity poses a major challenge for memory based approach, to alleviate this dimensionality reduction techniques are used [23, 24]. In general, collaborative filtering approach can be seen as generalized classification problem. Therefore, number of classification and regression models can be utilized for solving this paradigm recommender problem. Such an approach is called model based collaborative filtering. Many classification techniques such as bayesian, rule based and decision tree are combined with collaborative filtering techniques produced convincing performances [25, 26, 27]. Another prominent approach for collaborative filtering that produced good experimental results is latent factor models. Considerable amount of literature is available on deterministic/probabilistic based matrix factorization techniques for recommender systems [28]. Enhancements of these systems, with exploitation of the supplementary side information are not carried out extensively.

Deploying deep learning techniques for this exploitation can prompt new approaches of leveraging deep learning for recommendations. Trends of this new paradigm are observed in some recent publications. Jingyuan chen et al. proposed a model for multimedia recommendation based collaborative filtering augmented with Multi layered perceptron network [29]. Paul covington, Adams, and Emresargin has developed

deep candidate generation model for generating YouTube recommendation using a multi layered network [30]. Even factorization methods are mixed with feature engineering using deep learning to build as Deep Factorization Matrix to challenge most successful wide and deep models of deep learning [31]. Another prominent model of deep learning family, *Restricted Boltzmann Machine* [RBM] is also used in various applications such as recommendations for android background services and music [32, 33]. RBMs produced superior performance as they effectively utilize factorized representations and same is evident in Netflix Prize contest. Similarly, recurrent neural networks notoriously known for modelling sequence and time series data are augmented for the tasks of recommender systems [34, 35]. Shumpei okura et al. proposed an approach combining auto encoder for article representation and recurrent neural network for referring user purchase behavior [36]. CNNs which created increasing interest in researchers from the domain of visual representation are also successfully applied in document, resource and e-learning recommender systems [37, 38, 39, 40]. Several attempts have been made to incorporate Auto encoders embedded with deep learning techniques for the tasks of recommendations [41, 42, 43].

Inspired by the deep insights gained by the deep learning techniques, a combination of deep learning techniques and traditional recommender systems was considered for the study. In particular this paper seeks to investigate the development of a unified framework for ecommerce, integrating CNN technique with traditional recommendation technique. The key research questions are to study the techniques to capture the customer’s preferences from the images of the items he purchased in the past, and to incorporate this information to recommender model

3. Approach

In this section, we discuss the approach adopted to construct a framework for generating top-n recommendations in an e-commerce domain, where repository has set of items with images. The framework attempts to integrate traditional recommendation model with CNNs. The proposed framework put forwards a hybrid recommendation model that utilizes content (metadata) about items along with collaborative filtering technique to generate top-n recommendations. Items metadata is used for classifying items into multiple categories, subsequently applying collaborative filters to generate personalized

recommendations for a customer. This work, emphasize on image of the item as a vital metadata property. Classification of items is performed by leveraging pre-trained CNNs and a fully connected neural network to capture the features of each item with respect to their categories. Therefore, the proposed approach employs two models and proceeds in two phases. First phase, item image feature vector extraction and item classification is performed by construction of a multi-class classification model using deep CNNs architecture. Second phase involve in building a hybrid model for rendering top-n personalized recommendations, utilizing extracted feature vectors of classification model. An illustration of the proposed CNN architecture is given in Fig. 1.

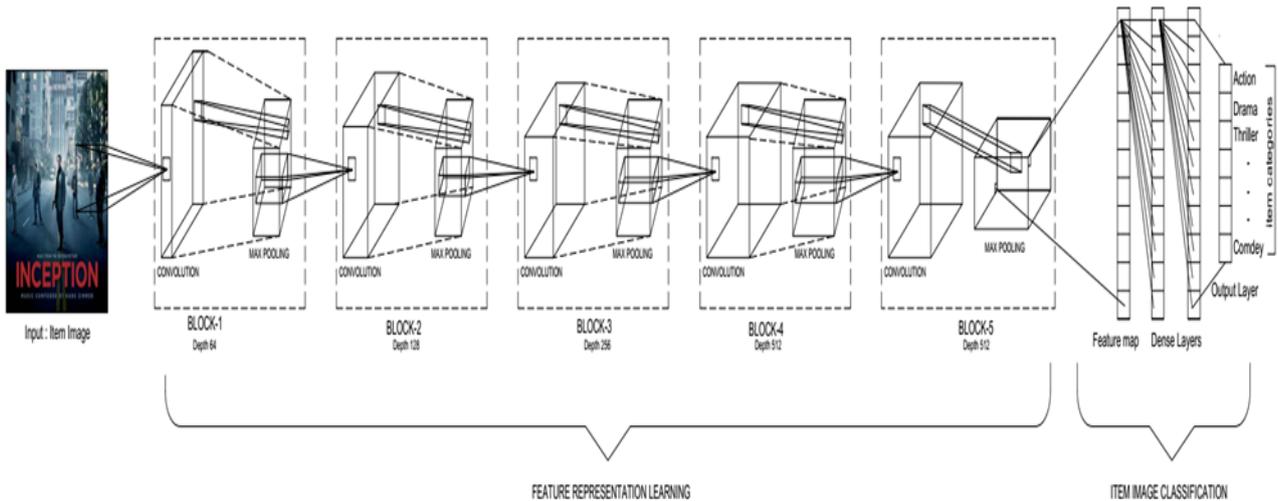


Fig. 1. CNN architecture for classifying Items image

3.1.The Classification Process

In e-commerce scenario, guiding customers in selection and purchase of items from a large scale of alternatives is a difficult task. Most of the e-commerce company’s repositories are overwhelmed with items and their metadata (item specifications, features, images, reviews). When rendering items for customer’s perusal, category of items plays an important role/factor. Items may belong to more than one or more categories (labels). Therefore, here we present a CNNs based item image classification model is developed to identify and extract features of each category. The extracted feature map of each category is used as potential information for generating personalized recommendations.

Although, numerous CNNs architectures are available for image classification, proposed architecture is made up of stacked network of a pre-trained convolution network and fully connected layer. This architecture leverages the weights of the pre-trained model through the concept of transfer learning. Consequently, overhead of learning parameters (weights) starting from random values is reduced during model training. In general, when neural networks are trained on a data, gained knowledge is preserved in the form of weights of the network. When training large scale data on deeply layered network from scratch, training maybe required carrying out for days. One approach, to reduce the training time is enabling a neural network to extract weights and transfer to another neural network. This process is called transfer learning. In this paper we propose to leverage pre-trained models for image classification as we want to classify items’ image with its category. A pre-trained CNN model VGG19 (Visual Geometry Group), is considered in this frame works as it is very large network with 19 layers and is trained on millions of labeled image data of ImageNet. VGG

19 is motivated by the philosophy of going deeper with more layers in the network to attain good representation of features and semantics of image.

Outline of Network Architecture- Network is made up of layers of convolutional, pooling and classification (fully connected) stacked to form CNN architecture to classify an image of an item to the categories it may belong to. It has architecture of [INPUT – BLOCK1 – BLOCK2 – BLOCK3- BLOCK4 – BLOCK5 - CMODEL]

- INPUT [Width x Height x Channel] will be a matrix representation of image, specifying image width, height and also the channels. If it’s a colored image, its value is 3 (Red, Green, and Blue).
- BLOCK1 has convolutional layers and a polling layer, lower level feature extraction is performed using RELU as activation function. Aggregation of activations in feature map is summarized by max-pooling layer. Number of parameter to be learned are (1792+36928)
- BLOCK2 constitutes of two convolutional layers and a polling layer, with RELU as activation function for the units in the layers. Feature map of preceding convolutional layers is summarized by max-pooling layer. Number of parameter to be learned has increased to (73856 +147584)
- BLOCK3 comprises of four convolutional layers and a polling layer, RELU is employed as activation function in the layers. Maximum activations of the feature map

are briefed with max-pooling layer. Number of parameter to be learned are (295168 + 590080 + 590080 + 590080)

- BLOCK4 has four convolutional layers and a polling layer, feature extraction is performed using RELU as activation function. Aggregation of activations is summarized by max-pooling layer. Number of parameter to be learned has increased to (1180160 + 2359808 + 2359808 + 2359808)
- BLOCK5 constitutes of four convolutional layers and a polling layer, high level feature extraction is performed using RELU as activation function. Max-pooling layer is employed to down sample the output of preceding convolutional layers. Number of parameter to be learned has increased to (2359808 + 2359808 + 2359808 + 2359808)
- CMODEL has fully connected layers. There is dense layer with 1024 units having non-linear activation function, followed by dropout layer and layer with sigmoid function. The last layer computes the class scores to perform classification of item category. Class scores measured indicates to what certainty it belongs to a particular category. Activation function RELU is used to capture non linearity in the data.

This CNNs architecture transform the items' image from block to block, initially performing low level extraction and later extracting high level feature from the image, to determine its category/class with scores. During training, at each layer large numbers of parameters are to be learned resulting in a subtle task. The parameters in every layer are updated/ optimized to keep current item's image in processing to be consistent with its class/category in the training set. This is done by employing an optimization technique called gradient descent. For the purpose of attaining optimal values for the parameters, vital properties of the optimizer are learning rate and momentum. In the proposed CNN architecture, stochastic gradient descent technique is employed as optimizer, to optimize the weights by reducing the error of the loss function. Once the feature maps of each item category in repository are extracted, this information will be utilize in the subsequent phase. These feature maps aid the recommendation model to infer customer's preference from previously purchased items. The successive part of the paper moves on to discuss the design and development of such recommendation model.

3.2 Recommendation Process

An item based top-N recommendation model is considered in our approach. The Motivation behind our rational for these models is the fact that, in the context of a repository having a large number of homogenous items belonging to various categories, understanding the customer's preferences is subtle. However, analysing the history of customer's purchases, to identify similar items that customer is likely to purchase can be done efficiently with an item based top-n recommendation model. Task of top-n recommendation model is to generate ranked list of items for a target/active user. In this proposed model, customer's preferences on items are investigated from the previously purchased items' images.

In spite of popularity for user based recommendation models, many of the ecommerce gaits are still predominately

opting for item based recommendation models. This is due to erupting real time computational overhead of finding similar customers among millions of customers and subsequently comparing each product the similar customers has rated. Consequently, item based filtering are preferred for real time performance and scalability. However, a mere calculation of similarity of all items in repository may produce recommendations that are obvious and lack in diversity and serendipity. Therefore, proposed filtering algorithm is designed to leverage items' image and its category to develop list of items for a target customer. Following this strategy gives an opportunity to discover bold recommendations than making customer frustrated with to obvious items.

The presented algorithm is similar to the base line models for top-n recommendations, but in addition it has the ability of performing item image similarity to capture unexpressed customers preferences from the previously purchased items. Extracted feature maps of item's the customer queried and purchased in the past, are compared for similarity with the extracted feature map of item's category in the repository, to retrieve the most similar items that customer may pursue. In the proposed algorithm, ' R ' refers to rating matrix having customers' rating of items. ' C ' denotes the set of customers, S denotes the set of items in the repository. Similarly, $I \subset S$ denotes the sub set of items in S , $I_c \subset S$ denotes set of Items rated by the target customer C and $I_i \subset S$ denote target item for which similar items are to be identified. And also, $i_j \subset S$ refers to item with id j , $NN_i \subset S$ refers to set of Nearest neighbors of Item i . Finally, # symbol in the algorithm represents comment line describing next step.

Algorithm: Top-n recommendation

top_Recommendations (c, R, n) generates top-n recommended items (I) for a target customer (c) given a R -Rating matrix holding the all customers' preferences(ratings) for all items, n - indicates the number of items to be recommended.

In addition, the algorithm utilizes the following
 Set the *threshold* rating for the algorithm to examine items
list –specifies the data type used to store the items data.
add(), *sort()* modules to add and sort elements in a list

The major steps of the algorithms are as follows:

1. Compute the nearest neighbors(items) of a target item
2. Generate top-n items as recommendations using neighbor's

Explaining the algorithmic in detail:

1. Compute the neighbor items in repository for a target item Feature vector

```
List nearest_neighbors ( Item i)
# compute it similarity to the categories in the
repository using CNNs Architecture
neighbors[ ] = { }

for ( j in |Item | )
    if distance( i, j ) < distance_th
        neighbors[ ] = j
return neighbors
```

2. Generate top-n items as recommendations

list top_Recommendations(c, R, n)

```

#Extract list of items rated by customer c from R
Ic = ratings(c)
# Iterate across items rated by target customer
for( k = 1 to |Ic| )
# check if item rating is greater than threshold rating
if ( rk > threshold )
# for each neighbor of rated items compute scores for
ranking
    NNk = nearest_neighbors( ik )
    for( j = 1 to |NNk| )
        ij = NN[j]
        score = similarity(ik,ij) * rk
        ts[j] = t[j] + score
        item_list = item_list.add( ij,
score)
# sort the item list on scores
item_list.sort( )
# extract top-n recommendations
rec_list = item_list[1:n]
return rec_list

```

This item based filtering approach performs the task of ranking using model based algorithm. In the proposed algorithm, item based filtering utilizing nearest neighbor model where neighborhood is determined by items' image feature. Apparently, this reduces computational overheads of user-based approach which employs memory based algorithm. Each item in the repository has feature map pre-generated before the computation of neighbors of items. The features maps are obtained by extracting the final layers of the CNNs as detailed in section III.A above. As the similarity measures once computed can be preserved in a data structure to utilize in successive computations. Similarity measure that can be employed here are *euclidean* distance, *L2 regularization* and *cosine* distance. Furthermore, implementation complexity of this approach is less compared to other traditional and hybrid recommender approaches. This type of approaches can generate moderate novelty, serendipity in recommendations. Item based recommendations with content analysis are known to handle cold start problem.

4. Experimentation

The performance evaluation of the proposed approach for generating effective recommendations is carried out and insights gained are discussed here. The objective of this experimentation is to assess the quality of items' feature maps generated by the classification model, which later are utilized for measuring similarity. Consequently, emphasis is on performance evaluation of the classification model rather than recommendation model. This is due to maiden approach adopted for hybrid item based recommendation using leveraging pre-trained CNN (VGG19) for classifying item category using item images.

Data for evaluation of proposed approach is crawled from the *The Movie Database*, a community built for storing meta data about Movies and Television (TV) shows [44]. With the advancements of web crawler's technology, efficient frameworks for effective evaluation for information retrieval systems are developed. Researchers and practitioners around the world are harnessing data crawled from the real world applications, to gain insights. Motivated by this, we crawled data from the source *The Movie database*, as it is having rich meta-data about its items,

including the images of its items as movie posters. The Movie database web community has provided an open API for scraping data about its items (movies and TV shows). Subsequently, using the API, meta-data about movies (items) is crawled for period of 5 years from 2012-2017. The crawled dataset has more than 5000 records about movies and their meta-data. Data is preprocessed for removing inconsistent, incomplete data and unlabeled records. Each movie may belong to any of the 20 categories often termed as 'movie genres'. Therefore, the proposed multi label classification model developed is appropriate for this dataset having items' image as movie posters. Different types of movie genres (categories) and their distribution is presented in Fig. 2.

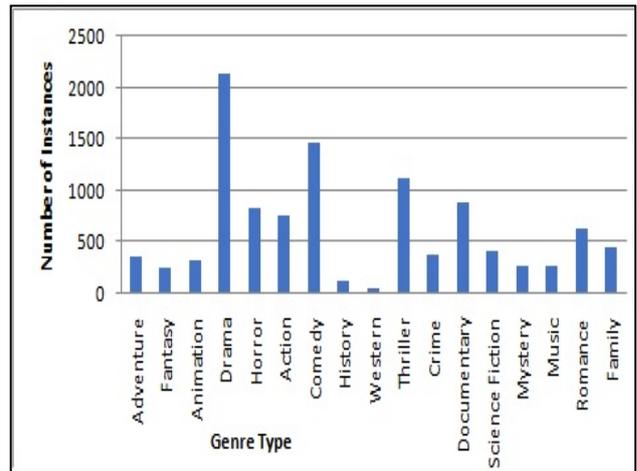


Fig. 2. Genre's Distribution in the Crawled Movie Dataset

Primary exploration of the dataset as presented in Fig. 2, reports that the dataset has more than one thousand records for genres' drama, comedy and thriller. Therefore, items belonging to these categories are classified and performance of the model is presented in Table 2. Initially, all the items belonging to these categories are trained on a CNN, Fig. 3 depicts the training time the model under gone to reduce the error of the loss function. Though the model is trained for 50 epochs, very little reduction in error is observed after 30 epochs.

Table 2. Classification model performance with VGG19 (50 Epochs)

Genre Type	Classification Performance Metrics		
	Accuracy	F1 score	Precision
Drama	0.9259	0.8985	0.9077
Comedy	0.8117	0.6705	0.7564
Thriller	0.7556	0.5065	0.6724

Performance measures shown in Table 2 are obtained after the model is fine-tuned on hyper parameters. Large number of hyper parameters are there in proposed architecture as the network has more than 19 layers. Initially, crawled data is partitioned in ratio of 80:20 to obtain training and testing datasets. And the model is trained on the training set to learn the parameters. Subsequent experimental analysis performed on the model using the testing datasets produced convincing performance. To improve the performance the model is again trained with reduced learning rate compared to initial model training. However,

parameters in all the layers are not relearned. Right most layers holding more semantic representation are relearned leaving forgoing layers parameters unchanged. And also new dense layers are added along with dropout layers. This

improved the models performance significantly. Performing cross validation during training helped the model to have a better representation of knowledge without getting into overfitting.

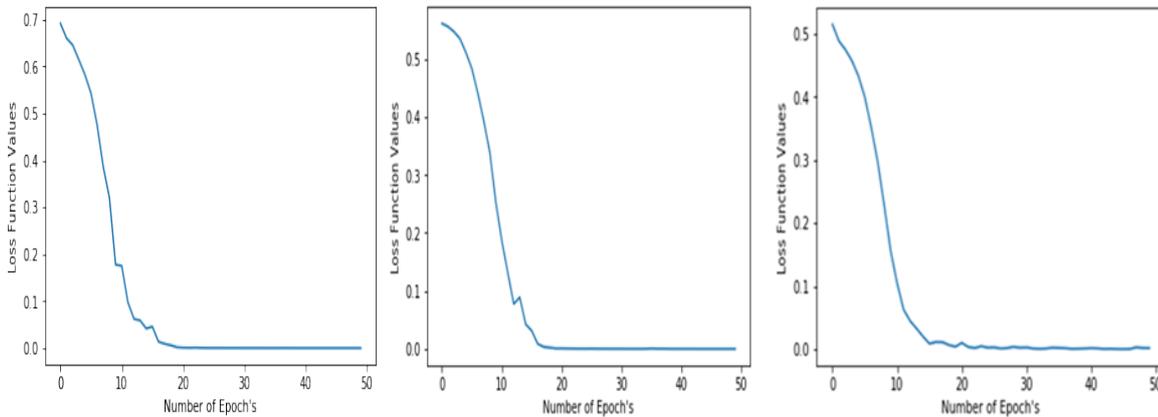


Fig. 3. Model Training Summary for Genres: Drama - Comedy - Thriller (Epoch Vs Loss)

Despite utilizing this items image as supplementary side information the considered approach is viable, generating top-n recommendations with convincing normalized discounted cumulative gain (nDCG) values. And also operations in the algorithm are performed in a good response time over traditional item based recommendation approaches.

5. Conclusion

This paper presented an approach for generating item based top-n recommendations by leveraging stacked neural networks. Recommendations generation process involves in classification of target item into most appropriate categories and filtering of most similar items within those categories. Classification model is a stacked network of CNN having fully connected layers on the top trained on meta-data of the items. Filtering of items is based on content similarity from items' metadata and in specific to item images features. This work proposes leveraging of pre-trained CNNs architectures (VGG19) for identifying image features. This proves to be efficient way of identifying and capturing items' image feature map for performing image similarity. This paper presented the maiden experimental results that can be further improved by fine tuning the model. Limitation of neural networks is again evident in this experiment also, as the accuracy of the classification model improves with the good number of records feed for training.

This study provides an opportunity to advance in integrating deep learning techniques to provide potential side

information to recommendation models. For future work, this paper concludes from the experimental observations that employing ensemble of pre-trained models for classification of item labelling with category can increase the performance. Furthermore, modelling customer's dynamic behaviour can be beneficiary compared to utilization of static models. Deep learning sequence models can be leveraged to modelling customer's behaviour in a given session. Customer situation or contexts can be modelled with aid of emerging Internet of Things (IoT) technology. Processing of heterogeneous data from multiple Internet of Things (IoT) devices needs powerful techniques from deep learning. However, deep learning techniques application in recommending systems need extreme attentiveness to avoid breakdowns. In specific when designing new approaches, time for training and scalability are to be considered properly.

Acknowledgment

This Publication is an outcome of the R&D work undertaken in the project under the Visvesvaraya PhD Scheme of Ministry of Electronics & Information Technology, Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia).

This is an Open Access article distributed under the terms of the Creative Commons Attribution License



References

1. Yann LeCun, YoshuaBengio, Geoffrey Hinton, Deep learning, Nature 521 pp. 436-444 (2015).
2. Bengio, Y., Lamblin, P., Popovici, D. &Larochelle, H. Greedy layer-wise training of deep networks. In Proc. Advances in Neural Information Processing Systems 19 pp. 153-160 (2006)
3. Sutskever, I. Vinyals, O. & Le. Q. V. Sequence to sequence learning with neural networks. In Proc. Advances in Neural Information Processing Systems 27 pp. 3104-3112 (2014)
4. Hinton, G. et al. Deep neural networks for acoustic modeling in speech recognition. IEEE Signal Processing Magazine 29, pp .82-97 (2012).
5. LeCun, Y. et al. Handwritten digit recognition with a back-propagation network. In Proc. Advances in Neural Information Processing Systems pp. 396-404 (1990)
6. M.Egmont-Petersen, D.de Ridder, H.Handels, Image processing with neural networks—a review, Elsevier Pattern Recognition, Volume 35, PP. 2279-2301 (2002)

7. JurgenSchmidhuber, Deep learning in neural networks: An overview, Elsevier Neural Networks, Volume 61, pp. 85-117 (2015)
8. Y. LeCun, F.J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, volume 2, pp. II-97 (2004)
9. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database. IEEE Computer Vision and Pattern Recognition (CVPR) (2009).
10. The MNIST database of handwritten digits by Yann Lecun, Corinna Cortes (1999)
11. <https://www.cs.toronto.edu/~kriz/cifar.html>
12. <http://cocodataset.org/#home>
13. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE (1998).
14. Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton, ImageNet Classification with Deep Convolutional Neural Networks (NIPS) (2012).
15. Matthew D Zeiler, Rob Fergus, Visualizing and Understanding Convolutional Networks, European Conference on Computer Vision 2014, pp. 818-833 (2014).
16. Christian Szegedy et al. Going deeper with convolutions, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)(2015).
17. Karen Simonyan and Andrew Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, Proc. Intl.conf. Learning Representations. (2015).
18. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun Deep Residual Learning for Image Recognition, conference Computer Vision and Pattern Recognition (CVPR),(2016).
19. C. Desrosiers and G. Karypis. A comprehensive survey of neighborhood-based recommendation methods. Recommender Systems Handbook, pp. 107-144 (2011).
20. J. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. Conference on Uncertainty in Artificial Intelligence (1998).
21. Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, pp. 263-272 (2008).
22. G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, pp. 734-749 (2005).
23. C. Aggarwal, J. Wolf, K.-L. Wu, and P. Yu. Horting hatches an egg: a new graph theoretic approach to collaborative filtering. ACM KDD Conference, pp. 201-212 (1999).
24. H. Yildirim, and M. Krishna moorthy. A random walk method for alleviating the sparsity problem in collaborative filtering. ACM Conference on Recommender Systems, pp. 131-138, (2008).
25. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. World Wide Web Conference, pp. 285-295 (2001).
26. D. Billsus and M. Pazzani. Learning collaborative information filters. ICML Conference, pp. 46-54, (1998).
27. W. Lin, S. Alvarez, and C. Ruiz. Efficient adaptive-support association rule mining for recommender systems. Data Mining and Knowledge Discovery, 6(1), pp. 83-105 (2002).
28. R. Devooght, N. Kourtellis, and A. Mantrach. Dynamic matrix factorization with priors on unknown values. ACM KDD Conference (2015).
29. Jingyuan Chen, Hanwang Zhang, Xiangnan He, LiqiangNie, Wei Liu, and Tat-Seng Chua. Attentive Collaborative Filtering: Multimedia Recommendation with Item- and Component-Level Attention. Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval (2017).
30. Paul Covington, Jay Adams, and EmreSargin. Deep neural networks for youtube recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems. ACM, pp. 191-198 (2016).
31. HuifengGuo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In IJCAI. pp. 2782-2788 (2017).
32. XiaoweiJia, Aosen Wang, Xiaoyi Li, GuangxuXun, Wenyao Xu, and Aidong Zhang. Multi-modal learning for video recommendation based on mobile application usage. In 2015 IEEE International Conference on Big Data (Big Data). IEEE, pp. 837-842 (2015).
33. Xinxin Wang and Ye Wang, Improving content-based and hybrid music recommendation using deep learning. In Proceedings of the 22nd ACM international conference on Multimedia, pp. 627-636 (2014).
34. Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song, Recurrent coevolutionary latent feature processes for continuous-time recommendation. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. ACM, pp. 29-34 (2016).
35. Harold Soh, ScoŠ Sanner, Madeleine White, and Greg Jamieson, Deep Sequential Recommendation for Personalized Adaptive User Interfaces. In Proceedings of the 22nd International Conference on Intelligent User Interfaces. ACM, pp. 589-593 (2017).
36. Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song, Recurrent coevolutionary latent feature processes for continuous-time recommendation. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. ACM, pp. 29-34 (2016).
37. Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima, Embedding-based News Recommendation for Millions of Users. In Proceedings of the 23th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM (2017).
38. Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu, Convolutional matrix factorization for document context-aware recommendation. In Proceedings of the 10th ACM Conference on Recommender Systems. ACM, pp. 233-240 (2016).
39. Donghyun Kim, Chanyoung Park, Jinoh Oh, and Hwanjo Yu, Deep Hybrid Recommender Systems via Exploiting Document Context and Statistics of Items. Information Sciences (2017).
40. Xiaoxuan Shen, Baolin Yi, Zhaoli Zhang, Jiangbo Shu, and Hai Liu, Automatic Recommendation Technology for Learning Resources with Convolutional Neural Network. In International Symposium on Educational Technology (ISET). IEEE, pp. 30-34 (2016).
41. SungyongSeo, Jing Huang, Hao Yang, and Yan Liu, Representation Learning of Users and Items for Review Rating Prediction Using Attention-based Convolutional Neural Network. In 3rd International Workshop on Machine Learning Methods for Recommender Systems (2017).
42. Xiaoxuan Shen, Baolin Yi, Zhaoli Zhang, Jiangbo Shu, and Hai Liu, Automatic Recommendation Technology for Learning Resources with Convolutional Neural Network. In International Symposium on Educational Technology (ISET). IEEE, pp. 30-34 (2016).
43. Moshe Unger, Latent Context-Aware Recommender Systems. In Proceedings of the 9th ACM Conference on Recommender Systems. ACM, pp. 383-386 (2015).
44. Montaleao Brum Alves, Renato, Information Retrieval Dataset - Internet Movie Database (IMDB), Mendeley Data, v2, (2017).