SN Computer Science (2022) 3:261
https://doi.org/10.1007/s42979-022-01131-y

ORIGINAL RESEARCH

Deep Learning for Visual-Features Extraction Based Personalized User Modeling

Aymen Ben Hassen[1] · **Sonia Ben Ticha**[1] · **Anja Habacha Chaibi**[1]

Received: 24 August 2021 / Accepted: 4 April 2022 / Published online: 30 April 2022

Abstract

Personalized Recommender Systems help users to choose relevant resources and items from many choices, which is an important challenge that remains actuality today. In recent years, we have witnessed the success of deep learning in several research areas, such as computer vision, natural language processing, and image processing. In this paper, we present a new approach exploiting the images describing items to build a new user's personalized model. With this aim, we use deep learning to extract and reduction dimensionality of latent features describing images. Then we associate these latents features with user preferences to build the personalized model. This model was used in a Collaborative Filtering (CF) algorithm to make recommendations. Experimentally, to evaluate our approach, we apply our approach on two large real data of differents domains, such as fashion and movies, using fashion data sets from Amazon.com and movies data sets from MovieLens, where we show that the best performance of clothing image is more important than the poster of a movie, which explains that the fashion image has an importance in the preferences of the users. Finally, we compare our results to other approaches based on collaborative filtering algorithms.

Keywords Visual features · Deep learning · Transfer learning · Auto-encoder · Recommender systems · Personalized user modeling

Introduction

Every day we are overwhelmed by many choices. Which news or book to read? Which product to buy? Which music to listen or movie to watch? The sizes of these decision areas are often massive. Personalized recommender systems are a solution to this information overload problem. The main purpose of these systems is to provide the user with recommendations that reflect their personal preferences. Although existing recommendation systems are successful in producing relevant recommendations, they face several challenges, such as cold start, scalability problem, data sparsity problem and support for complex data (audio, image, video) describing items to be recommended.

In recent years, we have witnessed the success of deep learning in several research areas. Furthermore, Deep learning models have recently provided exceptional performance and have shown great potential for learning effective representations of data of complex types (e.g., effective representation of functionalities from the content of the image). The influence of deep learning is also ubiquitous, recently demonstrating its effectiveness when applied to information retrieval and recommender systems [38]. After its relatively slow adoption by the recommender system community, deep learning for recommender systems became popular as of 2016 [18].

The most two widely used approaches in personalized recommender systems are Collaborative Filtering (CF), and Content-Based Filtering (CB). CB filtering uses item features for a recommendation, while CF filtering uses only the user-rating data to make predictions. Content-based

This article is part of the topical collection "Web Information Systems and Technologies 2021" guest edited by Joaquim Filipe, Francisco Domínguez Mayo and Massimo Marchiori.

✉ Aymen Ben Hassen
aymen.benhassen@ensi-uma.tn

Sonia Ben Ticha
sonia.benticha@ensi-uma.tn

Anja Habacha Chaibi
anja.habacha@ensi-uma.tn

[1] RIADI Laboratory, University of Manouba, 2010 Manouba, Tunisia

recommendation and collaborative recommendation have often been considered complementary [1]. A hybrid recommendation system is a system that combines two or more different recommendation techniques. There are many ways to hybridize and no consensus has been reached by the research community.

Here we are interested in applications in which visual decision factors has a significant impact on consumers' decisions, such as fashion and movies recommendation. In such settings, visual features play a key role naturally one would not watch a movie without being able to see a poster of this movie [33] , the same with fashion, one would not buy a t-shirt from Amazon without being able to see a image of the product, no matter what ratings or reviews the product had. Likewise then, when building a recommender system, we argue that this important source of information should be accounted for when modeling users' preferences. The users preferences are then used in a collaborative recommendation algorithm user-based to determine the K nearest neighbors of each user.

In previous work [2] we have presented solutions based on deep learing exploiting the images describing items to build the user's personalized model and then to make recommendations by applying a CF algorithm. We have proposed to apply transfer learning to extract latent features of images describing items. We have experimented with our approach only in the area of movie recommendation and more specifically MovieLens data sets. In this paper, we present a new approach exploiting only the images describing items to build the user's personalized model and then to make recommendations by applying a CF algorithm. Due to the high number of latents features from images, and to reduce the expensiveness of user similarity computing, we propose a solution based on deep learing to reduce the size of the number of features of the Items Profile using different methods of dimension reduction. We compare also our results on two differents domains, such as fashion and movies.

Specifically, our system consists of three components, the first component consists of Visual Features using transfer learning to extract latent features describing images of items and Autoencoder for dimensionality reduction. The second component consists in learning the personalized user model by inferring user preferences for latent features of images. The third component consists of using the personalized user model to calculate the k nearest neighbors of each user and finally to make recommendations by applying a user-based CF algorithm.

To take into account the scalability problem, the user model is computed offline and only recommendations are predicted online. To evaluate the performance of our recommender system, we adopted an empirical approach.

In the remainder of this paper, we give in "Related Work", an overview of related work on the use of deep learning for recommender systems. The proposed approach is described in "Proposed Approach". The experimental results of our approach are given in "Performance Study". Finally, in "Conclusions", we conclude with a summary of our findings and some directions for future work.

Related Work

Recent years have witnessed a considerable interest in deep learning in visual features. Feature learning plays an important role in computer vision. Increasingly more advanced technologies contribute to extracting the features describing content of items. We will introduce a brief literature review of deep learning for features extraction.

Deep Learning is one of the next big things in recommendation systems technology. The increasing of the number of studies combining deep learning and recommendation systems may be related to the popularity and overall effectiveness of deep learning in computer science. Concerning recommendation systems, deep learning models have been very successful in learning from different sources and extracting latent features from the complex data used for recommendation. Considering the capacity to big data processing capabilities and interpreting the current trend by applying deep models to recommendation systems, it can be said that collaboration between the two fields will continue to gain popularity soon [38].

Deng et al. [8] proposed a deep learning-based matrix factorization, which employed deep autoencoder to generate initial vectors of users and items and adopted matrix factorization with the pretrained vectors to prediction for recommendation in social rating networks. However, it also requires additional information of user relationship and interest to produce estimation. Hongliang et al. [16] learned deep features by combining the deep belief network (DBN) with a collaborative filtering algorithm to build a video recommendation system. As a typical network of deep learning, the convolutional neural network (CNN) can learn an abstract image deep feature by sharing the local weights [27].

The CNN can directly input the original image and avoid the complex preprocessing; thus, the deep feature learning from a CNN has been widely used in large-scale image processing and analysis. At present, many researchers have combined deep features into recommendation tasks. Geng et al. [12] utilized the CNN to learn the relationships between an image feature and a user deep feature and then implemented the recommendation in social networks. Van Den Oord et al. [34] predicted the latent relationships among resources and completed the relevant recommendations by learning deep feature. Experiments show that a deep network can obtain better results than shallow neural networks. Lei et al. [21] designed

a double convolution network structure by mapping heterogeneous information into the same space. The user's interest was learned through computing the relationships between images and users. The above studies have proved that the CNN can efficiently learn the latent content features of large-scale images by the unique convolutional structure. This great advantage can make a contribution to the analysis of diverse social image data.

To extend their expressive power, various works exploited image data [2, 4–6, 21, 23, 37, 39]. Image is a favorable recommendation item content, as it plays an important role in entertainment, knowledge acquisition, education and social networks. For example, Ben Hassen and Ben Ticha [2] used deep learning to build the user's personalized model using transfer learning to extract latent features of images describing items and then to make movies recommendations, Cui et al. [6] infused product images and item descriptions together to make dynamic predictions, Chu et al. [5] exploited the effectiveness of visual information (for example, images of dining dishes and restaurant furniture) for SR of restaurants. Yu et al. [37] proposed a coupled matrix and tensor factorization model for aesthetic-based clothing recommendation, in which CNNs is used to learn the images features and aesthetic features.

Zhou et al. [39] extracted visual features from images to use visual profiles of user interest in a hotel reservation system. Lei et al. [21] proposed a comparative deep learning model with a Convolutional neural network for a recommendation based on the personalized image. Nguyen et al. [23] presented a personalized recommendation approach for image tags taking into account the item's content based, which combines historical tags information and image features in a factorization model. Using transfer learning, they apply deep learning techniques to classify images to extract latent features from images. Biadsy et al. [4] used item-based transfer learning to solve the problem of data sparsity when user preferences in the target domain are rare or unavailable, while the information needed for preferences exists in another field.

After a review of the state of the art, we found that deep learning has been used in many works to address some challenges of recommendation systems, including data sparsity, cold start, and scalability. Recent work has also demonstrated its effectiveness when applied to the processing and features extraction from data source describing items (image).

Proposed Approach

Our goal is to extract first latent features from images describing the content of items, second to reduce the dimension of this features and thereafter infer user preferences for these features from their preferences for items.

The idea is to exploit the power of deep learning to extract latent features describing images and to reduce the dimensionality. Then, to build a new user's personalized model for personalized user modeling. To that end, we make recommendations by applying a user-based collaborative filtering algorithm. In our approach, each item is described only by one image. Once the latent features of each item have been extracted, they are used for personalized user modeling which will be used in a collaborative filtering algorithm to do recommendations.

Architecture

The general architecture of our approach is presented in Fig. 1. Our approach consists of three main components:

Component 1. Visual Features from Images: this component extracts the latent features by applying transfer learning technique and reduces the dimensionality of each this features of items using Autoencoder. The result of this component is a matrix of items profiles.

Component 2. Personalized user modeling: this component learns the personalized model of users by inferring the utility of each feature extracted for each user, by combining items profiles with the user preferences (rating matrix).

Component 3. Recommendations: This component is responsible for recommending the most relevant items to the current user by calculating the vote prediction for items that are unknown to him. The vote prediction is calculated from its K-Nearest-Neighbors by applying a collaborative user-based filtering algorithm. The personalized user model is then used to compute similarities between users in a user based collaborative algorithm using the rating matrix.

Visual Features from Images

The idea of this component as shown in figure is to extract latent features from images describing item using transfer learning and to reduce the dimensionality of this features using autoencoder.

INPUT: Images describing items. The entry for this component is the set of images describing items. Each item is described by only color image in RGB (Red, Green, Blue) values of size (M ', N'). Each image is modeled by three matrices of size (M ', N'). A matrix R (M ', N') for the color red R, a matrix V (M ', N') for the color green V and a matrix B (M ', N') for the color blue B, so the pixel i, j has three values :

- $R(i, j)$: represents the intensity of red color of pixel (i, j) .
- $V(i, j)$: represents the intensity of green color of pixel (i, j) .

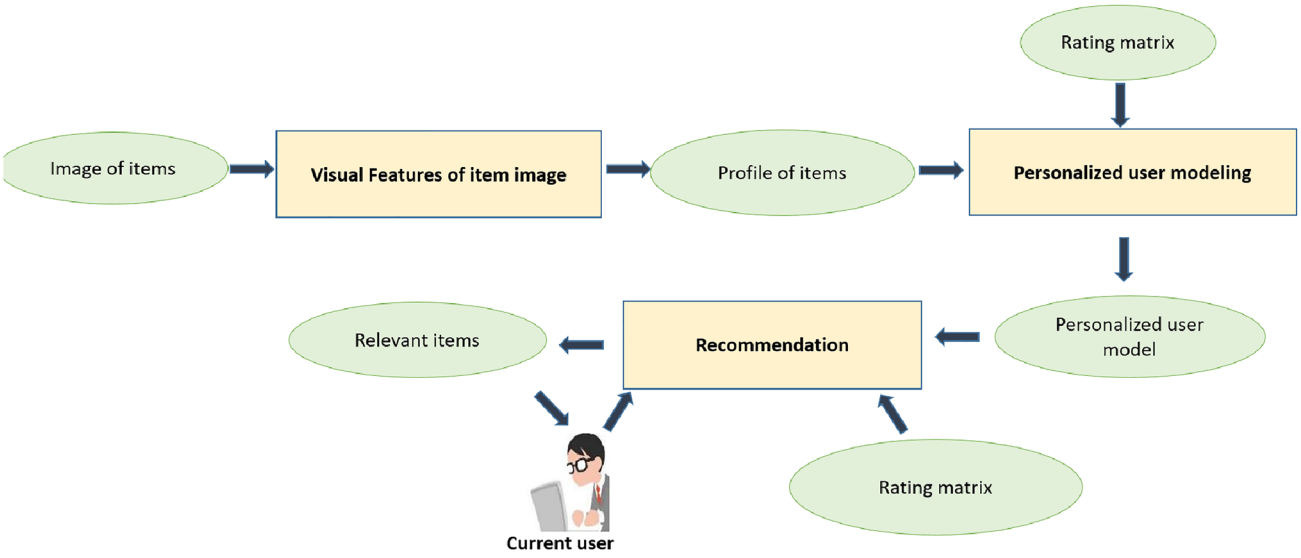

Fig. 1 Proposed architecture

– $B(i, j)$: represents the intensity of blue color of pixel (i, j) .

OUTPUT: Profile of items. After feature extraction and dimensionality reduction, we obtain the latent features of images, which will represent items profile. The profile of the items is then modeled by a matrix of dimension (N, K) , N is the number of items and K is the number of latent features extracted which we will call Matrix Items Profile $MIP_{(N,K)}$, given by (Table 1): where $f_{ij} = MIP(i, f_j)$ represents the value of feature f_j in item i, thus each item i is modeled by a vector $\mathbf{P_i}$ of dimension K defined by

$$\mathbf{P_i} = (f_{i_j})_{(j=1,\ldots,K)} = \begin{pmatrix} f_{i_1} \\ \vdots \\ f_{i_k} \end{pmatrix}$$

Features Extraction

Lately, deep learning showing significant improvement in the computer vision community using the huge number of imaging data sets. Though deep learning a significant number of features are extracted through different layers [7, 24, 28].

Feature extraction is an important technique commonly used in image processing. This technique designates the methods that select and/or combine variables in features. Feature extraction is used to detect features, such as the geometric shape in an image. To do this, we use transfer learning technique to extract latent features of item images. Transfer learning provides a pre-trained model on large sets of images.

This component extract features using transfer learning which is a deep learning technique that uses the convolutional layers with the correction layer ReLu (Linear rectification), some of which are followed by Max-Pooling layers.

Transfer Learning

Transfer learning [19] is a deep learning method and strategy that search to optimize performance on machine learning based on knowledge and other tasks done by another machine learning [36]. Moreover, transfer learning can be a powerful tool for learning on a large target network without overfitting. In addition, transfer learning helps us to use existing models for our tasks. The reasons for using pre-trained models are as follows: first, to transfer a learning by reusing the same model to extract features from a new image data set. Second, it takes more power computing to learn huge models on large data sets. Third, to take a long time to learn the network.

Therefore, we use Transfer Learning method to extract features describing images of items in our data set. We generally observe that the initial layers capture the generic

Table 1 Matrix items profile (MIP)

	f_1	.	f_j	.	f_K
1	f_{11}		f_{1j}		f_{1K}
⋮		⋱	⋮		
i		.	f_{ij}	.	
⋮			⋮	⋱	
N	f_{N1}		f_{Nj}		f_{NK}

features, while the deeper ones become more specific in features extraction. It consists in exploiting pre-trained models on large complex data sets. There are many CNN architectures, such as VGG, ConvNet [29], ResNet [30], etc. In the proposed transfer learning method, we used VGG-16 and VGG-19 as basic models [29], previously pre-trained for feature extraction task from ImageNet data set[1]. ImageNet is a data set of over 15 million labeled high-resolution images belonging to roughly 22,000 categories. Moreover, it is organized according to the WordNet hierarchy. We use convolutional layers of two models to extract features from our data set, and we eliminate fully connected layers for classification task. Therefore, VGG architecture for the two pre-trained models is a composite of five blocks of convolutional layers, some of which are followed by Max-Pooling layers.

The image is passed through a stack of convolutional layers, where the filters were used with a very small receptive field: 3×3. In one of the configurations, it also utilizes 2×2 convolution filters, which can be seen as a linear transformation of the input channels. The convolution stride is fixed to 1 pixel, the spatial padding of convolutional layer input is such that the spatial resolution is preserved after convolution, i.e., the padding is 1-pixel for 3×3 convolutional layers. Spatial pooling is carried out by five max-pooling layers, which follow some of the convolutional layers. Max-pooling is performed over a 3×3 pixel window, with stride 2. In the VGG16: 13 convolutional layers. In the VGG19 model: 16 convolutional layers. The width of convolutional layers (the number of channels) is rather small, starting from 64 in the first layer and then increasing by a factor of 2 after each max-pooling layer, until it reaches 512.

Dimension Reduction

The dimension reduction methods make it possible to project the features into a reduced dimension to deal with the scalability problems [26]. Several techniques exist in the literature for reducing the dimension of a matrix. Elkahky et al. [11] used Top-K features dimension reduction technique, such as selecting the most relevant Top-K features (eliminating non-significant features with a high zero rate). In addition, they use RBM[2] to reduce the size to manage large-scale data sets. Desrosiers et al. [9] used other methods, such as SVD[3] [20], that is to reduce the dimension of rating matrix, or to reduce the dimension of similarity matrix. Wang et al. [35] used an Auto-Encoder (AE) to reduce the size of data set and compare this technique with different dimension reduction techniques.

The number K of features thus obtained may be very high. It would be interesting to be able to reduce the K dimension of the *MIP* matrix by reducing the number of features and thus deal with the scalability problems. We choose to reduce the number of features of the MIP (Matrix Items Profile) using as techniques the Top-K features, SVD and Auto-Encoder.

We propose as a first solution, to apply a **Top-K features**, this technique selects the most relevant features. More specifically, we eliminated features with a number of zero greater than a given threshold *NF* that is determined empirically.

Singular Value Decomposition (SVD) allows us to project a dimension of the matrix (either rows l or columns c) onto another dimension defined by latent variables described by the singular values of initial matrix. The dimension of the projection is defined by the number of singular values of the initial matrix which is equal to the minimum between l and c. Latent semantic analysis (LSA) reduces the projection dimension by keeping only the largest R singular values.

We propose as a second solution, to apply a Latent Semantic Analysis (LSA) [10] of rank R of *MIP* matrix. The rank R is well below the number of features ($R << |F|$). LSA uses a truncated SVD keeping only the R largest singular values and their associated vectors. Therefore, the rank-R approximation matrix of the *MIP* matrix is provided by Formula (1)

$$MIP \approx I_{J,R} * \Sigma_{R,R} * V^{t}_{R,|F|}. \quad (1)$$

The rows in I_R are the item vectors in LSA space and the rows in V are the feature vectors in LSA space. Thus, each item is represented in the LSA space by a set of R latent variables instead of the features of F.

Auto-encoder based dimensionality reduction

An auto-encoder [3] is an artificial neural network for learning efficient codings, that compresses the data to lower dimension and then reconstructs the input back. Auto-encoder finds the representation of the data in a lower dimension by focusing more on the important features getting rid of noise and redundancy. It is based on encoder–decoder architecture, where encoder encodes the high-dimensional data to lower dimension and decoder takes the lower dimensional data and tries to reconstruct the original high-dimensional data. It is trained to encode input $\mathbf{x}$ into some representation $\mathbf{y}$ through a deterministic mapping

$$\mathbf{y} = s(Wx + b), \quad (2)$$

[1] http://www.image-net.org/.

[2] Restricted Boltzmann Machines.

[3] Singular Value Decomposition.

Table 2 Rating matrix (Mv)

	1	.	i	.	N
1	v_{11}	?	v_{1i}	?	v_{1N}
⋮	?	⋱	⋮	?	?
u	?	…	v_{ui}	…	?
⋮	?	?	⋮	⋱	?
L	v_{L1}	?	v_{Li}	?	v_{LN}

Table 3 Personalized user model (matrix of user profile (MUP))

	f_1	.	f_j	.	f_K
1	f_{11}		f_{1j}		f_{1K}
⋮		⋱	⋮		
u		…	f_{uj}	.	
⋮			⋮	⋱	
L	f_{L1}		f_{Lj}		f_{LK}

where **s** is a non-linearity function, such as the sigmoid. And the code y is then decoded back into a reconstruction of same shape through a similar transformation. Where s is a non-linearity function, such as the sigmoid. In addition, the code y is then decoded back into a reconstruction of same shape through a similar transformation

$$\mathbf{x} = s(W'y + b'), \tag{3}$$

and reconstruction error is to be minimized.

An auto-encoder with only one hidden layer and the mean squared error criterion are used to train the network, then the k hidden units learn to project the input in the span of the first k principal components of the data. However, if the hidden layer is non-linear, the auto-encoder behaves differently from PCA.

Although auto-encoder has been proposed for a long time, it is difficult to train encoders with deep architecture until recently Restricted Boltzmann machine (RBM) was used to train a Deep Belief Network [15]. Hinton et al. [14] uses RBM to train an deep auto-encoder which is used for dimensionality reduction

Personalized User Modeling

In this section, we will present the second component allowing personalized user modeling. The idea is to build a new user profile.

INPUT:

- Items profile modeled by *MIP* result of first component.
- Usage data is represented by rating matrix Mv having L rows and N columns. The lines represent the users and the columns represent the items. Ratings are defined on a scale of values. The rating matrix has missing value rate exceeding 95%, where missing values are indicated by a "?", $v_{u,i}$ the rating of user u for item i, given by (Table 2)

OUTPUT: At the end of personalized user modeling, we obtain a personalized user model which is represented by a matrix which we will call "*Matrix User Profile*" ($MUP_{L,K}$) without missing values, having L rows representing the users and K columns representing the features. This profile defines user preferences for the extracted features describing the items based on their assessments for these same items. $MUP(u, f)$: represents the utility of feature f for user u as shown in Fig. 3.

Personalized user modeling The idea is to infer the utility of each feature of items (the result of component 1) for each user. To do this we were inspired by [32] which gives different formulas for calculating matrix of user profiles. We used the formula which gave better results (see following Eq. (4)).

$$\mathrm{MUP}_{(u,j)} = \sum_{i \in I_{u_{\mathrm{relevant}}}} v_{u,j} \times \mathrm{MIP}_{(i,j)}. \tag{4}$$

Computing $I_{u_{\mathrm{relevant}}}$:

We denote by $I_{u_{\mathrm{relevant}}}$ the set of relevant items of user u. To compute $I_{u_{\mathrm{relevant}}}$, we used the formula given in [31]. An item i is relevant for a user u of U if it satisfies the following two conditions:

$$\begin{cases} v_{ui} \in [v_{\min}..v_{\max}] \; and \; v_{\mathrm{neutral}} = \frac{v_{\max}}{2} \\ I_{u_{\mathrm{relevant}}} = \{ i \in I_u \; / \; v_{ui} \geq \overline{v_u} \; and \; v_{ui} > v_{\mathrm{neutral}} \} \end{cases}, \tag{5}$$

where $\overline{v_u}$ indicates the average of rating. Using the user's average vote as a threshold to determine the relevance of an item has two advantages. The first is to avoid adding a new parameter. The second is the personalization of the threshold which allows taking into account the variation in the attribution of the marks, since all the users do not rate in the same way.

Recommendation

Among the existing collaborative approaches, CF algorithms based on the K-Nearest-Neighbors algorithm [9] are very popular because of their simplicity, their efficiency, and their ability to produce relevant personalized recommendations. The idea is to take advantage of the efficiency and simplicity of these algorithms to make recommendations using the Personalized User Model to determine the nearest neighbors of the current user.

The personalized user model is used to compute similarities between users. Similarities are used to select the

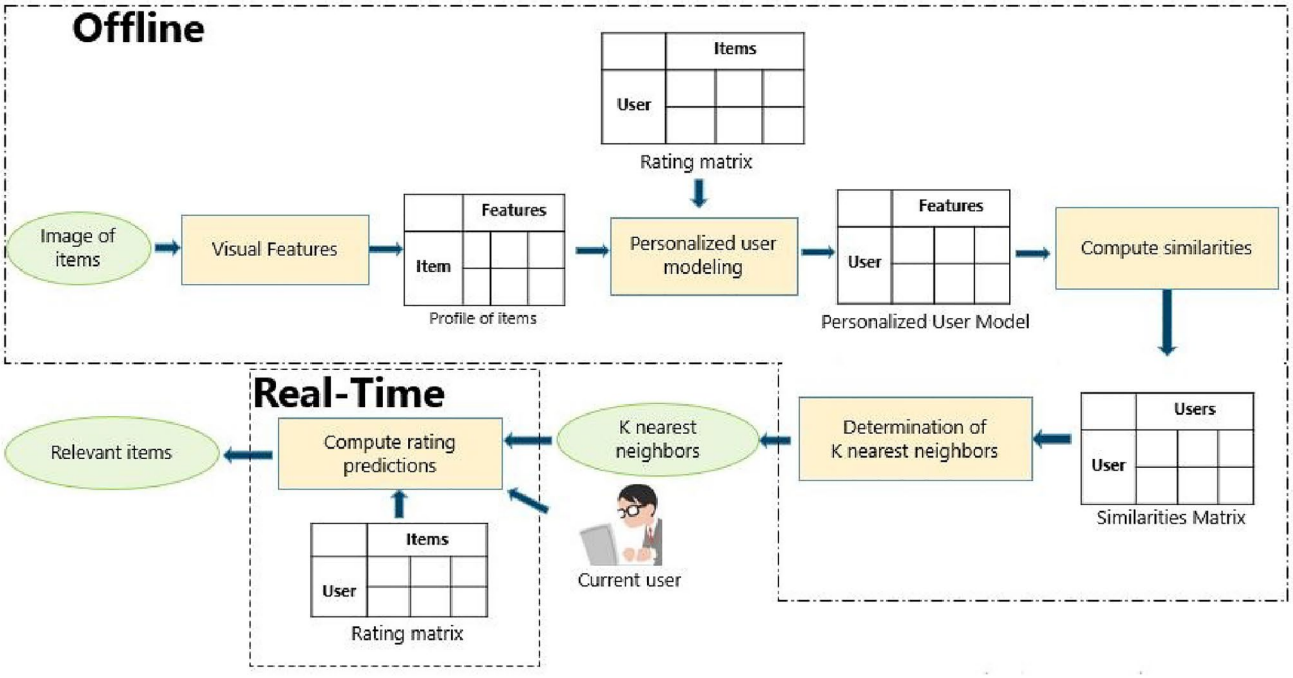

Fig. 2 Synthesis of our approach

K nearest neighbors of the current user in a user-based collaborative filtering algorithm [25].

The User Profile u (PU_u) is represented by index line u in User Profile matrix (MUP) modeling the personalized model of users. Computing the similarity between two users then amounts to calculating the correlation between their two profiles. In our case, the user profile u (PU_u) models the importance of the hidden features for the user u. The Cosine is utilized for calculating the correlation between two users u and v. It is defined by the Formula (6).

$$\text{sim}(u,v) = \cos(\mathbf{PU_u}, \mathbf{PU_v}) = \frac{\mathbf{PU_u} \cdot \mathbf{PU_v}}{||\mathbf{PU_u}||\,||\mathbf{PU_v}||}. \quad (6)$$

To compute predictions of rate value of an item i not observed by the current user u_a, we applied the Formula (7) keeping only the K nearest neighbors. The similarity between u and u_a being determined in our case from their user profiles applying the Formula (7).

$$\text{pred}(u_a, i) = \bar{v_{u_a}} + \frac{\sum_{k \text{ nearest neighbors}} \text{sim}(u_a, u)(u_{ui} - \bar{v_u})}{\sum_{k \text{ nearest neighbors}} |\text{sim}(u_a, u)|}. \quad (7)$$

The rating prediction in our approach is calculated by applying user-based collaborative filtering algorithm. In the standard algorithm, the similarity between users is calculated from rating matrix. In our case, we use *MUP* matrix modeling the personalized users profile to calculate the similarity between users.

Our approach provides solutions to the scalability problem. The first two components, namely, feature extraction and personalized user modeling, are executed in offline mode. To reduce the time complexity of computing the rating prediction, the determination of K nearest neighbors of each user is also computed in offline mode, keeping only the k nearest to them. The calculation of predictions for the current user is executed in real-time during his interaction with e-service (Fig. 2).

Performance Study

A recommendation algorithm aims to improve the usefulness of an e-service towards its users by increasing their satisfaction. Thus, measuring user satisfaction in terms of recommendation represents an important evaluation criterion for any recommendation algorithm.

To evaluate our approach, we opted for offline evaluation mode. The offline evaluation allows the performance of several recommendation algorithms to be compared objectively. We have adopted an empirical approach. The performances of our approach were analysed through different experiments on data sets.

We evaluated the performance of our approach by measuring the accuracy of the recommendations, which measures

the capacity of a recommendation system to predict recommendations that are relevant to its users. We measured the accuracy of the prediction by calculating the Root Mean Square Error (RMSE) [13], which is the most widely used metric in CF research literature.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{(u,i)\in T} (\mathrm{pred}(u,i) - v_{ui})^2}{|T|}}, \tag{8}$$

where T is the set of couples (u, i) of R_{test} for which the recommendation system predicted the value of the vote. It computes the average of the square root difference between the predictions and true ratings in the test data set, lowers the RMSE is, better the accuracy of predictions.

Experimental Data Sets

We experimented our approach to real data from two different fields of applications in the area of movie and clothing.

– **Movies data sets:** We use two data sets: data set for item content and data set to train the recommendation models.

For the item content data, we used the TMDb[4] (The Movie Database) data set to extract movie posters. TMDb provides the content of items data set and contains 10,590 movie posters with an image size of 500 by 750.

We used the HetRec 2011 data set of the MovieLens recommender system[5] [17] that links the movies of MovieLens data set with their corresponding web pages at Internet Movie Database (IMDb), which contain user ratings. The HetRec-2011 data set provides the usage data set and contains 1,000,209 explicit ratings of approximately 3900 movies made by 6040 users with approximately 95% of missing values.

The usage data set has been sorted by the timestamps, in ascending order, and has been divided into a training set (including the first 80% of all ratings) and a test set (the last 20% of all ratings). Thus, ratings of each user in the test set have been assigned after those of the training set.

– **Fashion data sets:** The Amazon data set is the consumption records from Amazon.com [22]. In this paper, we use the clothing shoes and jewelry category filtered with 5-score (remove users and items with less than 5 purchase records) to train all recommendation models. There are 39,371 users, 23,022 items, and 278,677 records in total. The sparsity of the data set is 99.969%. The images available from this data set are of high quality (typically centered on a white background) and have previously been shown to be effective for recommendation tasks.

Performance Evaluation of Features Extraction with VGG Models

To evaluate our approach, first, we started by features extraction, and we took all the features extracted of transfer learning. We used the pre-trained models VGG16 and VGG19 for transfer learning technique in the first component 3.2 (Visual Features from Images) available included in the library keras[6] with Python programming language[7] with version 3.7 and run on TensorFlow[8].

This technique gives us profile item modeled by Matrix Item Profile (*MIP*) containing the latent features for each movie poster *i*. Items in the row and the features of each item in the column. Each element has the importance of feature f for each item *i* which is a value between [0.100].

The precisions of the two models (VGG16 and VGG19) from two different fields of applications (movies and fashion) are shown in Fig. 3. The RMSE is plotted against the number K of neighbors. In all cases, the RMSE converges between 50 and 60 neighbors. The accuracy of predictions ratings of the VGG19 model is higher than that observed by VGG16, for all the neighbors. The best performance from movies-data set is obtained by VGG19 whose RMSE value is equal to 0.9263 for 60 neighbors. For VGG16, the best performance is obtained for the same number of neighbors with a RMSE equal to 0.9309. On the other hand, the best performance from fashion-data set is obtained by VGG19 whose RMSE value is equal to 0.9161 for 60 neighbors. For VGG16, the best performance is obtained for the same number of neighbors with a RMSE equal to 0.9243.

Performance Evaluation of Dimension Reduction

Dimension Reduction with Top-K Features

To improve the performance of our approach, we reduced the size of the Matrix Items Profile (MIP) by selecting the most relevant features. More specifically, we eliminated the features with a number of zero greater than a given threshold = "$\mathrm{NF}_{\mathrm{zero}}$" that is determined empirically. Where threshold is the rate % of zero in the features.

Figure 4 illustrates the performance of selecting the features "$\mathrm{NF}_{\mathrm{zero}}$" in fixing $K = 60$ of K-Nearest-Neighbors. In fact, for the VGG19 model, the initial number of features is equal to 25028, the selection of features from the item

[4] https://www.themoviedb.org/.

[5] https://grouplens.org/datasets/hetrec-2011/.

[6] https://keras.io/.

[7] https://www.python.org/.

[8] https://www.tensorflow.org/.

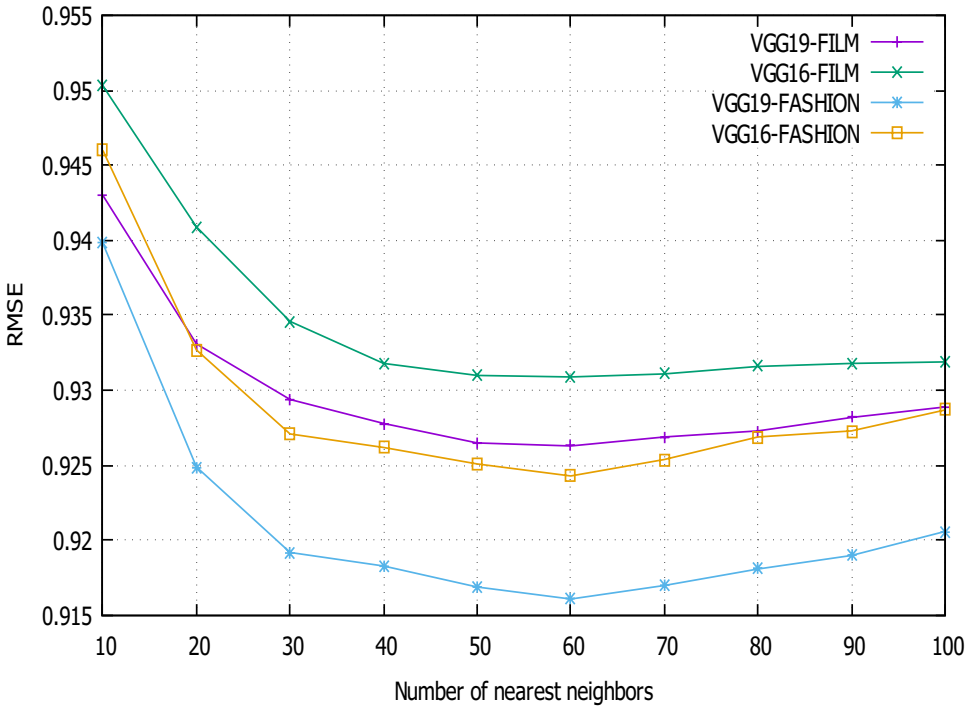

Fig. 3 Evaluation with VGG models

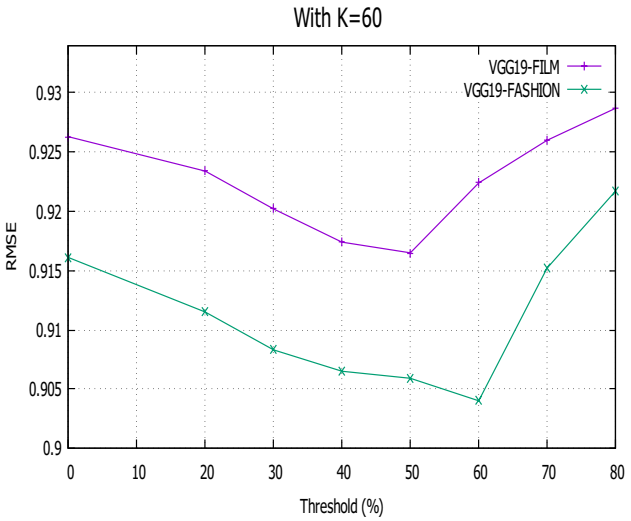

Fig. 4 Performance evaluation of selecting the relevant Top-K features

profile matrix (MIP) is 0%, the accuracy of recommendations of film-images which has reached the value of RMSE = 0.9263. On the other hand, the accuracy of recommendations of fashion-images reached the value of RMSE = 0.9161 of the accuracy of recommendations.

The feature selection of the matrix item profile increases the accuracy until its rank reaches a threshold value of the Percentage selection of features from which the accuracy begins to decrease. This observation remains the same with the other data set. The threshold value for the accuracy of

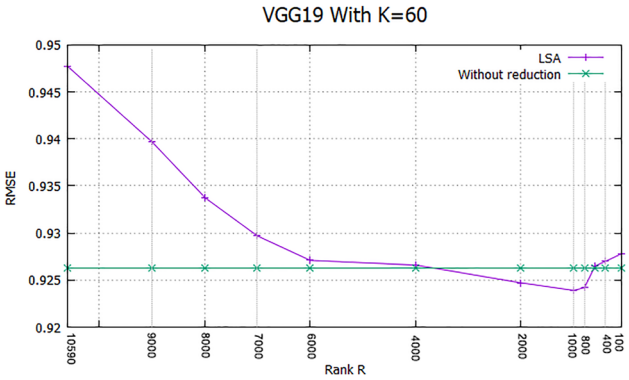

Fig. 5 Performance Evaluation of LSA

recommendations of the VGG19 model of movies-data set is equal to RMSE = 0.9165 corresponds to 50% of the selection of features from the Matrix Item Profile (MIP). On the other hand, with fashion-data set, the threshold value for the accuracy of recommendations is equal to RMSE = 0.9040 corresponds to 60% of the selection of features.

The dimension reduction made possible not only to reduce the size of the model and thus to improve its performance in terms of scalability, but also to improve its performance in terms of precision of the recommendations.

Dimension Reduction with LSA

In Fig. 5, the RMSE has been plotted with respect to the LSA rank. We reduce the size of *MSI* in fixing $k = 60$ of

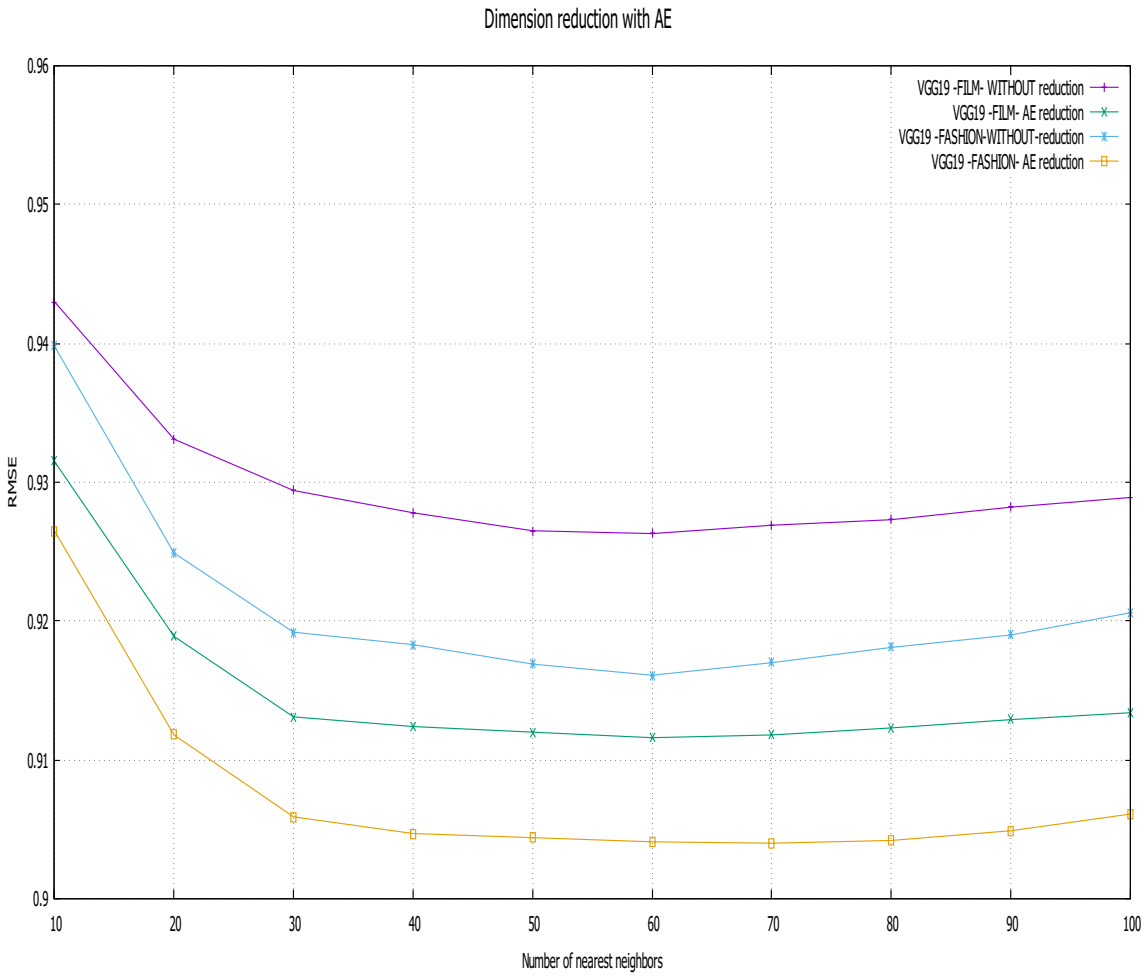

Fig. 6 Performance evaluation of auto-encoder

K-Nearest-Neighbors of the VGG-19 model by applying a LSA with rank R. The performances are compared with those obtained without reduction of the dimension (curve in green).

The factorization of a matrix MPI (10,590, 25,028) is the application an SVD, so the number of latent features is equal to $R = \min(10{,}590, 25{,}028)$. The factorization of MPI matrix resulted in a degradation of precision of the recommendations which reached the value of RMSE = 0.9477 for $R = 10{,}590$ against RMSE = 0.9263 without factorization. The dimension reduction increases the precision until reaching a threshold value of R from which the precision begins to decrease. The optimum is reached for R equal to 1000 with RMSE = 0.9239 slightly better than that obtained without dimension reduction (RMSE = 0.9263). Although the LSA does not improve the accuracy, dimension reduction is significant. Thus, it allows to reduce the cost of users similarity computing, specially when the number of features is very high.

Dimension Reduction with Auto-encoder

To improve the performance of our approach, we also use auto-encoder to reduce the dimension of image feature space. Figure 6 illustrates the performance of auto-encoder, the RMSE has been plotted with respect to the number K of neighbors in the k-Nearest-Neighbor algorithm, with $K \in [10, 100]$. The precisions of the VGG19 model from two different fields of applications (movies and fashion) are shown in Fig. 6. The RMSE is plotted against the number K of neighbors. In all cases, the RMSE converges between 50 and 60 neighbors. The accuracy of predictions ratings when applying auto-encoder is higher than that observed without any method of dimension reduction, for all the neighbors. The best performance is obtained from fashion-field by VGG19 whose RMSE value is equal to 0.9041 for 60 neighbors.

On the other hand, the best performance from movie-field is obtained by VGG19 by applying Auto-encoder whose RMSE value is equal to 0.9116 for 60 neighbors. In fact, we

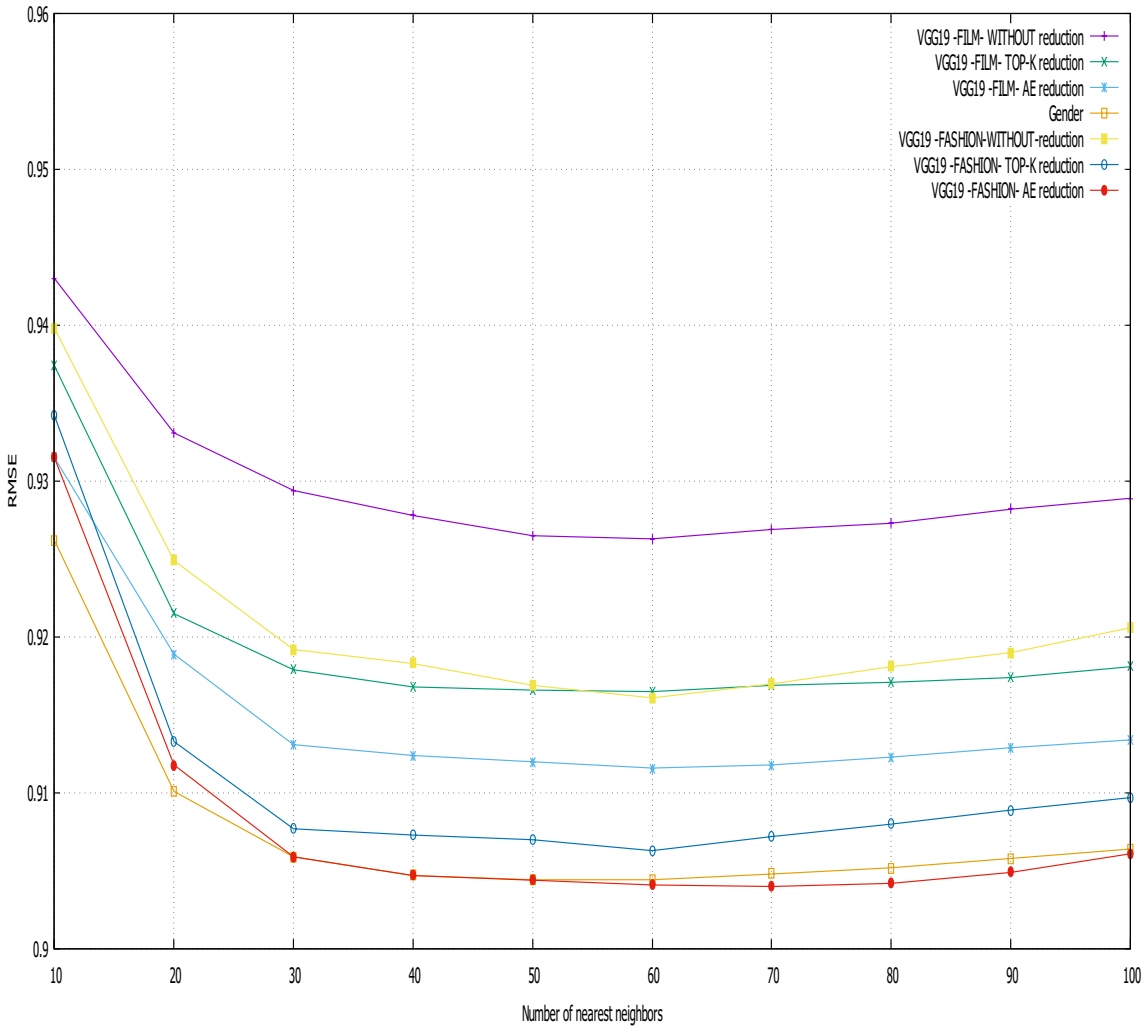

Fig. 7 Comparative results of our approach against other approaches based CF

can say that the best performance which apply autoencoder with fashion-field of VGG19 model.

Comparative Results of Our Approach Against Other Approaches Based on CF

In Fig. 7, the RMSE has been plotted with respect to the number K of neighbors in the k-Nearest-Neighbor algorithm, with $K \in [10, 100]$. We compared the performance of our approach using VGG19 model compared to our previous work "Transfer Learning to Extract Features for Personalized User Modeling" [2] and to a "User Semantic Collaborative Filtering" approach [31] which treated with different text attributes describing movies (Genre, Origin).

We represented the performances of different experiments on the two data set:

- MovieLens data set: the Genre of movie attribute (e.g., comedy, drama) represented by the "Genre" plot, the movie poster without reduction of the dimension represented by the "VGG19 -Film- without reduction" plot and the movie poster with different methods of dimensions reduction with Autoencoder and Top-K features in size.

- Amazon data set: We evaluated the performance of our approach of fashion data set with VGG19 model using different methods of dimension reduction(Autoencoder and Top-K features) in size of the number of features for items profile.

By analyzing the plots of the graph, we see that all the plots have the same appearance, the RMSE decreases to a given value of K (The Nearest Neighbors) then increase. All the plots converge for N between 50 and 60 neighbors. Using Movielens data set, the accuracy of the genre rating predictions is higher than that observed by when using VGG19 with Autoencoder method, which themselves are higher to

Table 4 Experimental results of our models

Model	MovieLens	Amazon Fashion
VGG16	0.9309	0.9243
VGG19	0.9263	0.9161
VGG19 with TOP-K features	0.9165	0.9063
VGG19 with AE	0.9116	0.9041

those recorded by our previous approach [2] which processes the image content of items using VGG19 and this for all neighbors. The best performance is obtained by the movie Genre attribute whose RMSE value is equal to 0.9044 for 60 neighbors, again of the order of 1 point compared to our approach with AE whose RMSE is equal to 0.9116 for the same number of neighbors.

Using amazon data set, which processes the clothing image content of items the accuracy of VGG19 model of extraction features and with Autoencoder reduction method is higher than that observed by when using VGG19 with topK features and this for all neighbors. The best performance is obtained by AE method whose RMSE value is equal to 0.9041 for 60 neighbors, again of the order of 2 points compared to our approach with TopK features whose RMSE is equal to 0.9063 for the same number of neighbors.

Table 4 compares the best performance of each model on the two data set (MovieLens and Amazon Fashion) of our approach.

In conclusion, we can say that the best performance which deals with the textual data describing the item (Genre). The results of our approach are acceptable compared to the results of [2, 31] which explains this by the fact that the poster of a movie has an importance in the preferences of the users and it may not be discriminating enough as the genre. Now and when using AE to reduce the dimension with the clothes we can say that the image of the clothes is more better than our previous approach when we used the movie poster which explains this by the fact that the image of clothing has an more importance in the preferences of the users than the movies' poster. Thus, we used transfer learning with the pre-trained VGG-16 and VGG19 models with ImageNet data set but if we will build a model Convolutional Neural Network (CNN) of classification task by trained from the movie poster data set, then we will apply transfer learning of our data set. Perhaps, in the case, the results can be better.

Conclusions

In this paper, we have proposed to apply transfer learning to extract latent features of images describing items by subsequently reducing the dimensionality of this latent features. We have used the resulting model for personalized user modeling by inferring user preferences for latent features of images from the history of their preferences for items and thus building the user model. The personalized model obtained was then user used collaborative filtering algorithm on users to make recommendations.

We evaluated the performance of our approach by applying two different feature extraction models VGG16, VGG19. To improve the performance of our approach, we applied different methods Top-K features, LSA and Autoencoder for the reduction dimension. Finally, we compared the accuracy of our approach to real data from two different fields of applications in the area of movie and clothing. We also compared to other approaches based on hybrid filtering which deals with different text attributes describing items.

Declarations

Conflict of Interest The authors declare that they have no confict of interest.

References

1. Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng. 2005;6:734–49.
2. Be Hassen A, Ticha SB. Transfer learning to extract features for personalized user modeling. In: WEBIST, 2020;15–25.
3. Bengio Y. Learning deep architectures for AI. Now Publishers Inc 2009.
4. Biadsy N, Rokach L, Shmilovici A. Transfer learning for content-based recommender systems using tree matching. In: International Conference on Availability, Reliability, and Security, pp. 387–399. Springer 2013.
5. Chu WT, Tsai YL. A hybrid recommendation system considering visual information for predicting favorite restaurants. World Wide Web. 2017;20(6):1313–31.
6. Cui Q, Wu S, Liu Q, Zhong W, Wang L. Mv-rnn: a multi-view recurrent neural network for sequential recommendation. IEEE Trans Knowl Data Eng. 2018.
7. de Souza GB, da Silva Santos DF, Pires RG, Marana AN, Papa JP. Deep features extraction for robust fingerprint spoofing attack detection. J Artif Intell Soft Comput Res. 2019;9(1):41–9.
8. Deng S, Huang L, Xu G, Wu X, Wu Z. On deep learning for trust-aware recommendations in social networks. IEEE Trans Neural Netw Learn Syst. 2016;28(5):1164–77.
9. Desrosiers C, Karypis G. A comprehensive survey of neighborhood-based recommendation methods. In: Recommender systems handbook. Springer, 2011;107–44.
10. Dumais ST. Latent semantic analysis. Ann Rev Inf Sci Technol. 2004;38(1):188–230.
11. Elkahky AM, Song Y, He X. 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, pp. 278–88. International World Wide Web Conferences Steering Committee 2015.
12. Geng X, Zhang H, Bian J, Chua TS. Learning image and user features for recommendation in social networks. In: Proceedings

of the IEEE International Conference on Computer Vision, 2015;4274–82.
13. Herlocker JL, Konstan JA, Terveen LG, Riedl JT. Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst (TOIS). 2004;22(1):5–53.
14. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science. 2006;313(5786):504–7.
15. Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. Neural Comput. 2006;18(7):1527–54.
16. Hongliang C, Xiaona Q. The video recommendation system based on DBN. In: 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, pp. 1016–21. IEEE 2015.
17. IMDB: Internet movie database, 2019; https://www.imdb.com/, accessed Jun 2019.
18. Karatzoglou A, Hidasi B. Deep learning for recommender systems. In: Proceedings of the eleventh ACM conference on recommender systems. ACM, 2017; pp. 396–7.
19. Karpathy A. et al. Cs231n convolutional neural networks for visual recognition. Neural Netw 2016;1.
20. Koren Y. 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, 2008; pp. 426–34.
21. Lei C, Liu D, Li W, Zha ZJ, Li H. Comparative deep learning of hybrid representations for image recommendations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016;pp. 2545–53.
22. McAuley J, Targett C, Shi Q, Van Den Hengel A. Image-based recommendations on styles and substitutes. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval, 2015; pp. 43–52.
23. Nguyen HT, Wistuba M, Schmidt-Thieme L. Personalized tag recommendation for images using deep transfer learning. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 2017; pp. 705–720.
24. Rashid M, Khan MA, Sharif M, Raza M, Sarfraz MM, Afza F. Object detection and classification: a joint selection and fusion strategy of deep convolutional neural network and sift point features. Multimed Tools Appl. 2019;78(12):15751–77.
25. Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J. Grouplens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM conference on Computer supported cooperative work, 1994; pp. 175–86.
26. Schafer JB, Frankowski D, Herlocker J, Sen S. Collaborative filtering recommender systems. In: The adaptive web. Springer, 2007; pp. 291–324.
27. Schmidhuber J. Deep learning in neural networks: an overview. Neural Netw. 2015;61:85–117.
28. Sharif M, Attique Khan M, Rashid M, Yasmin M, Afza F, Tanik UJ. Deep CNN and geometric features-based gastrointestinal tract diseases detection and classification from wireless capsule endoscopy images. J Exp Theor Artif Intell. 2019;33(4):577–599.
29. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint. 2014. arXiv:1409.1556.
30. Targ S, Almeida D, Lyman K. Resnet in resnet: Generalizing residual architectures. arXiv preprint. 2016. arXiv:1603.08029.
31. Ticha SB. Hybrid personalized recommendation. Ph.D. thesis, Faculty of Sciences of Tunis 2015. https://hal.univ-lorraine.fr/tel-01752090
32. Ticha SB, Roussanaly A, Boyer A, Bsaïes K. Feature frequency inverse user frequency for dependant attribute to enhance recommendations. In: The Third Int. Conf. on Social Eco-Informatics - SOTICS. IARIA, Lisbon, Portugal 2013.
33. TMDB: The movie database. 2019. https://www.themoviedb.org/, accessed Jun 2019.
34. Van Den Oord A, Dieleman S, Schrauwen B. Deep content-based music recommendation. In: Neural Information Processing Systems Conference (NIPS 2013), vol. 26. Neural Information Processing Systems Foundation (NIPS) 2013.
35. Wang Y, Yao H, Zhao S. Auto-encoder based dimensionality reduction. Neurocomputing. 2016;184:232–42.
36. Wei Y, Xia W, Huang J, Ni B, Dong J, Zhao Y, Yan S. Cnn: Single-label to multi-label. arXiv preprint arXiv:1406.5726 2014.
37. Yu W, Zhang H, He X, Chen X, Xiong L, Qin Z. Aesthetic-based clothing recommendation. In: Proceedings of the 2018 World Wide Web Conference, pp. 649–658. International World Wide Web Conferences Steering Committee 2018.
38. Zhang S, Yao L, Sun A, Tay Y. Deep learning based recommender system: a survey and new perspectives. ACM Comput Surv (CSUR). 2019;52(1):5.
39. Zhou J, Albatal R, Gurrin C. Applying visual user interest profiles for recommendation and personalisation. In: International Conference on Multimedia Modeling. Springer, 2016; pp. 361–6.