THE UNIVERSITY OF

MELBOURNE

COMP90050 Advanced Database Systems

Group Report: Recommendation Systems

Group 29

An Overview on Recommendation System

Lingyu Tang
1040295
tanglt@student.unimelb.edu.au
Cheng Sun
900806
sunc4@student.unimelb.edu.au
Hongwei Yin
901012
hongweiy@student.unimelb.edu.au
Yi Wang
860072
ywang20@student.unimelb.edu.au

Abstract

Recommendation Systems are software tools that interact with high dimensional information space, fully map the interaction history of all users and all items and provide personalized recommendations for users. World-leading companies, such as Google, Amazon, and Facebook, are leading the research in this field, as they rely heavily on personalization functionalities. This paper provides an overview of the history and state-of-art technologies on recommendation systems, and comparisons among different methods.

1. Introduction

Recommendation systems are facilities used to predict the user's preference for items, and such tools become one of the core functions of products we are using nowadays. Some well-known companies, such as Amazon (Dacidson et al. 2010), Alibaba (Zhou et al., 2019), YouTube (Linden et al. 2003), Netflix (Gomez-Uribe & Hunt, 2015) and ByteDance (U.S. Patent No. 10,360,230 B2, 2019), are utilizing the recommendation systems to offer better personalized user experience; users may then discover items that they might not notice before but are potentially interested in. In this way, companies could retain customers and gain more profit.

The development of recommendation systems initiated from a simple phenomenon: people often rely on others' suggestions in the decision-making routine. (Ricci et al., 2011). The development of recommendation algorithms and computer facilities enables users to leverage recommendations from daily interactions with people they know to people they never meet but share similar preferences.

With the advent of big data technology, the number of publications in this field has increased exponentially. As academia and industry show great interest in related technologies, the methods to facilitate recommendation systems are frequently updated.

The technologies used in recommendation systems could be classified into two groups:

- *Content-based systems*: This kind of system only investigates into the similar items a user liked before. For example, if a YouTube user liked a video containing unicorn elements, the system would recommend more videos that relate to unicorns.
- *Collaborative Filtering System:* This kind of system relies on the similarity among users. Items are recommended to a user based on other similar users' browsing history.

This paper focuses only on collaborative filtering algorithms, as it is the most successful and efficient technology (Sun et al., 2019; Burke et al., 2011). More specifically, it reviews the development of recommendation systems, from conventional methods to deep learning models. The rest of the paper is organized by focusing on state-of-art techniques in deep learning models, including Auto-encoder based, MLP based, CNN based, RNN based, and Attention-based methods. The motivation and examples are introduced in each method, and the cross-comparisons are addressed among various methods.

2. Conventional Methods

In this section, we introduce some conventional methods of the recommendation system, which are intended to extract the patterns in the user-item matrix without using deep learning models; thus, they belong to shallow models. The order of these methods in this section is based on the time they are widely investigated.

2.1 Memory-based Algorithms

Memory-based algorithms, also known as Neighborhood-based collaborative filtering algorithms, are the most classic and earliest algorithms for collaborative filtering. These algorithms are developed based on the fact that similar users share similar rating behavior patterns, and one single user gives similar ratings to similar items. (Aggarwal, 2016) As the name suggested, a neighbor is an essential indicator in this kind of recommendation system. The user-item interaction history would be exploited to derive the user-user or item-item similarity, and the recommendation would apply different weights according to similarity parameters. (Herlocker et al., 2002; Schclar et al., 2009). Two typical approaches are user- and item-based methods:

1) User-based approaches:

In this case, the system would apply the clusters to users to identify like-minded users to the target user. From these like-minded users, the system may construct a set of items that the target user has not rated and ranked each item based on other users' behaviors. The recommendation would be generated according to the ranking result.

2) Item-based approaches:

Item-based approaches would evaluate preference to similar items using the item that the user has already rated. Therefore, the system selects the set of items that are most identical to the target item; the ranking of the rates would be used for the final recommendation.

Memory-based algorithms are widely adopted in real-world applications; companies such as YouTube, Amazon applied these approaches in their early phase. (Linden et al., 2003; Dacidson et al., 2010) However, the scalability of these algorithms is ineffective, as searching for similar objects (users or items) is time-consuming, especially for companies with a large dataset.

2.2 Latent Factor Models

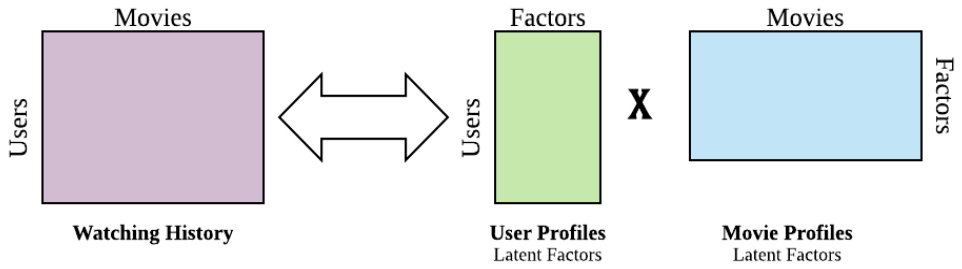

Figure 1. Latent Factor Model Example

The recommendation system intends to use the user-item matrix and recommend items that the target user might enjoy. Unlike memory-based algorithms, which require calculating all distances between users or items and finding nearest neighbors, latent factor models decompose the user-item matrix into a low-dimensional item and user matrices, thus providing better scalability. The Latent Factor Model was developed from matrix factorization (MF) techniques (Koren et al., 2009); it believes that users and items could be characterized by some factors (Jallouli et al., 2017).

The latent factor model has dominated the state-of-art recommendation methods, due to the high efficiency (Sun et al., 2019) and high predictive accuracy when dealing with datasets with huge sparsity (Rendel, 2010). As shown in Figure 1, the user-movie matrix has been factorized by factors, and some example factors could be gender, the categories of movies, or others.

Many pieces of research fall into this category to optimize performance. Probabilistic Matrix Factorization (PMF) firstly introduced the probabilistic linear model and probability density function in this field. (Salakhutdinov & Mnih, 2008) It has been further extended by utilizing side information to improve performance. (Adam et al., 2010) Other approaches are developing this kind of approach to better incorporate several different concerns, including non-negative matrix factorization (Lee & Seung, 2001), Bayesian matrix factorization (Proteous et al., 2010), factorization machine (Rendle, 2010) and others.

2.3 Representation learning models

In the traditional solution of the recommendation system, the items are well-formatted as structured data, as the conventional machine learning algorithms rely heavily on the data representation. However, in reality, the well-formatted data is not enough to satisfy the need of the recommendation system; new features need to be added or created to provide more features for machine learning models.

The earliest representation learning algorithms could be traced back to 1901, K. Pearson (1901) proposed the principal component analysis (PCA), which could project the data in high dimensions to lower dimensions. This method is using the original data and trying to generate new data from it. However, at that time, PCA had not been classified into the new field.

Representation learning has been confirmed as a new field to discover from 2000, as researchers started to investigate the intrinsic structure of high dimensional data. (Zhong et al., 2017) Researches in this field are motivated due to the proven efficiency in capturing local item relationships. As a result of this, words, image information could be well transformed into structural data recently. Some breakthrough results have been achieved in some areas such as Natural Language Processing, Computer Vision. Techniques, including Word2Vec (Mikov et al., 2013; Rong, 2016), Item2Vec (Barkan & Koenigstein 2016), are developed by academic and industry researchers. (Bengio et al., 2014)

2.4 Discussion

Memory-based algorithms are the most traditional algorithms in the recommendation system. It only uses the user-item matrix to predict what other items that the user might enjoy. However, since it needs to calculate all neighbors to find out the nearest neighbors, this algorithm only works well for small companies with small numbers of data, if they want to generate recommendations in real time.

To improve scalability, the Latent Factor Models are introduced; it adds some extra factors and decomposes the user-item matrix to simpler matrices. However, the latent factor models required a good set of features for decomposition; sometimes, this is hard to achieve. Therefore, the representation learning models were developed to create more features for recommendation system algorithms to learn from. It will only be applied if the features for recommendation systems are proven to be insufficient.

These three models show the development of algorithms in recommendation systems. Some researches argued that latent factor models and representation learning models could be treated as one-layer deep learning models. (Sun et al., 2019) Deep learning models could not be well developed without the research on these conventional models, as they provide a solid foundation for some principles of deep learning models.

3. Deep Learning Methods

3.1 Autoencoder

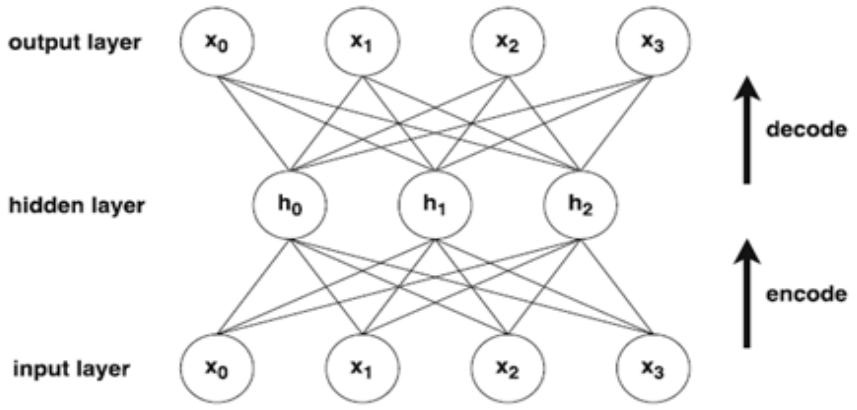

Figure 2. An Autoencoder (Batmaz1 et al., 2018)

Autoencoder (AE) is a neural network machine learning model that reconstructs the input data. It uses an "encoder" network to encode the data into a low-dimensional code and a "decoder" network to recover the data from the code (Hinton & Salakhutdinov, 2006). The structure of AE is shown in figure 2.

Autoencoder has many variants. We will discuss the four most representative examples of them. Sparse autoencoders (SAE) choose the Top-k largest hidden units and set others to 0 to avoid the output layer copying the input layer (Makhzani & Frey, 2013). Denoising autoencoders (DAE) introduce some noise to create corrupted data from original data and use it to train the autoencoder network to achieve the same aim. Stacked denoising autoencoder (SDAE) is an extension of DAE that includes multiple DAE as layers; each of layers uses the output from previous as input (Vincent et al., 2010). Contractive autoencoder (CAE) is similar to DAE in the way that it also uses regularization, but the regularization term is calculated using Frobenius norm of the Jacobian matrix with respect to the input (Rifai et al., 2011). Almost all the autoencoder variants can be applied to the recommendation system. They are generally used to learn lower-dimensional feature representations from the middle-most layer or fill the interaction matrix in the reconstruction layer (Zhang et al., 2019).

Autoencoder can be employed with collaborative filtering (CF) to achieve a better prediction. AutoRec is an example of this kind of implementation. In rating-based CF, each item or user can be represented by a partially observed vector. AutoRec takes the vectors as input and reconstructs it using autoencoders to predict missing ratings for recommendations (Sedhain et al., 2015). Due to the two types of input, there are two variants of AutoRec: I-AutoRec (item based AutoRec) and U-AutoRec (user-based Auto Rec). One extension of AutoRec is CFN, which implements the denoising techniques and includes side information into the network to make the model more scalable and robust (Strub et al., 2016). Besides AutoRec and CFN, Autoencoder-based Collaborative Filtering (Ouyang et al., 2014), Collaborative Denoising Auto-Encoder (Wu et al., 2016), Muli-VAE and Multi-DAE (Liang et al., 2018) are also autoencoder based CF model.

Another popular implementation of the autoencoder is learning feature representation from content features from user or item; there are many approaches using autoencoder. For example, Wang, Wang et al. (2015) suggest a hierarchical Bayesian model called Collaborative Deep Learning (CDL) to improve the performance of collaborative filtering in sparse ratings(feedback) data. CDL allows the two-way interaction between presentation learning for the content information. Also, the collaborative filtering for the rating matrix combines a probabilistic interpretation of ordinal SDAE with probabilistic matrix factorization (PMF) (Zhang et al., 2019). This structure enables the CDL to balance the influence of rating and side information. Relational stacked denoising autoencoders (RSDAE) (Wang, Shi, et al., 2015) and Collaborative Deep Ranking (CDR) (Ying et al., 2016) also use autoencoders for features extraction.

3.2 Multilayer perceptron (MLP)

Multilayer perceptron (MLP) with one or more hidden layers and non-linear activations functions can approximate any measurable function to any desired degree of accuracy (Hornik et al., 1989). It can also be used to add the non-linear transformation to recommendation system algorithms (Zhang et al., 2019).

MLP can be used to construct a dual neural network that models the user-items interaction. Neural Network Matrix Factorization (NNMF) is a representative work of this kind of model. NNMF uses matrix factorization techniques, but instead of using the inner product of latent features vectors to approximate the entries of the matrix, it uses an MLP neural network. It learns by alternating between optimizing the network for latent features (Dziugaite & Roy,2015). Neural Collaborative Filtering (He et al., 2017), Deep Factorization Machine (Guo et al., 2017), and their variants and extensions also fall into this category.

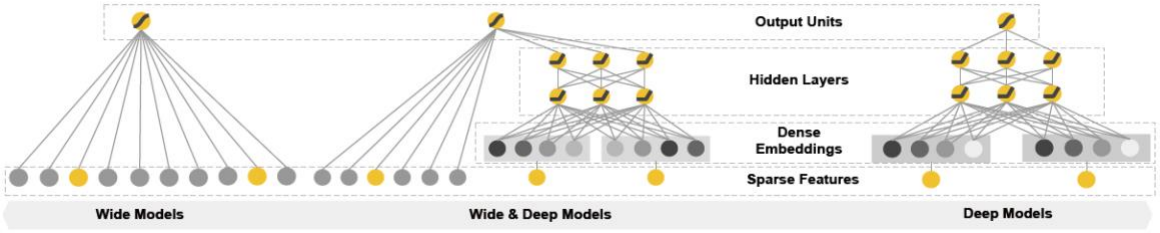

Figure 3. The spectrum of Wide & Deep models. (Cheng et al.,2016)

Some models use MLP to learn the feature representation, even though it may not be as expressive as CNN, RNN, and autoencoder (Zhang et al., 2019). One famous model is Wide & Deep learning (Figure 3), which was introduced for App recommendation in Google play and hugely increased the app acquisitions (Cheng et al.,2016). This framework includes the wide components (generalized linear model) and the deep components (feed-forward neural network). The two components are trained jointly by combining their outcome of the two models using a weighted sum of their output odds and feeding it to one common logistic loss. This joint training optimizes all parameters by taking both components into account at training time and combines the benefits of memorization and generalization for recommendation systems.

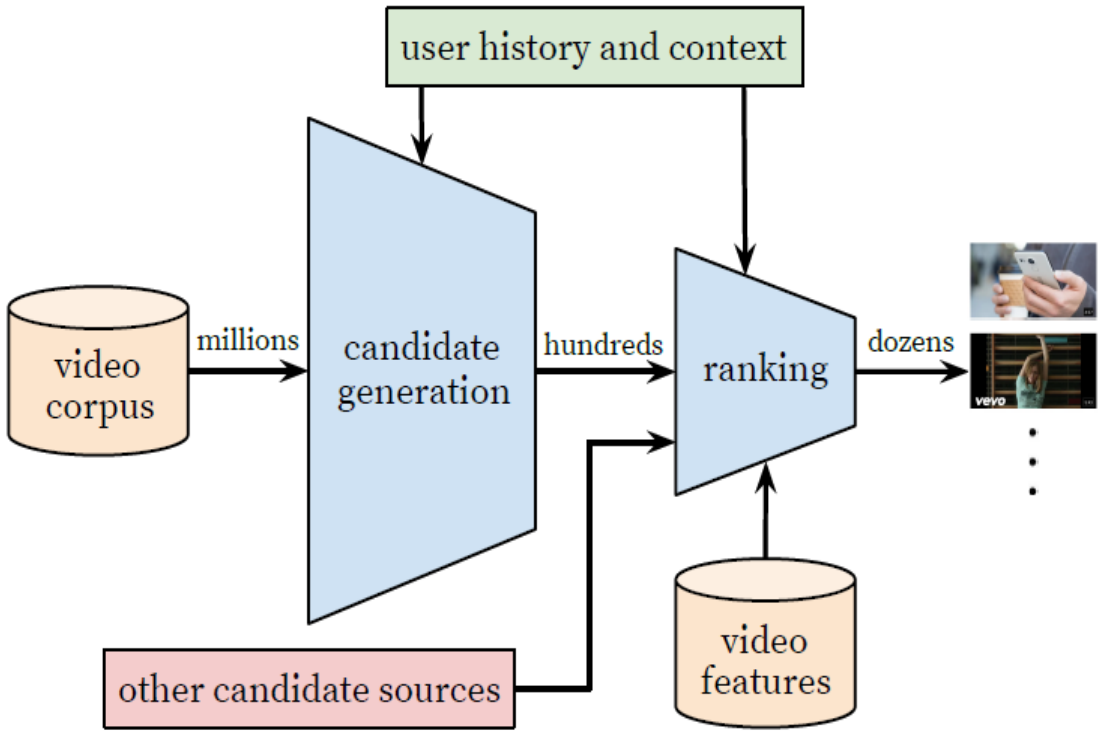

Figure 4. The recommendation system architecture of YouTube (Covington et al., 2016)

The old recommendation system of YouTube uses MLP in a similar way (Covington et al., 2016). This system contains two MLP based models: a deep candidate generation model and a separate deep ranking model. Candidate generation models produce a subset of hundreds from all videos that may be relative to the user. The ranking model specialized and calibrated the list to create personalized recommendation lists for users according to the score of the nearest neighbors from the candidates. The system overview is shown in Figure 4.

3.3 CNN based Methods

Convolutional Neural Network consists of convolutional layers, pooling layers, fully connected layers and input and output layers with a fixed size and considers the input as an image to extract more information from it. Convolutional Neural Network is a powerful approach for information extraction from raw data such as text, image, video, and audio. Moreover, it is often used with other techniques in combination to make recommendations for users.

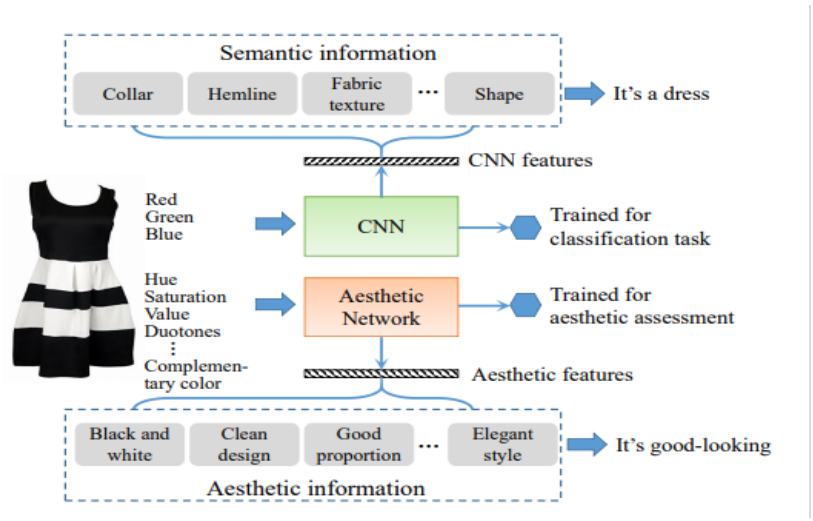

Figure 5. Comparison between CNN network and Aesthetic network in which the former trained for classification task and the latter trained for aesthetic assessment. (Yu et al., 2018)

Yu et al. (2018) proposed a coupled network combining CNN and aesthetic networks to make clothing recommendations, in which CNN plays a feature extraction and classification role, as shown in Figure 5. Moreover, Lei et al. (2016) introduced an approach named Comparative Deep Learning for image recommendation through a deep network consisting of two CNNs as sub-networks. This approach's insight is to map positive, negative images and user preferences into a latent space and train the model on these vectorized features.

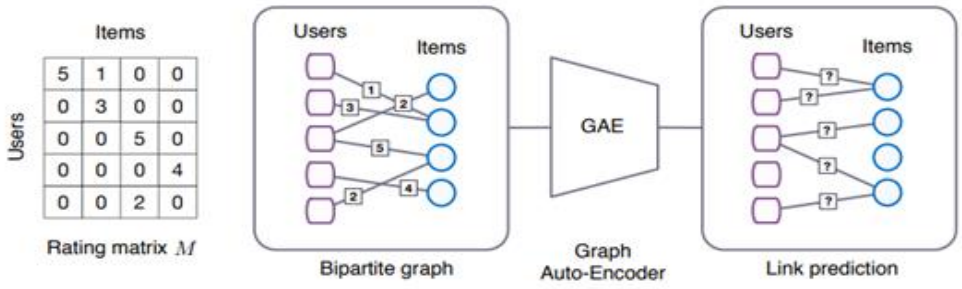

Figure 6. User-item interaction graph with bipartite structure. (Berg et al., 2018)

Furthermore, as graph convolutional networks became popular in deep learning in recent years, Berg et al. (2018) viewed items and users as nodes and ratings and purchases as edges. Thus, they proposed a new solution to the recommendation system with the use of a bipartite interaction graph. Figure 6 indicates that the framework "contains a graph convolutional layer that constructs user and item embeddings through message passing on the bipartite user-item interaction graph. Combined with a bilinear decoder, new ratings are predicted in the form of labeled edges." (Berg et al., 2018)

3.4 RNN based methods

Since RNNs are extremely effective and suitable for processing sequential data, it is natural for researchers to choose RNN as an approach to deal with session-based recommendations for e-commerce websites (Zhang et al., 2018). The insight of such an approach is to extract information from website sessions; hence, to predict users' future behaviors and make recommendations for next items. For example, a user is more likely to view other similar products after browsing the product description and view related peripheral equipment after adding a product to the cart (Liu et al., 2018).

Model Type (GRU Size)	Recall@20	MRR@20
S-POP (-) [10]	0.2672	0.1775
Item-KNN (-) [10]	0.5065	0.2048
TOP1 (1000) [10]	0.6206	0.2693
BPR (1000) [10]	0.6322	0.2467
M4 (1000)	0.6676	0.2847
M2 (100)	**0.7129**	**0.3091**

Figure 7. Comparison of recall between model M2 and other benchmarks. (Tan et al., 2016)

Tan et al. (2016) introduced two methods, data augmentation technique and adaption of temporal changes, to enhance the performance of session based RNN recommendation systems. New training sequences are generated along with corresponding labels given an input training session. Embedding dropout is also adopted to reduce noisy clicks in the training data. Furthermore, Tan et al. (2016) found a way to prevent the recommendation system predicting some out-of-date items. A model is trained on the whole dataset to initialize a new model, which will be fed only with a more recent subset of the dataset. Figure 7 reveals that the approach proposed by Tan et al. (2016) outperforms other existing models.

Session-based RNN provides a solution for the cold-start problem in the recommendation system by providing new users with no history, a considerably ideal recommendation through website sessions processing. However, focusing only on users' short-term preferences may not always generate the best recommendations when user identifiers are present. The following researches emphasize both the long-term and short-term interests of users, hence increasing the performance of existing session-based recommendation models.

Liu et al. (2018) proposed a new model named Short-Term Attention/Memory Priority (STAMP). It introduced a recent action priority mechanism into the session-based recommendation system, which considers both the user's long-term and short-term interests simultaneously (Liu et al., 2018). In STAMP, the user's long-term preferences are captured by historical clicks in a session while the last click is considered as the user's short-term interests. A novel finding is that the user's next action is most likely affected by the user's last click (Liu et al., 2018).

Furthermore, Zhu et al. introduced another method named Time-LSTM to model users' sequential actions to capture their long-term and short-term interests in the recommendation system (Zhu et

al., 2016). Three versions of Time-LSTM are designed in which time intervals between consecutive actions by users are modeled by time gates to improve session-based recommendation systems' performance.

Moreover, RNN is also capable of making recommendations of restaurants and stores based on previous user reviews. David and Gurbir (2016) proposed an approach by employing a multi-stack bi-directional RNN to make recommendations based on a review text data set from the Yelp Dataset Challenge website, which outperforms other approaches mentioned in the research. Furthermore, recent research conducted by Jiang et al. found that RNN could also be applied to music recognition and comparison and, hence, give recommendations based on three factors: "1. Music and songs are sequential data. 2. Lyrics itself is made of language. 3. The genre may contain meanings." (Jiang et al., 2017).

3.5 Attention-based methods

3.5.1 Motivation and ideas

Attention-based methods have recently gained popularity in the field of recommendation systems. They originate from Neural Machine Translation (NMT) field (Zhou et al., 2019) and have further advanced existing neural networks like CNN (Convolutional Neural Network) and RNN (Recurrent Neural Network) (Sun et al., 2019). These relatively new methods have become an indispensable part of transduction models and compelling sequence modeling in various tasks, enabling modeling of dependencies irrespective of distances among input or output sequences (Vaswani et al., 2017). Attention-based methods mainly aim to deal with noisy data problems and their side effects using a user-item interactive model based on relevance identified in input; specifically, it may help distinguish the importance level of raw inputs, filtering out the uninformative features and selecting the most representative items (Sun et al., 2019; Zhang et al., 2019). Some of the representative methods are DeepMove and AttRec (Sun et al., 2019). The fundamental idea of attention-based methods is explained when analyzing the paper "Attention is All You Need" in the later section (Vaswani et al., 2017).

3.5.2 Categorization of the approaches and Comparison among them

Attention-based models analyze input with attention score algorithms (Zhang et al., 2019). Based on the way the attention scores are calculated, one may categorize the attention models as standard vanilla attention and co-attention. While co-attention calculates attention weights from two-sequence, vanilla uses a parameterized context vector to complete it. Besides, self-attention is a special case of co-attention (Zhang et al., 2019).

The vanilla attention-based method calculates normalized attention scores for inputs by transforming data through layers of data processing; specifically, the additional Softmax layer is responsible for normalizing the scores (Sun et al., 2019). An example is an attentive and collaborative filtering model proposed by Chen et al. (2017). The model consists of component- and item-level attention: the item-level attention selects the most representative items that can characterize users; the component-level attention captures the informative features from auxiliary information of each user.

Self-attention/intra-attention is an attention mechanism that computes a representation of a sequence using different positions (Vaswani et al., 2017). Without any additional information, one can still extract relevant aspects from users or items by allowing it to attend to itself using self-attention (Ruder, 2017). In sequence learning, self-attention may achieve better accuracy with lower computation complexity (Sun et al., 2019). It is expected that self-attention may replace many complex neural models like RNN and CNN (Zhang et al., 2019). This mechanism inspires AttRec; as a novel sequence-aware recommendation model, AttRec can consider both long- and short-term user interests (Zhang et al., 2019). By capitalizing on the strength of both metric learning and self-attention, this hybrid model improves the sequential recommendation performance (Zhang et al., 2019). More specifically, self-attention is used to learn the short-term intents of the user from his or her recent interactions; relative weights of individual items in the users' interaction trajectories are estimated; the result is then used to learn better representation for the user transient interests (Sun et al., 2019; Zhang et al., 2019). The metric learning is responsible for learning more expressive user and item embeddings (Zhang et al., 2019).

3.5.3 Hybrid methods

Typically, attention may be combined with RNN and CNN to get better results (Zhang et al., 2019). However, some study groups like Vaswani et al. (2017) have used pure attention-based methods to solve problems; their research details will be explained later.

Although LSTM (Long Short-Term Memory) can be used to deal with long memory problems, integrating attention mechanisms into RNNs enables it to handle noisy and long inputs with long-range dependencies more efficiently; this hybrid mechanism helps the network to memorize inputs better (Zhang et al., 2019). For example, Feng et al. (2018) designed DeepMove based on the hybrid method for user mobility prediction. The RNN in DeepMove is responsible for capturing the sequential transitions of the current trajectory; they then designed a historical attention model to capture the mobility regularity based on lengthy historical records.

Attention-based CNNs, on the other hand, can help capture the most informative elements of the inputs (Zhang et al., 2019). For instance, Seo et al. (2017) developed D-Attn to learn item and user representations using global (G-Attn) local (L-Attn) attention. The model uses the embeddings of words in reviews as input and adopts G-Attn and L-Attn to learn the saliency of words with respect to the entire input texts and a local window. The results are passed to two CNNs to learn the L-Attn and G-Attn representations of a user/item. Finally, the model uses the inner product of item and user representations to estimate a user's preference for an item.

3.5.4 Self-attention Example: Attention Is All You Need

According to Vaswani et al. (2017), in traditional methods, the encoder in RNN (Recurrent neural network) predicts the hidden state of a word using its word vectors and the hidden state of the previous word. The first word's hidden state is then processed by the decoder to output a translated word and a new hidden state. The output, i.e., the hidden state and the translated word, will be used in the next decoding process to aid translation. In this case, to handle long-range dependencies, many repetitive transformations need to be processed; all hidden states and words need to be memorized during the process (Vaswani et al., 2017).

In attention-based methods, the decoder may learn information directly from the input sentence instead of going to their hidden states (Vaswani et al., 2017). In doing so, the decoder can access the hidden states that may help in the translation process directly; therefore, the process is shorter than circling through all the hidden states traditionally; it also reduces the number of computation steps and information transaction in the network, thereby minimizing information loss (Vaswani et al., 2017).

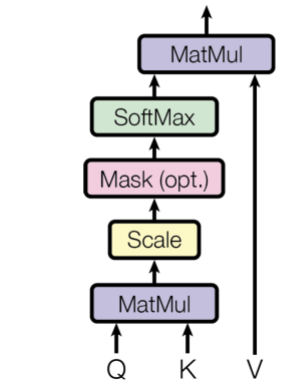

Figure 8. Scaled Dot-Product Attention (Vaswani et al., 2017)

The Scaled Dot-Product Attention in their paper is the principle of the Self-Attention mechanism (Vaswani et al., 2017). The responsibility of attention function is to map a set of key-value pairs and a query to output; the output, keys, values, and query are all vectors. As shown in figure 8, the weight of each value is computed by a set of compatibility functions of the corresponding key and the query; the result is then combined with the weighted sum of values to generate an output.

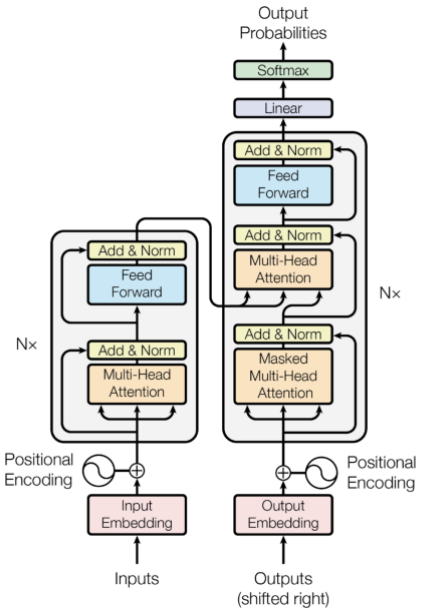

Figure 9. The Transformer-model architecture

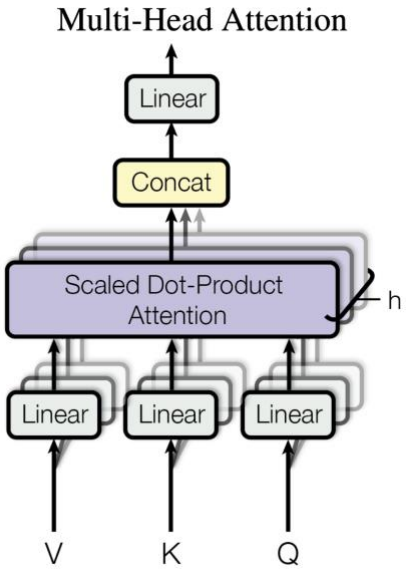

Figure 10. Multi-Head Attention (Vaswani et al., 2017)

As shown in figure 9, the source sentence goes to the encoder (the block on the left), and the target sentence goes to the decoder (the block on the right); their positional information is also passed as input; they are then combined to get an output probability for the next word (Vaswani et al., 2017).

Specifically, the Multi-Head Attention in figure 10 receives three types of values from the source and target sentences: the key-value pairs are outputs of the encoding part of the source sentence, and the query is that of the encoded target sentence. Values are interesting attributes of the source sentence indexed by unique keys. The job of the Multi-Head Attention is to find the most interesting key-value pair that the query should pay attention to.

Compared with the sequential processing of RNN by back-propagating through all encoding and decoding processes, every step in this attention-based method is one independent training sample; as there is no multi-step back-propagation and recurrent steps, the computation complexity is significantly lower compared to that of the traditional models (Vaswani et al., 2017).

3.5.5 Attention-based methods examples: DIN & DSIN

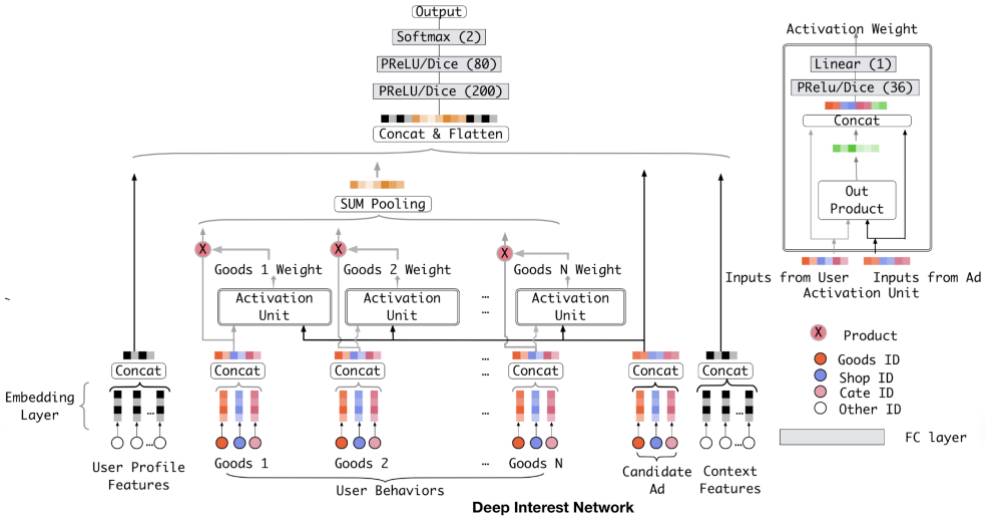

Figure 11. Deep Interest Network (Zhou et al., 2019)

Alibaba's Deep Interest Network (DIN) utilized the attention-based method to capture users' dynamic and evolving interests from behavior sequences, thereby increasing CTR (Zhou et al., 2019). The local activation unit shown in figure 11 takes a weighted sum pooling based on the soft-search result of user behaviors; the result may then be used to obtain the adaptive representation of user interests concerning a given advertisement; data loss during the process is less than that of traditional methods. Also, the reason they have achieved a higher CTR is that the user representation vector varies over different advertisements; this is different from traditional methods in which there is no interaction between user and advertisement. That said, different user behaviors receive different attention or weight according to the relevance between individual historic user behaviors and the characteristics of advertisements. Test results also proved that, by exploiting the relationship between target item and users' historical behaviors, DIN helps to capture the primary intent of queries.

Moreover, Alibaba's proposed a novel CTR model, Deep Session Interest Network (DSIN), based on their observation on the intrinsic structure of users' behavior sequences (Feng et al., 2019). They found that behavior sequences are composed of sessions; sessions are user behaviors separated by their occurring time. User behaviors in each session are highly homogeneous, and heterogeneous cross sessions. Motivated by this observation, they proposed DSIN that leverages users' multiple historical sessions in their behavior sequences. Firstly, they divide users' sequential behaviors into sessions and use the self-attention mechanism with bias encoding to learn users' interests in each session. Then Bi-LSTM is used to model how users' interests interact and evolve among sessions. Finally, it again used the local activation unit to learn the influences of various session interests on the target item.

Model	Advertising	Recommender
YoutubeNet-NO-UB[a]	0.6239	0.6419
YoutubeNet	0.6313	0.6425
DIN-RNN	0.6319	0.6435
Wide&Deep	0.6326	0.6432
DIN	0.6330	0.6459
DIEN	0.6343	0.6473
DSIN-PE[b]	0.6357	0.6494
DSIN-BE-NO-SIIL[c]	0.6365	0.6499
DSIN-BE[d]	**0.6375**	**0.6515**

Figure 12. Results on the advertising and recommender Datasets (Feng et al., 2019)

The table shown in figure 12 lists results on the advertising datasets and recommendation datasets (Feng et al., 2019). It shows that DSIN outperforms other state-of-the-art modes, including DIN, on both datasets. The comparison between DIN and DSIN also illustrate that, depending on the characteristics of the recommendation system, models like self-attention and LSTM may be combined to achieve better results. While LSTM brings no improvement to DIN, Bi-LSTM, as one of the RNN models that is excellent at dealing with sequential patterns, was added in the DSIN to capture the sequential relation of contextual session interest (Zhou et al., 2019; Feng et al., 2019). Plus, both DIN and DSIN fed data into layers of MLP in the final stage for better information extraction. Therefore, sometimes self-attention is not all one needs.

3.6 Conclusion of Deep Learning methods

3.6.1 Autoencoder and MLP

Autoencoder is suitable for learning feature representation, reducing the dimensionality of the data and making predictions in recommendation systems, and they are mostly used to solve sparsity and scalability problems (Batmaz et al., 2018). While using MLP to extract features from data may not be as expressive as autoencoder, CNN and RNN, many systems use it for classification and prediction for its ability to approximate nonlinear functions (Zhang et al., 2019).

3.6.2 CNN and RNN

Recent researches on recommendation systems employ convolutional neural networks as an information extraction tool, combining them with other advanced techniques to propose an approach to recommendation systems due to its remarkable performance on feature engineering. Furthermore, as Zhang et al. (2019) stated, graph convolutional networks have novel performances on non-Euclidean data, such as social networks. It is believed that the current role of CNN in recommendation systems will be further reinforced in future studies.

As RNNs perform well on sequential data processing, such as extracting information from audio and video features, researchers start to utilize RNN in session-based recommendation systems in e-commerce, where sequential patterns of user behaviors are emphasized. Short-term interests of users are captured through web sessions, while long-term interests are captured from users' historical data. Moreover, RNNs are also employed in many other domains, such as music recommendation and restaurant recommendation, as information could be learned from lyrics and reviews through Natural Language Processing.

3.6.3 Attention-based methods

As an intuitive but effective technique, attention mechanism has garnered attention across several areas like speech recognition, computer vision, natural language processing, and recommendation systems (Zhang et al., 2019). In its early stages, attention mechanisms can be combined with mature models like CNN or RNN to achieve better results. Researchers later managed to achieve better performance using self-attention (Vaswani et al., 2017). In fact, self-attention can potentially replace CNN and RNN in many fields (Zhang et al., 2019). Self-attention's first advantage is that it has a less complicated structure; this means that they require less input and fewer processing steps, thus enjoying lower computation complexity. Secondly, experiments have shown that those models are more parallelizable and require less time to train (Vaswani et al., 2017). Compared to RNN, attention-based methods may also handle long-range dependencies more effectively, reducing data loss and increasing accuracy. That said, attention-based methods are not a panacea for all problems; one may choose appropriate models based on the focus of the recommendation task and data processing characteristics of that recommendation system.

In general, each deep learning approach has its advantage; for example, autoencoder reduces the dimensionality of given data, RNN performs better in session-based recommendation systems, and CNN extracts more exclusive information from raw data. In terms of attention-based methods, they commonly have a lower computation complexity and can be an ideal alternative to CNN and RNN in many fields.

4. Comparison between conventional and deep learning methods

Although conventional recommendation systems can produce decent recommendations, they are gradually replaced by deep learning-based methods in the industries. This trend can be easily observed from the big companies which need to process a large amount of data such as YouTube, whose recommendation system evolved from conventional (Davidson et al., 2010) to deep learning-based (Covington et al., 2016). Deep learning models have four advantages compare to the conventional models (Batmaz et al., 2018):

- Nonlinear Transformation: Neural networks are capable of modeling the nonlinear relationship in data so that these models can capture the complex interaction between users and items.
- Representation Learning: Deep learning models are efficient in learning the features representations, enabling the recommendation systems to include information from different sources automatically.

- Sequence Modelling: Deep learning models are proven to be able to handle sequential modeling tasks such as natural language understanding
- Flexibility: The proper modularization of deep learning techniques makes it easy to build hybrid and composite recommendation models.

Due to these advantages, deep learning models can produce practical solutions for the accuracy, sparsity, scalability, and cold-start problems of conventional models, making it popular among the big companies with enough resources. However, neural network models lack interpretability, making it hard to analyze the system's feature interactions. Also, deep learning models have requirements for large scale data and extensive hyperparameter tuning, making it not suitable for the recommendation systems for a small number of users and items (Zhang et al., 2018). So, sometimes, especially for small companies with a restriction on the cost and a small number of users, a conventional recommendation system may be a more practical choice.

5. Conclusion

In this paper, we review researches on recommendation systems, which includes conventional methods and deep learning. The trend of replacing deep learning-based approaches over traditional methods is inevitable, as it is the consequence of the recommendation system's development. The emergence of neural networks proposes an ideal solution to Non-linear transformation, sequence modeling, and representation learning in recommendation systems.

In this article, an extensive review of the development and current state-of-art techniques of recommendation systems is provided. It reveals a trend of more and more employment of deep learning in this domain. Although recent researches on deep learning in recommendation systems have achieved a certain degree of success, most proposed approaches employ deep learning only as a method of feature representation learning. Furthermore, the gap between theories and applications in this domain is still quite large. Therefore, the utilization of these approaches to tackle real-world problems should also be considered in future studies. For example, one may make deep learning more scalable and flexible to fit small companies' recommendation systems.

Finally, we emphasize that deep learning is not the only method in recommendation systems. To solve problems in the real world, more sophisticated models and solutions to reduce expenditure in recommendation systems are indispensable.

6. Reference

Adams, R. P., Dahl, G. E., & Murray, I. (2010). Incorporating Side Information in Probabilistic Matrix Factorization with Gaussian Processes. ArXiv:1003.4944 [Cs, Stat]. http://arxiv.org/abs/1003.4944

Aggarwal, C. C. (2016). Neighborhood-based collaborative filtering. In *Recommender systems* (pp. 29-70). Springer, Cham.

Batmaz, Z., Yurekli, A., Bilge, A., & Kaleli, C. (2019). A review on deep learning for recommender systems: challenges and remedies. *Artificial Intelligence Review, 52(1), 1-37.* doi: https://doi.org/10.1007/s10462-018-9654-y

Beijing ByteDance Network Technology Co., LTD., Beijing (CN) (2019) U.S. Patent No. 10,360,230 B2. Washington, DC: U.S. Patent and Trademark Office.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8),* 1798–1828. https://doi.org/10.1109/TPAMI.2013.50

Berg, R. van den, Kipf, T. N., & Welling, M. (2017). Graph Convolutional Matrix Completion. ArXiv:1706.02263 [Cs, Stat]. http://arxiv.org/abs/1706.02263

Burke, R., Felfernig, A. & Goker, M.H. (2011). Recommender Systems: An Overview. 6.

Carlos A. Gomes-Uribe and Neil Hunt, 2015. The Netflix recommender system: Algorithms, business value, and Innovation. *ACM Transactions on Management Information Systems, 6(4),* 1–19. https://doi.org/10.1145/2843948

Cheng, H.-T., Ispir, M., Anil, R., Haque, Z., Hong, L., Jain, V., Liu, X., Shah, H., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., & Chai, W. (2016). Wide & Deep Learning for Recommender Systems. *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems - DLRS 2016,* 7–10. https://doi.org/10.1145/2988450.2988454

Covington, P., Adams, J., & Sargin, E. (2016). Deep Neural Networks for YouTube Recommendations. *Proceedings of the 10th ACM Conference on Recommender Systems,* 191–198. https://doi.org/10.1145/2959100.2959190

Chen, J., Zhang, H., He, X., Nie, L., Liu, W., & Chua, T.-S. (2017). 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, 335–344. https://doi.org/10.1145/3077136.3080797

Davidson, J., Livingston, B., Sampath, D., Liebald, B., Liu, J., Nandy, P., Van Vleet, T., Gargi, U., Gupta, S., He, Y., & Lambert, M. (2010). The YouTube video recommendation system. *Proceedings of the Fourth ACM Conference on Recommender Systems - RecSys '10,* 293. https://doi.org/10.1145/1864708.1864770

Dziugaite, G. K., & Roy, D. M. (2015). Neural Network Matrix Factorization. ArXiv:1511.06443 [Cs, Stat]. http://arxiv.org/abs/1511.06443

Fisher, R. A. (1936). The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics, 7(2),* 179–188. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x

Feng, J., Li, Y., Zhang, C., Sun, F., Meng, F., Guo, A., & Jin, D. (2018). DeepMove: Predicting Human Mobility with Attentional Recurrent Networks. Proceedings of the 2018 World Wide Web Conference, 1459–1468. https://doi.org/10.1145/3178876.3186058

Feng, Y., Lv, F., Shen, W., Wang, M., Sun, F., Zhu, Y., & Yang, K. (2019). Deep Session Interest Network for Click-Through Rate Prediction. ArXiv:1905.06482 [Cs]. http://arxiv.org/abs/1905.06482

Guo, H., Tang, R., Ye, Y., Li, Z., & He, X. (2017). DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. ArXiv:1703.04247 [Cs]. http://arxiv.org/abs/1703.04247

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural Collaborative Filtering. *Proceedings of the 26th International Conference on World Wide Web*, 173–182. https://doi.org/10.1145/3038912.3052569

Herlocker, J., Konstan, J. A., & Riedl, J. (2002). An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms. Information Retrieval, 5(4), 287–310. https://doi.org/10.1023/A:1020443909834

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the Dimensionality of Data with Neural Networks. Science, 313(5786), 504–507. https://doi.org/10.1126/science.1127647

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. Neural Networks, 2(5), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

Jallouli, M., Lajmi, S., & Amous, I. (2017). Latent Factor Model Applied to Recommender System: Realization, Steps and Algorithm. In M. Themistocleous & V. Morabito (Eds.), Information Systems (pp. 606–618). Springer International Publishing. https://doi.org/10.1007/978-3-319-65930-5_47

Jiang, M., Yang, Z., & Zhao, C. (2017). What to play next? A RNN-based music recommendation system. 2017 51st Asilomar Conference on Signals, Systems, and Computers, 356–358. https://doi.org/10.1109/ACSSC.2017.8335200

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. Computer, 42(8), 30–37. https://doi.org/10.1109/MC.2009.263

Lee, D. D., & Seung, H. S. (2001). Algorithms for Non-negative Matrix Factorization. In T. K. Leen, T. G. Dietterich, & V. Tresp (Eds.), Advances in Neural Information Processing Systems 13 (pp. 556–562). MIT Press. http://papers.nips.cc/paper/1861-algorithms-for-non-negative-matrix-factorization.pdf

Lei, C., Liu, D., Li, W., Zha, Z.-J., & Li, H. (2016). Comparative Deep Learning of Hybrid Representations for Image Recommendations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2545–2553. https://openaccess.thecvf.com/content_cvpr_2016/html/Lei_Comparative_Deep_Learning_CVPR_2016_paper.html

Liang, D., Krishnan, R. G., Hoffman, M. D., & Jebara, T. (2018). Variational Autoencoders for Collaborative Filtering. Proceedings of the 2018 World Wide Web Conference. (pp. 689-698). Retrieved July 24, 2020, from https://dl.acm.org/doi/abs/10.1145/3178876.3186150

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 7(1), 76–80. https://doi.org/10.1109/MIC.2003.1167344

Liu, D. Z., & Singh, G. (2016). A recurrent neural network based recommendation system. tech. rep., Stanford University.

Liu, Q., Zeng, Y., Mokhosi, R., & Zhang, H. (2018). STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 1831–1839. https://doi.org/10.1145/3219819.3219950

Makhzani, A., & Frey, B. (2013). [1312.5663] k-Sparse Autoencoders. (n.d.). Retrieved July 21, 2020, from https://arxiv.org/abs/1312.5663

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 26 (pp. 3111–3119). Curran Associates, Inc. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

Ouyang, Y., Liu, W., Rong, W., & Xiong, Z. (2014). Autoencoder-Based Collaborative Filtering. In C. K. Loo, K. S. Yap, K. W. Wong, A. T. Beng Jin, & K. Huang (Eds.), Neural Information Processing (pp. 284–291). Springer International Publishing. https://doi.org/10.1007/978-3-319-12643-2_35

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space: The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science: Vol 2, No 11. (n.d.). Retrieved July 24, 2020, from https://www.tandfonline.com/doi/pdf/10.1080/14786440109462720

Porteous, I., Asuncion, A., & Welling, M. (2010, July 3). Bayesian Matrix Factorization with Side Information and Dirichlet Process Mixtures. Twenty-Fourth AAAI Conference on Artificial Intelligence. https://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/view/1871

Rendle, S. (2010). Factorization Machines. 2010 IEEE International Conference on Data Mining, 995–1000. https://doi.org/10.1109/ICDM.2010.127

Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to Recommender Systems Handbook. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), Recommender Systems Handbook (pp. 1–35). Springer US. https://doi.org/10.1007/978-0-387-85820-3_1

Rifai, S., Vincent, P., Muller, X., Glorot, X., & Bengio, Y. (2011, January 1). Contractive Auto-Encoders: Explicit Invariance During Feature Extraction. ICML. https://openreview.net/forum?id=HkZN5j-dZH

Rong, X. (2016). Word2vec Parameter Learning Explained. ArXiv:1411.2738 [Cs]. http://arxiv.org/abs/1411.2738

Ruder, S. (2017). Deep Learning for NLP Best Practices. Ruder. Retrieved July 21, 2020, from http://ruder.io/deep-learning-nlp-best-practices/index.html#bestpractices.

Salakhutdinov, R. & Mnih, A. (2008) Probabilistic Matrix Factorization. In J. C. Platt, D. Koller, Y. Singer, & S. T. Roweis (Eds.), Advances in Neural Information Processing Systems 20 (pp. 1257–1264). Curran Associates, Inc. http://papers.nips.cc/paper/3208-probabilistic-matrix-factorization.pdf

Sedhain, S., Menon, A. K., Sanner, S., & Xie, L. (2015). AutoRec: Autoencoders Meet Collaborative Filtering. Proceedings of the 24th International Conference on World Wide Web, 111–112. https://doi.org/10.1145/2740908.2742726

Strub, F., Gaudel, R., & Mary, J. (2016). Hybrid Recommender System based on Autoencoders. Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, 11–16. https://doi.org/10.1145/2988450.2988456

Sun, Z., Guo, Q., Yang, J., Fang, H., Guo, G., Zhang, J., & Burke, R. (2019). Research commentary on recommendations with side information: A survey and research directions. Electronic Commerce Research and Applications, 37, 100879. https://doi.org/10.1016/j.elerap.2019.100879

Seo, S., Huang, J., Yang, H., & Liu, Y. (2017). Interpretable Convolutional Neural Networks with Dual Local and Global Attention for Review Rating Prediction. Proceedings of the Eleventh ACM Conference on Recommender Systems, 297–305. https://doi.org/10.1145/3109859.3109890

Schclar, A., Tsikinovsky, A., Rokach, L., Meisels, A., & Antwarg, L. (2009). Ensemble methods for improving the performance of neighborhood-based collaborative filtering. Proceedings of the Third ACM Conference on Recommender Systems - RecSys '09, 261. https://doi.org/10.1145/1639714.1639763

Tan, Y. K., Xu, X., & Liu, Y. (2016). Improved Recurrent Neural Networks for Session-based Recommendations. Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, 17–22. https://doi.org/10.1145/2988450.2988452

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), Advances in Neural Information Processing Systems 30 (pp. 5998–6008). Curran Associates, Inc. http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P. A., & Bottou, L. (2010). Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. Journal of machine learning research, 11(12).

Wang, H., Shi, X., & Yeung, D.-Y. (2015, February 21). Relational Stacked Denoising Autoencoder for Tag Recommendation. Twenty-Ninth AAAI Conference on Artificial Intelligence. https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9350

Wang, H., Wang, N., & Yeung, D. Y. (2015, August). Collaborative deep learning for recommender systems. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1235-1244). Retrieved July 14, 2020, from https://dl.acm.org/doi/abs/10.1145/2783258.2783273

Wu, Y., DuBois, C., Zheng, A. X., & Ester, M. (2016, February). Collaborative denoising auto-encoders for top-n recommender systems. Proceedings of the Ninth ACM International Conference on Web Search and Data Mining. (pp. 153-162). Retrieved July 13, 2020, from https://dl.acm.org/doi/abs/10.1145/2835776.2835837

Ying, H., Chen, L., Xiong, Y., & Wu, J. (2016). Collaborative Deep Ranking: A Hybrid Pair-Wise Recommendation Algorithm with Implicit Feedback. In J. Bailey, L. Khan, T. Washio, G. Dobbie, J. Z. Huang, & R. Wang (Eds.), Advances in Knowledge Discovery and Data Mining (pp. 555–567). Springer International Publishing. https://doi.org/10.1007/978-3-319-31750-2_44

Yu, W., Zhang, H., He, X., Chen, X., Xiong, L., & Qin, Z. (2018, April). Aesthetic-based clothing recommendation. Proceedings of the 2018 World Wide Web Conference. (pp. 649-658). Retrieved July 21, 2020, from https://dl.acm.org/doi/abs/10.1145/3178876.3186146

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep Learning Based Recommender System: A Survey and New Perspectives. ACM Computing Surveys, 52(1), 5:1–5:38. https://doi.org/10.1145/3285029

Zhong, G., Wang, L., Ling, X. & Dong, J. (2017) An overview on data representation learning: From traditional feature learning to recent deep learning. http://dx.doi.org/10.1016/j.jfds.2017.05.001

Zhou, G., Mou, N., Fan, Y., Pi, Q., Bian, W., Zhou, C., Zhu, X., & Gai, K. (2019). Deep Interest Evolution Network for Click-Through Rate Prediction. Proceedings of the AAAI Conference on Artificial Intelligence, 33(01), 5941–5948. https://doi.org/10.1609/aaai.v33i01.33015941