Recommendation Systems Based on Collaborative Filtering Using Autoencoders: Issues and Opportunities

Ria Banerjee, Preeti Kathiria, and Deepika Shukla

Abstract With the advancement of deep-learning-based approaches, complex problems under Artificial Intelligence can now be addressed in a comparatively easier way. One such application domain is Recommendation Systems. Recommendation systems are powerful tools in this age of data explosion for providing meaningful insights from data. Collaborative Filtering is one of the popular approaches for building recommendation systems and extensive literary works suggest that it is very effective. In recent years deep-learning-based models have been bounteously applied for the development of recommendation systems using collaborative filtering. Autoencoders are a deep-learning based neural architecture which can be used for implementing collaborative filtering. This paper presents a survey of different autoencoder based models which employ collaborative filtering methodology for making recommendation systems. The paper initially provides an understanding of models and thereafter summarizes various works reported in the literature in the light of the methodology used, taxonomy, datasets used for experimentation, limitations and results reported.

Keywords Deep learning · Collaborative filtering · Autoencoders

1 Introduction

In the environment of Big Data and E-commerce, machine learning has widened its scope of applications. Table 1 shows the relationship between E-commerce, Big Data Analytics (BDA) and Neural Networks (NN) as three entities intermixed and

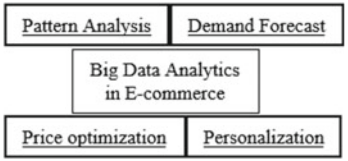
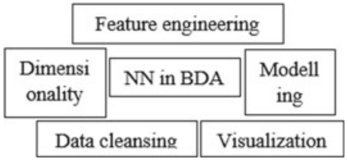
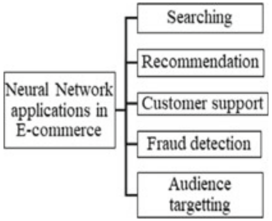
R. Banerjee · P. Kathiria (✉) · D. Shukla (✉)
Nirma University, Ahmedabad, Gujarat, India
e-mail: preeti.kathiria@nirmauni.ac.in

D. Shukla
e-mail: deepika.shukla@nirmauni.ac.in

R. Banerjee
e-mail: 19mca168@nirmauni.ac.in

P. K. Singh et al. (eds.), *Recent Innovations in Computing*, Lecture Notes
in Electrical Engineering 701, https://doi.org/10.1007/978-981-15-8297-4_32

Table 1 Big data analytics in E-commerce

 Pattern Analysis Demand Forecast Big Data Analytics in E-commerce Price optimization Personalization Fig. 1 Companies use BDA to track user patterns to provide price optimization and personalization, thus forecasting demand	 Feature engineering Dimensi onality NN in BDA Modell ing Data cleansing Visualization Fig. 2 NNs are employed to handle manipulations on enormous amount of data	 Neural Network applications in E-commerce Searching Recommendation Customer support Fraud detection Audience targetting Fig. 3 NN applications in E-commerce activities

interdependent, with the goal of creating a profitable domain for the E-commerce industry (Figs. 1, 2 and 3).

As the pool of online content widens and deepens, users get a plethora of choices. But this often poses ‘paradox of choice’. Recommendation systems are expert systems that harness the power of data to enhance the decision-making process for the user.

Recommendation systems gather information in three ways: *Implicitly, Explicitly* [1] and by *Hybrid* methods. In *Implicit* feedback, user behaviour is captured from the actions made by the user while accessing the application. It does not require user intervention. In *Explicit* feedback, information is gathered by explicit user intervention. Hybridization of feedback is achieved in two ways: (i) User is allowed to participate in explicit feedback gathering, and (ii) implicit data as a check on explicit feedback.

Recommendation systems can be broadly classified into Content-based [2] (based on the features extracted from the items that the user has rated positively).

Collaborative filtering based [3, 4] (recommends items to the target users by mapping similar users) which can be further categorized into item-based and user-based

- Item-based: Find the correlation between items based on user’s previous rating.
- User-based: Finds a correlation between target user’s and other users’ profile.

The third major category of recommendation systems is Hybrid [5], which is a combination of both content-based and collaborative filtering (CF).

Major issues in recommender systems are cold start, shilling attacks, synonymy, grey sheep, Limited Content Analysis and Overspecialization, privacy, latency, Evaluation and Availability of Online Datasets, context awareness, sparsity and scalability [6, 7]. However, due to the inherent nature of CF, cold start, sparsity, and shilling attacks are the most prominent problems. The 1990s saw information overload problems [3] as a major challenge in context-based recommender systems. Amazon was one of the earliest commercial applications of CF. Over the years CF algorithms have advanced in complexity and performance. The algorithms under CF are grouped as:

1. Memory-based [8, 9]: they employ statistical techniques in the user-item matrix to find the nearest neighbours of the users and make recommendations.
2. Model-based [10]: Huge datasets are compressed and fed into a model that generate recommendations. Matrix Factorization (MF) and clustering are examples of traditional methods. In the recent past deep-learning-based methods have also gained popularity [11, 12]. Algorithms under deep learning can be of the following type:

   (a) Multi-layered Perceptron (MLP): it is a feed-forward NN architecture with hidden layers, where back-propagation is used for learning.
   (b) Convolutional NN (CNN) [12, 13]: It incorporates convolution (a mathematically represented linear operation) on data projected in a grid.
   (c) Recurrent Neural Networks (RNN) [12, 14]: These NN are good for modelling sequential data. They can remember former computations.

(d) Restricted Boltzmann Machines (RBM) [12, 15]: These generative stochastic NN with 2 layers produce a random probability distribution of the input.

(e) Autoencoders (AE) [16]: AE is an unsupervised NN model attempting to reconstruct its input data in the output layer.

Lots of content has been published where the surveys are provided for deep-learning based recommendation systems that use CNN, RNN, or RBM as the architecture, whereas this paper focuses on providing a survey of literature and approaches that use CF for recommender systems using autoencoders as the architecture.

1.1 Research Challenges and Future Direction

It is established by this study that the benchmark datasets are available for research in the area of recommender systems, however, it is strongly recommended that more datasets pertaining to other fields of recommendation should be generated so that the recommender systems for other fields can also be developed. One striking observation that has come up during this survey is that the literature available for this area lacks in comparative analysis wherein all the models are compared experimentally. The comparative analysis of all models can be carried out experimentally so that better insights can be projected and application-specific choice of model can be suggested.

The rest of the paper is organized as Sect. 2 discusses AEs and their types. Section 3 presents the literature available for the subject of the paper, various modes for evaluating the approaches and datasets used for comparing the approaches in the area. Section 4 throws the light on a comparison between various works reported in the literature for recommendation systems using CF and AEs. Section 5 concludes the study with major insights and throwing open-ended research issues and challenges.

2 Autoencoders

In general, Autoencoders consists mainly of four main parts [16]: Encoder, bottleneck, Decoder, and reconstruction Loss method. The encoder learns the way to input dimensions, compresses the input into an encoded representation. The bottleneck layer is used as a salient feature representation of the input data. The bottleneck layer represents the data in the lowest possible dimensions. The decoder reconstructs the encoded data into a representation that is as close to the input as possible. The reconstruction loss method is responsible for making out the losses incurred during this encode-decode process. Training an AE means minimizing the network's loss using back propagation (Fig. 4).

A vanilla AE is a simple two-layered NN with one hidden layer, where the length of the encoder and decoder is the same, and that of the hidden layer is smaller than

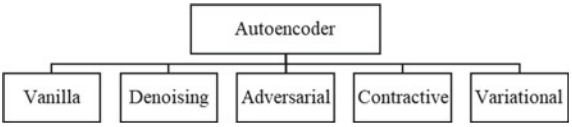


Fig. 4 Types of autoencoders

them. A Denoising AE [17] introduces noise in the input, and then it reconstructs this corrupted data to produce cleaned output. Adversarial AEs consists of a discriminator and a generator. The discriminator generates the probability of a point x belonging to the data distribution, while a generator generates data that is fed to the discriminator intending to fool the discriminator [18]. Contractive AE [19] adds a regularizer to the objective function of the AE. This regularizer belongs to the Frobenius norm of the Jacobian mix. Variational AE maps the input to a distribution, where the bottleneck consists of a mean vector and a standard deviation vector.

3 Literature Survey

The recent past has seen a shift from matrix factorization and towards deep learning approaches with CF for recommendation systems. Of the numerous architectures available in the literature, autoencoders exactly fit the CF problem. This section discusses and critically reviews various models proposed in the literature that use AE as underlying architecture, and then presents factors of evaluation for these models. This section also discusses prominent datasets used in research in recommender systems.

One of the integrations of AEs in CF was introduced as AutoRec [20], a state-of-the-art model for recommendation. It has two variations: *I-AutoRec* (item based), and *U-AutoRec* (user based). It takes as input partial vectors of either users or items and aims to project it to a lower dimensional latent space in the hidden layer. It projects the reconstructions to the output layer, which are the recommendation results. It uses identity and sigmoid activation functions. By comparing RMSE values, authors conclude that I-AutoRec outperforms U-AutoRec on MovieLens 1M and 10M dataset.

Denoising Autoencoders (DAE) [17] introduce the concept of adding noise to the input. Strub and Mary [21] introduced an AE model that computes non-linear matrix factorization from sparse rating inputs. It has two versions: *Uencoder* and *Vencoder*. It converts the missing values to zero in the input and backpropagated layers. This is known as masking noise, which helps make recommendations. Input is sparse representation of users and items, and gives a dense output, hence addressing sparsity issues. The authors report V-AutoRec outperforms *Uencoder* and *Vencoder.*

Marginalized Stacked DAE (mSDA) [22] uses linear denoisers to marginalize out random feature corruption. It uses Stochastic Gradient Descent (SGD). This architecture addresses the issues of scalability, high-dimensional features, and high computation costs. *Deep CF* (DCF) model intermixes MF methods with marginalized DAEs, where the time complexity is $O(tN)$ (t is the number of iterations, N is the number of ratings). *mSDA-CF*, a variant of mSDA, stacks marginalized DAEs, and updates latent features on the layers given by the average of total number of layers and 1. The authors claim that this significantly improves time complexity.

Collaborative Denoising AE (*CDAE*) [23] is a generalized model of several other state-of-the-art. It generalizes Latent Factor Model [24, 25] and Factorized Similarity Model. CDAE has $I + 1$ node in the input layers, where I is the number of items, and the last node is a user-specific node. The hidden layer has a bias node, and the output has I nodes. Layers are fully-connected. CDAE uses SGD to learn parameters.

Based on Stacked DAE (SDAE), a CF model was developed [26]. It is user-based in nature and uses Gaussian noise to corrupt the data. It uses output of hidden layers for training process. Built upon it is CF-based Stacked Denoising AE model *CF-SDA*. It calculates the difference between the latent similarity obtained from the aforementioned model and the surface similarity. It employs Sigmoid and Identity mapping activation functions, Mean Squared Error as loss function and Adam optimizer.

Recommendation via *Dual AEs* (*ReDa*) [27] is a representational learning framework that learns latent features from user-item data and aims to minimize deviations in the training data. It uses stacked AEs to reach an optimal global solution. *ReDa* uses Sigmoid function during encoding and decoding. The algorithm requires calculation of partial derivatives of the variables used in the equations of optimizations. The model uses gradient descent for optimization, until convergence, which is not guaranteed every time as the optimization problem proposed is not convex in nature.

Collaborative Adversarial AE (*CAAE*) [28] is a framework that is based on Generative Adversarial Networks (GAN) [18]. A GAN has a discriminator NN and a generator NN. The discriminator generates the probability of a point belonging to a data distribution, and the generator generates data to feed in the discriminator, to fool it. CAAE consists of a discriminator that uses L2 regularization and is implemented with Bayesian Personalized Ranking (BPR) for learning relative preferences on items. It has a positive and negative item generator AE. SGD is used for the learning process.

Fine-Grained Collaborative AE (*FG-ACAE*) [29] is an adversarial framework for non-linear model optimization, where the noises iteratively minimize and maximize the loss function. This framework is applied to CDAE [23], where the noise mixing layer replaces denoising layer. Identity activation function is used in the mixing layer, loss function is cross-entropy, and mini-batch gradient descent for optimization.

Mult-VAE$^{\text{PR}}$ [30] extends Variational AE (VAE) [31, 32] for CF for implicit feedback incorporating non-linear probabilistic model. The model samples K-dimensional latent representations from the Gaussian prior, and non-linear function f_θ are applied to it to produce probability distribution over the items. Softmax function is used for output normalization. The model uses variational inference to minimize the Kullback-Leiber (KL) divergence. Using multinomial likelihood for the data distribution and adjusting the over-regularized VAE objective gives significant improvement over state-of-the-art baselines. It uses Bayesian inference for parameter estimation.

Sequential Variational AE (SVAE) [33] conditions each event to the previous event, thus modelling temporal dependencies. The authors argue that there is a recurrent relationship between the current datapoint and the previous one.

Queryable Variational AE (*Q-VAE*) [34] models the joint probability of the user's preferences and the conditional probability of other similar user's preferences. It models the log joint probability of the subset of partial preferences of the user, where the partition of this subset is arbitrary. Authors claim this is different from the existing VAE based models. It uses KL divergence for regularizing posterior distributions.

RecVAE [35] is built upon *Mult-VAE*$^{\mathrm{PR}}$. It changes the encoder of *Mult-VAE*$^{\mathrm{PR}}$ to a denoising encoder. The decoder uses Softmax activation. It makes a convex combination of standard Gaussian prior and a regularization term in the form of KL divergence. The training takes alternatively on the encoder and the decoder. Monte-Carlo sampling is used for log-likelihood and KL divergence. It uses cross-entropy as the main component of the loss function. The authors claim it outperforms other methods.

AutoSVD++ [36] is a hybrid CF model which uses Contractive Auto-Encoders (CAE) [19] and SVD++ [37]. Atop this is, a new hybrid model that takes implicit feedback as input. The model takes high-level feature representations from CAE, which is integrated with AutoSVD++ model. The model tries to minimize the regularized squared error loss. It uses SGD to learn parameters. Authors claim that their model outperforms *mSDA-CF* and *U-AutoRec*. They claim that this model is scalable.

Semi Auto-Encoder (*SA-HCF*) [38] proposes an architecture where the output layer is shorter than the input layer which makes incorporating side information easier. Based on it, the authors have designed a hybrid CF model. The authors claim it can solve the problem of rating prediction and ranking prediction. For ranking prediction, the user's partial observed vector and profile vector is used, and for rating prediction item's partially observed vectors and explicit ratings are used.

Hybrid collaborative filtering (HCF) model with semi-stacked denoising AE (Semi-SDAE), referred to as *HCF-SS* [39] is a hybrid model. The Semi-SDAE takes three inputs: the original, the noise-corrupted, and the side information data. Semi-SDAE can incorporate extended side information as input, without changing the dimension of the output layer. The model uses two layers of Semi-SDAE which is incorporated into MF. The parameters are learnt using SGD. The authors demonstrate that *HCF-SS* model outperforms *AutoSVD*++, *ReDa* and *SA-HCF.*

AutoCOT [40] is based on cooperative training (COT). It implements AutoRec and combines COT model with it. The mediator model generates a mixture distribution of real and generated user browsed data, and aims to optimize the KL divergence, and the generator model optimizes the JS divergence between them. The mediator and generator are one hidden layered architecture. The training process occurs iteratively on them. Authors claim AutoCOT can alleviate sparsity issues in large datasets.

Deep Heterogeneous AEs (*DHA*) [41] utilize information from multiple domains for accuracy improvement. It uses SDAE and RNN to extract latent features from non-sequential and sequential data respectively. It uses two Long Short Term Memory networks for the sequential data. The model incorporates two DHAs. It uses coordinate descent for optimizations and SGD for learning weight matrix and bias vectors. The author's experimentations demonstrate competitive model performance.

Built upon U-AutoRec, is an AE based CF (*ACF*) framework that provides top-N recommendation. It employs two AEs: *PR-Net and IL-Net* [42]. *PR-Net* employs pairwise regularization and samples negative samples, while *IL-Net* captures the items which are more likely to be true negatives. *PR-Net* is fed explicit feedback and *IL-Net* is fed the implicit feedback. SGD is used to learn models. The authors claim that this framework can also be extended for one-class CF by altering the inputs.

EASE$^{\mathrm{R}}$ [43] is a linear model with zero hidden layers. It takes a user-item interaction matrix X (generally binary) as the input. The diagonal of the item-item weight matrix must be constrained to zero. This model uses square loss between X and predicted scores. EASE$^{\mathrm{R}}$ states that the constrained convex optimization problem has a closed solution. The authors state the computational complexity of the algorithm as $O(|I|^3)$ where I is the number of items. Authors claim the model's superior accuracy.

AE algorithm for CF (AE-CF) [44] presents a combination of AE with clustering. The scoring data is combined with the user data to construct a scoring matrix, for extracting user features. This is used to plot the clusters, and using k-means recommendations are made to the user. This algorithm narrows the search space. The AE is used here for dimensionality reduction. It uses Pearson's correlation coefficient.

3.1 Matrices Used for Evaluation of Approaches

Models surveyed in this paper use Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Precision and Recall, Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG), Mean Reciprocal Rank (MRR) as evaluation metrics. These are very well-known metrics and have been thoroughly studied in the literature.

3.2 Datasets

Table 2 mentions the highlights of the major datasets used in research related to recommendation systems and existing in the literature.

Table 2 Datasets used in the models

Name	Domain	#users	#item	#ratings
Movielens (ML)100K	Movies	943	1682	100,000
Movielens (ML)1M	Movies	6039	3883	1,000,209
Movielens (ML)10M	Movies	71,567	10,681	10,000,054
Movielens (ML)20M	Movies	136,677	20,108	20,000,000
Netflix	Movies	463,435	17,769	56,900,000
Book-crossing (BC)	Books	278,858	271,379	1,149,780
Advertising	Services	448,158	737	N.A
Yelp	Business	9600	7000	243,000
MillionSong data (MSD)	Songs	571,355	41,140	33,600,000
Watcha	Movies	1391	1927	101,037

4 Comparison of Various Approaches Surveyed in the Study

This section provides a comparison of various approaches reported in the last decade and existing in the literature for making the recommendation based on CF and AE as architecture. The approaches are reviewed based on the model of AEs used in the approach, the category of information taken for applying CF, the datasets used. Table 3 also summarizes the methodology adapted, results reported, and their comparison with other contemporary approaches in the light of evaluation metrics (Table 4).

5 Contribution of the Survey

In this paper, multiple models based on AE architecture for collaborative filtering are surveyed. It forms an informative basis for decision making when trying to select a model for implementation. This survey also highlights the limitations in these models, which can inspire innovation of new models that will overcome those limitations. Similarly, the plus points of the models surveyed can inspire new models to build upon one or more of them, therefore combining the positives of multiple models.

6 Conclusion and Future Work

In this paper, a thorough survey has been conducted on AE based models that employ CF methods for recommender systems. Through the survey, it is well evident that the research in the field of recommender systems is well flourishing and has a great scope of applications. In the literature, it is proposed that, as the number of layers in the AEs

Table 3 Comparison of various approaches surveyed in the study

Refs.	Model	Basis	Dataset	Description	Limitation	Metrics	Results
[20]	AutoRec	User and Item	Ml1M Ml10M, Netflix	Vanilla AE	Does not guarantee performance [23]	RMSE	No. of hidden units determine error
[45]	mSDA-CF	Item–Item	Ml100K ML1M, BC, Advertising	Uses MF method with mSDA	Includes only ratings, not review text. High computational cost	RMSE	Reduced RMSE, better stability
[23]	CDAE	User –User	Movielens, Netflix, Yelp	Generalizes other state-of-the-art	Poor performance with sparse inputs [34]	Precision and Recall, MAP	Outperforms other compared models
[26]	CF-SDA	User –User	ML1M	SDAE based, computes latent and surface similarity	High RMSE values which indicate dissimilarity between recommendations	RMSE, Jaccard similarity	Standardization improves accuracy
[27]	ReDa	Item–Item	ML100K, ML1M, DB, DM	Based on representational learning	Organizes user's rating behaviour in a sparse matrix [46], convergence not guaranteed always	MAE, RMSE	Denser data gives better results
[28]	CAAE	Item–Item	Ciao, Watcha, Ml100K	Employs GAN, implements BPR	The training time observed is comparatively very high	Precision and Recall, nDCG, MRR	Accuracy improves with BPR
[29]	FG-ACAE	Item–Item	Ciao, ML1M	Designs minimax game for CDAE	Less efficient on sparse dataset	Hit ratio and NDCG	Almost converges on 500 epochs

(continued)

Table 3 (continued)

Refs.	Model	Basis	Dataset	Description	Limitation	Metrics	Results
[30]	Mult-VAE^{PR}	Item–Item	ML20M, Netflix, MSD	Implicit feedback used in non-linear probabilistic model	Input and output vectors are sparse, thus may lead to instability during training [35]	Recall and truncated nDCG	Robust, less sensitive to hyperparameter choice
[34]	Q-VAE	Both user, item based	ML1M, Netflix prize (2M users)	Combines joint and conditional likelihood	Does not perform well on Recall	Precision, Recall, MAP, NDCG	Fast and smooth convergence
[35]	RecVAE	User –User	ML20M, Netflix, MSD	Extends Mult-VAE^{PR}, adds DAE	Might not be pragmatically better-performing	Recall and NDCG	Improves Mult-VAE^{PR}
[36]	AutoSVD + +	User –User	ML100K, ML1M	Hybrid of CAE and SVD++	No incorporation of any content information [38]	RMSE	Time complexity reduced, scalable
[38]	SA-HCF	Item–Item	ML100K. ML1M	Asymmetrical architecture	Only considers explicit feedback	RMSE	Improved rating prediction
[39]	HCF-SS	Item–Item	ML1M, MovieLens-HetRec	Uses semi-SDAE, on asymmetrical architecture	Requires input from three sources, might be computationally expensive	MAE and RMSE	Low error as training set increases
[40]	AutoCOT	User–item	ML100K, ML1M	Uses cooperative learning	Requires more iteration for a smaller dataset	Precision and Recall	Alleviates sparsity

(continued)

Table 3 (continued)

Refs.	Model	Basis	Dataset	Description	Limitation	Metrics	Results
[41]	DHA model	User–item	ML100K, ML10M	Uses SDAE and RNN	Rich text as input, some versions may require higher time steps	MAP, Recall	Better Recall and Precision
[42]	ACF	Item–Item	Ciao, Watcha, Ml100K, ML1M	Extends U-AutoRec, employs PR-Net, IL-Net	Larger amount of training time than almost all compared methods	Precision, Recall, NDCG, reciprocal rank	Outperforms compared methods
[43]	EASE^R	Item–Item	ML20M, Netflix, MSD	No hidden layer has L2 normalization	Author's implementation may require too much memory	Recall, NDCG	Shorter training time
[44]	AE-CF	User–User	ML1M	Combines AE with clustering	Dependent on specific iterations	MAE	Alleviates cold start and sparsity

Table 4 Acronyms used in the table and their full forms

MF	Matrix factorization
LFM	Latent factor model
FSM	Factorized similarity model
GAN	Generative adversarial networks
BPR	Bayesian personalized ranking
CAE	Contractive autoencoders
SVD	Singular value decomposition

is increased from two to four, they perform slightly better. Also, the resultant setup behaves more robustly when the number and type of hyperparameters are changed. Through the survey, it has also been found that HCF-SS, EASE^R. HCF-SS and AE-CF perform great, in terms of evaluation metrics such as RMSE, MAE, Precision and Recall and NDGC when used with MovieLens as a dataset. A future work might be that a combination of two or more models can be implemented, for example, EASE^R with AE-CF, to see their synergetic effect on recommendations made.

References

1. Núñez-Valdéz, E.R., et al.: Implicit feedback techniques on recommender systems applied to electronic books. Comput. Hum. Behav. **28**(4), 1186–1193 (2012)
2. Isinkaye, F.O., Folajimi, Y.O., Ojokoh, B.A.: Recommendation systems: principles, methods and evaluation. Egypt. Inf. J. **16**(3), 261–273 (2015)
3. Ekstrand, M.D., Riedl, J.T., Konstan, J.A.: Collaborative filtering recommender systems. Found. Trends Hum. Comput. Interact. **4**(2), 81–173 (2011)
4. Terveen, L., Hill, W.: Beyond recommender systems: helping people help each other. In: HCI in the New Millennium, vol. 1, pp. 487–509 (2001)
5. Burke, R.: Hybrid recommender systems: survey and experiments. User model. User Adap. Inter. **12**(4), 331–370 (2002)
6. Khusro, S., Ali, Z., Ullah, I.: Recommender systems: issues, challenges, and research opportunities. In: Information Science and Applications (ICISA) 2016. Springer, Singapore, pp. 1179–1189 (2016)
7. Xin, Y.: Challenges in recommender systems: scalability, privacy, and structured recommendations. Diss. Massachusetts Institute of Technology (2015)
8. 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. **17**(6), 734–749 (2005)
9. Bobadilla, J., et al:. Recommender systems survey. Knowl. Based Syst. **46**, 109–132 (2013)
10. Do, M.-P.T., Van Nguyen, D., Nguyen, L.: Model-based approach for collaborative filtering. In: Proceedings of the 6th International Conference on Information Technology for Education (2010)
11. Yang, Z., et al.: A survey of collaborative filtering-based recommender systems for mobile internet applications. IEEE Access **4**, 3273–3287 (2016)
12. Zhang, S., et al.: Deep learning based recommender system: a survey and new perspectives. ACM Comput. Surv. (CSUR) **52**(1), 1–38 (2019)

13. Yan, A., et al.: CosRec: 2D convolutional neural networks for sequential recommendation. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management (2019)
14. Devooght, R., Bersini, H.: Collaborative filtering with recurrent neural networks. arXiv preprint arXiv:1608.07400 (2016)
15. Salakhutdinov, R., Mnih, A., Hinton, G.: Restricted Boltzmann machines for collaborative filtering. In: Proceedings of the 24th International Conference on Machine Learning (2007)
16. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press. https://www.deeplearningbook.org (2016)
17. Vincent, P., et al.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning (2008)
18. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (2014)
19. Rifai, S., et al.: Contractive auto-encoders: explicit invariance during feature extraction (2011)
20. Sedhain, S., et al.: Autorec: autoencoders meet collaborative filtering. In: Proceedings of the 24th International Conference on World Wide Web (2015)
21. Strub, F., Mary, J.: Collaborative filtering with stacked denoising autoencoders and sparse inputs (2015)
22. Chen, M., et al.: Marginalized denoising autoencoders for domain adaptation. arXiv preprint arXiv:1206.4683 (2012)
23. Wu, Y., et al.: Collaborative denoising auto-encoders for top-n recommender systems. In: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining (2016)
24. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: Advances in Neural Information Processing Systems, pp. 1257–1264 (2007)
25. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
26. Suzuki, Y., Ozaki, T.: Stacked denoising autoencoder-based deep collaborative filtering using the change of similarity. In: 2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA). IEEE (2017)
27. Zhuang, F., et al.: Representation learning via dual-autoencoder for recommendation. Neural Netw. **90**, 83–89 (2017)
28. Chae, D.-K., Shin, J.A., Kim, S.-W.: Collaborative adversarial autoencoders: an effective collaborative filtering model under the GAN framework. IEEE Access **7**, 37650–37663 (2019)
29. Yuan, F., Yao, L., Benatallah., B.: Adversarial collaborative neural network for robust recommendation. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (2019)
30. Liang, D., et al.: Variational autoencoders for collaborative filtering. In: Proceedings of the 2018 World Wide Web Conference (2018)
31. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
32. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. arXiv preprint arXiv:1401.4082 (2014)
33. Sachdeva, N., et al.: Sequential variational autoencoders for collaborative filtering. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (2019)
34. Wu, G., Bouadjenek, M.R., Sanner, S.: One-class collaborative filtering with the queryable variational autoencoder. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (2019)
35. Shenbin, I., et al.: RecVAE: a new variational autoencoder for top-N recommendations with implicit feedback. In: Proceedings of the 13th International Conference on Web Search and Data Mining (2020)
36. Zhang, S., Yao, L., Xu, X.: AutoSVD++: an efficient hybrid collaborative filtering model via contractive auto-encoders. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (2017)

37. 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 (2008)
38. Zhang, S., et al.: Hybrid collaborative recommendation via semi-autoencoder. In: International Conference on Neural Information Processing. Springer, Cham (2017)
39. Zou, H., et al.: Hybrid collaborative filtering with semi-stacked denoising autoencoders for recommendation. In: 2019 IEEE International Conference on Dependable, Autonomic and Secure Computing, International Conference on Pervasive Intelligence and Computing, International Conference on Cloud and Big Data Computing, International Conference on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech). IEEE (2019)
40. Bai, R., et al.: AutoCOT-autoencoder based cooperative training for sparse recommendation. In: 2018 IEEE 15th International Conference on e-Business Engineering (ICEBE). IEEE (2018)
41. Li, T., et al.: Deep heterogeneous autoencoders for collaborative filtering. In: 2018 IEEE International Conference on Data Mining (ICDM). IEEE (2018)
42. Chae, D.K., Kim, S.W., Lee, J.T.: Autoencoder-based personalized ranking framework unifying explicit and implicit feedback for accurate top-N recommendation. Knowl.-Based Syst. **176**, 110–121 (2019)
43. Steck, H.: Embarrassingly shallow autoencoders for sparse data. In: The World Wide Web Conference (2019)
44. Chu, H., et al.: Towards a deep learning autoencoder algorithm for collaborative filtering recommendation. In: 2019 34rd Youth Academic Annual Conference of Chinese Association of Automation (YAC). IEEE (2019)
45. Li, S., Kawale, J., Fu, Y.: Deep collaborative filtering via marginalized denoising auto-encoder. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (2015)
46. Yi, B., et al.: Deep matrix factorization with implicit feedback embedding for recommendation system. IEEE Trans. Ind. Inf. **15**(8), 4591–4601 (2019)