User Preference Learning in Multi-criteria Recommendations using Stacked Auto Encoders

Dharahas Tallapally, Rama Syamala Sreepada, Bidyut Kr. Patra, Korra Sathya Babu
Knowledge and Data Engineering Laboratory
Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela, Odisha, INDIA
kanna.dharahas@gmail.com,{515cs1002,patrabk,ksathyababu}@nitrkl.ac.in

ABSTRACT

Recommender System (RS) is an essential component of many businesses, especially in e-commerce domain. RS exploits the preference history (rating, purchase, review, etc.) of users in order to provide the recommendations. A user in traditional RS can provide only one rating value about an item. Deep Neural Networks have been used in this single rating system to improve recommendation accuracy in the recent times. However, the single rating systems are inadequate to understand the usersfi preferences about an item. On the other hand, business enterprises such as tourism, e-learning, *etc.* facilitate users to provide multiple criteria ratings about an item, thus it becomes easier to understand users' preference over single rating system. In this paper, we propose an extended Stacked Autoencoders (a Deep Neural Network technique) to utilize the multi-criteria ratings. The proposed network is designed to learn the relationship between each user's criteria and overall rating efficiently. Experimental results on real world datasets (Yahoo! Movies and TripAdvisor) demonstrate that the proposed approach outperforms state-of-the-art single rating systems and multi-criteria approaches on various performance metrics.

CCS CONCEPTS

•**Information Systems** → *Information Retrieval;* •**Retrieval Tasks and Goals** → **Recommender Systems;**

KEYWORDS

Collaborative filtering, Multi-criteria ratings, Deep Learning, Stacked Autoencoders

ACM Reference format:
Dharahas Tallapally, Rama Syamala Sreepada, Bidyut Kr. Patra, Korra Sathya Babu. 2018. User Preference Learning in Multi-criteria Recommendations using Stacked Auto Encoders. In *Proceedings of Twelfth ACM Conference on Recommender Systems, Vancouver, BC, Canada, October 2–7, 2018 (RecSys '18),* 5 pages.
DOI: 10.1145/3240323.3240412

1 INTRODUCTION

In the last few decades, Recommender Systems (RS) has been playing a pivotal role in overcoming the problem of information overload especially in e-commerce domain. The primary task of an RS is to provide personalized suggestions (items/ services) to the users by processing the available users' information (ratings, reviews, purchases, *etc.*). This task of recommendations is performed in two ways: *content based filtering* and *collaborative filtering (CF).* Content based filtering techniques analyze active user's content, profiles of the current item and items preferred in the past [11], [17]. CF techniques on the other hand are most successful and widely used in e-commerce industry [12]. These algorithms recommend unrated items to the users by processing the users' rating history. CF algorithms are classified into 1/itMemory-based CF and *Model based CF.* Memory based CF finds the neighbors (similar users/ items) of the active user (target item) and utilizes the neighbors ratings while providing recommendations [19]. Model based CF techniques, which are proven to be more accurate, learn users' and items' features while building a model using machine learning techniques. Several efficient techniques such as matrix factorization, factorization machines, deep neural networks are adopted in learning the preferences of the users [9], [18], [3], [23]. The above mentioned approaches are traditionally used on single rating systems.

However, single rating recommender systems are not sufficient to capture users' feedback in multiple dimensions or criteria, especially in service domains such as restaurants, hotels, movies, *etc.*. In multi-criteria recommender systems, a user can share more information in-terms of criteria ratings for an item. For instance, a user can provide six ratings (including overall rating) in a restaurant domain with the criterion being taste, hygiene, ambiance, hospitality and price. Therefore, multi-criteria recommender system receives richer information and they have a better ability in recommending relevant items to the users compared to single rating systems [1]. A few researchers attempted to utilize multi-criteria rating information while providing the recommendations [1], [8], [27]. However, various machine learning techniques are unexplored in multi-criteria recommender systems.

In this paper, we adapt a deep learning technique called Stacked Autoencoders. We propose an extended version of Stacked Autoencoders to cater the requirement of multi-criteria rating systems. The input layer of the traditional network and the loss function are effectively modified to learn the relationship between multi-criteria ratings and their respective overall ratings. The proposed approach significantly outperforms traditional single rating recommender systems and state-of-the-art multi-criteria recommender systems.

Rest of the paper is organized as follows. In Section 2, we discuss existing literature in multi-criteria recommender systems. Background of this paper is discussed in Section 3. Proposed approach and experimental details are discussed in Sections 4 and 5, respectively. Finally, we conclude our work in Section 6.

2 RELATED WORK

One of the earliest works on multi-criteria recommendation system includes G Adomavicius and Y Kwon's work [1]. In this approach, statistical techniques are used to derive an aggregation function between the multi-criteria ratings and the overall rating. Multi-criteria ratings are predicted using traditional techniques and aggregation function is used to obtain the overall rating while providing recommendations. However, the preference of criteria remains same across all the users in this approach.

In [26], Zhang *et al.* proposed two types of Probabilistic Latent Semantic Analysis (PLSA) models. However, the PLSA models tend to converge at local minima.

Liu *et al.* in [13] applied T-test on users' ratings to obtain significant criteria and form a "preference lattice" utilizing all the significant criteria for all the users. However, this approach assumes that the rating dataset follows a normal distribution and T-test cannot be applied on sparse rating set. Shambour and Lu [21] proposed a hybrid model which combines content based filtering and item-based CF to alleviate the problem of cold-start and sparsity in multi-criteria recommender system.

Jannach *et al.* proposed accuracy improvement techniques in [8]. This work closely follows G Adomavicius and Y Kwon's work [1]. Support Vector Regression (SVR) is used to learn the relationship between overall rating and criteria ratings. Further, gradient descent technique is applied to find the weightage for item-based rating prediction and user-based rating prediction for each user. However, the weightage of each criteria remains unchanged across the users, which might lead to non-personalized suggestions.

Nilashi contributed a series of works in multi-criteria recommender systems [14], [16] and [15]. In [14], Nilashi *et al.* proposed fuzzy based algorithms for improving the accuracy in multi-criteria recommender systems. In [16], Nilashi proposed a hybrid approach to utilize Ant K-means clustering algorithm and significant criteria from each user cluster are extracted using Principal Component Analysis. SVR is used to learn the relationship between overall rating and extracted criteria. However, if the resultant clusters have lesser rating information, regression techniques fail to learn the weightage of criterion.

In [22], Sreepada *et al.* proposed a preference learning approach, where the preferred criteria of each user and popular criteria of each item are extracted. The extracted criteria preferences are used to find the neighbors of active user (and target item). Finally, user-based and item-based CF techniques are adapted while providing the recommendations. However, the popular criteria of an item is not personalized *w.r.t.* the active user in this approach.

Recently in [27], Zheng proposed three types of 'criteria chain' approach. In the first approach, first criteria rating is predicted and it is used as context while predicting the second criteria rating and so on. Finally, SVR is used to predict the overall rating using the criteria ratings. In the second approach, Context-Aware Matrix Factorization (CAMF) is used to predict the final rating by utilizing the criteria ratings as the contextual information. In the third criteria chain approach, each criteria rating is predicted independently and the predicted ratings are used as context while predicting the overall rating using CAMF. The criteria ratings are predicted and used in the chain, which might lead to an accumulated loss while predicting the overall rating.

In the following section, we discuss stacked autoencoders which is adapted in our paper.

3 BACKGROUND

Deep learning techniques have shown tremendous achievements in speech recognition, image recognition, natural language processing [5], [4]. Deep learning in recommender systems has recently started gaining momentum [23], [3]. Autoencoders which is a deep learning technique has been applied on single rating recommender systems in [23].

Autoencoder is an unsupervised deep neural network, which was initially proposed by Rumelhart in 1986 [20], [7]. This network works by compressing the input into a latent-space representation, and then reconstructing the output from it. Autoencoder Neural Network has two parts encoder and decoder shown as follow.

- **Encoder**: This function encodes the input layer as shown in Equation 1.

$$f(x) = \sigma(W_1 x + b_1) \tag{1}$$

- **Decoder**: This function decodes the hidden layer as shown in Equation 2.

$$g(f(x)) = \sigma(W_2 \times f(x) + b_2) \tag{2}$$

where, $\sigma(.)$ is an activation function, x is the input data and (W_1, W_2) and $(b_1, b2)$ are weights and biases for each layer respectively. Here, $f(x)$ encodes the input data, $g(f(x))$ decodes $f(x)$ and the network is trained to equate $g(f(x))$ with x.

In [24], Vincent *et al.* introduced Stacked Autoencoders, in which multiple encoder layers are connected followed by decoder layers. This process enables the last layer of the encoder network to find the lowest dimensional representations. It experimentally increases the quality of the whole network. The parameters of each layer are trained separately, while the parameters of the rest of the network remain unchanged.

The first layer of a stacked autoencoder tends to learn first-order features in the raw input (such as edges in an image). The second layer of a stacked autoencoder tends to learn second-order features. Higher layers of the stacked autoencoder tend to learn even higher-order features. Recent works in deep learning advocates to stack pretrained encoders to initialize deep neural networks.

4 PROPOSED APPROACH

We propose to extend stacked autoencoders to cater multi-criteria ratings in the network. Subsequently, the modified loss function is discussed.

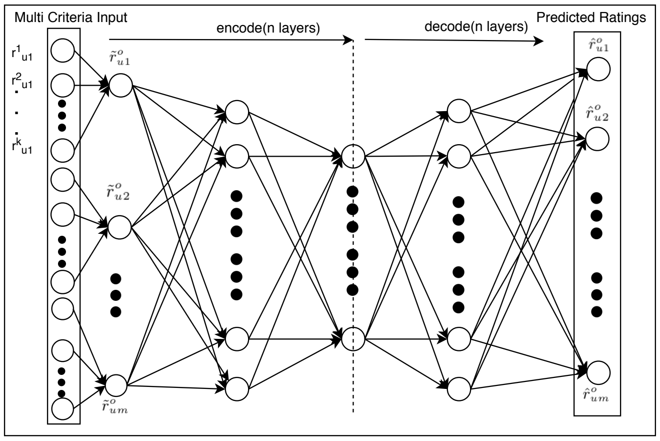

Figure 1: Proposed Extended Stacked Auto-Encoder

4.1 Extended Stacked Autoencoder

As discussed in the Section 3, stacked autoencoders are unsupervised networks where the output of the network aims to reconstruct the initial input. The reconstruction is performed by back-propagating the squared error loss.

In conventional stacked autoencoders, the number of neurons in the input layer is equal to the number of input's features. Each feature's information is fed at respective neurons in the input layer. However, in multi-criteria scenario, each item has multiple criteria ratings which are required as input to the network. Therefore, we extend the traditional stacked autoencoder by incorporating an extra layer which accomodates the multi-crteria ratings. This added layer acts as an input layer in our extended stacked autoencoder. The extended stacked autoencoder for multi-criteria rating scenario is shown in Figure 1. The first layer acts as multi-criteria input which is connected to the respective item neurons in the succeeding layer, termed as intermediate layer. This intermediate layer is connected to N successive encoding layers, which are used to encode the latent representation of the items. The last encoding layer is connected to N consecutive decoding layers. The first decoder layer decodes the latent factors learnt from N^{th} encoder layer. Likewise, other decoder layers are used to decode the previously obtained latent factors from respective encoders. The final layer acts as an output where the overall ratings of the items are predicted. Let r^o_{ui} be the overall rating and $r^1_{ui}, r^2_{ui}, ..., r^k_{ui}$ be the criteria ratings made by user u on item i, where k is the number of criteria. In the intermediate layer, we compute an intermediate rating ($\tilde{r}^o_{ui}$) at item neuron i using multi-criteria ratings of the item. For example, the intermediate rating ($\tilde{r}^o_{ui}$) at neuron i is computed as shown in Equation 3.

$$\tilde{r}^0_{ui} = w^1_i * r^1_{ui} + w^2_i * r^2_{ui} + w^3_i * r^3_{ui} + ... + w^k_i * r^k_{ui} \tag{3}$$

where, $w^1_i, w^2_i, w^3_i, .., w^k_i$ are the criteria weights of item i. Likewise, the intermediate rating of each rated item is computed using the respective criteria ratings. These intermediate ratings are fed to the subsequent layers of the network and finally the overall ratings are predicted while training the proposed stacked autoencoder. Therefore, a fully connected network is designed using the extended input layer. The weights of multi-criteria of each item are optimized along with the weights of stacked autoencoder during training. It can be noted that the neurons (in the intermediate layer) corresponding to the unrated items are deactivated while training the network. Due to this high sparsity issue, we modify the loss function as discussed in the following subsection.

4.2 Loss Function

We provide multi-crtieria ratings as input and we train the network to predict the overall ratings as the output. For all the deactivated item neurons, the respective predicted ratings are not considered while computing the loss. Therefore, we modify loss function to compute squared error only on the known item ratings. For the deactivated neurons, the network is not trained and hence error is not back-propagated. In other words, unrated instances do not bring information to the network. The modified loss function $\mathcal{L}(X, y)$ of the stacked autoencoder is shown in Equation 4.

$$\mathcal{L}(X, y) = \sum_{j \in \check{X}} \left(ext\_sae(X)_j - y_j\right)^2 \tag{4}$$

where, X is the whole set of rating instances such that $X = \bar{X} \cup \check{X}$, $\check{X}$ consists of all the known rated instances and $\bar{X}$ contains the unknown rated instances. The network $ext\_sae(.)$ is the fully connected extended stacked autoencoder as shown in Figure 1. The predicted overall rating of instance j is $ext\_sae(\check{X})_j$ and y_j is the actual overall rating of instance j. From the Equation 4, it can be observed that only the known rating instance set ($\check{X}$) is used while computing the loss.

We include the regularization to avoid overfitting of the network. As a complete fully connected network, with multi-criteria ratings as input and using L2-reqularization from overfitting the network, the new training loss ($mod\_\mathcal{L}(X, y)$) is computed as shown in Equation 5.

$$mod\_\mathcal{L}(X, y) = \sum_{j \in \check{X}} \left(ext\_sae(X)_j - y_j\right)^2 + \lambda\left(||W||^2 + ||w||^2\right) \tag{5}$$

where, $||W||$ is the flatten vector of weights of the stacked autoencoder and $||w||$ are the flatten vector of weights of the multi-criteria ratings of each item. Here, regularization hyper-parameter is λ. Additionally dropout regularization is added at each layer with a probability p, in order to overcome the identity network.

5 PERFORMANCE EVALUATION

In this section, we discuss the datasets used, evaluation metrics, experimental setup and results of the proposed approach.

5.1 Dataset and Evaluation Metrics

Two real world datasets from tourism (TripAdvisor (TA) [25]) and movie (Yahoo! Movies (YM) [8]) domains are used to evaluate the proposed approach. To obtain the working data subset from TA, we retain the instances of users who rated atleast 5 hotels and hotels which were rated by atleast 5 users. The obtained subset (TA 5-5) contains 19,374 rating instances by 3,160 users and 3,550 hotels

Table 1: Results for TA 5-5

Technique	MAE	GIMAE	GPIMAE	F1
MF [10]	1.2077	1.3055	0.8079	0.4491
2016_Hybrid AE [23]	0.6531	0.6022	0.8406	0.6789
2011_Liwei Liu [13]	0.7720	0.5262	0.6282	0.6102
2017_Learning [22]	0.6204	0.5907	0.6103	0.6907
2017_CCC [27]	0.6737	0.5878	0.5901	0.4497
2017_CCA [27]	0.6914	0.6124	0.6095	0.4826
2017_CIC [27]	0.7129	0.6536	0.6814	0.4636
Extended_SAE_3	0.5674	0.5210	**0.5379**	**0.7458**
Extended_SAE_5	**0.5593**	**0.5075**	0.5490	0.7384

Table 2: Results for YM 5-5

Technique	MAE	GIMAE	GPIMAE	F1
MF [10]	1.2961	1.2755	0.6204	0.4882
2016_Hybrid AE [23]	0.7691	0.6314	0.8244	0.6798
2011_Liwei Liu [13]	0.7233	0.5750	0.7232	0.6706
2017_Learning [22]	0.6514	0.5019	0.5824	0.7107
2017_CCC [27]	0.6888	0.6242	0.7577	0.5380
2017_CCA [27]	0.6891	0.5417	0.5972	0.5640
2017_CIC [27]	0.7012	0.6420	0.7439	0.5370
Extended_SAE_3	0.6080	0.4636	0.5673	**0.7109**
Extended_SAE_5	**0.5854**	**0.4633**	**0.5592**	0.6073

Table 3: Results for YM 10-10

Technique	MAE	GIMAE	GPIMAE	F1
MF [10]	0.8478	0.7461	0.6765	0.5998
2016_Hybrid AE [23]	0.7811	0.6595	0.8269	0.7042
2011_Liwei Liu [13]	0.6574	0.5204	0.6574	0.6640
2017_Learning [22]	0.6576	0.5054	0.6576	0.6629
2017_CCC [27]	0.6374	0.6240	0.7857	0.5361
2017_CCA [27]	0.6618	0.6015	0.7990	0.5343
2017_CIC [27]	0.6719	0.6542	0.7743	0.5327
Extended_SAE_3	0.5783	0.4870	**0.6501**	0.7113
Extended_SAE_5	**0.5640**	**0.4842**	0.6503	**0.7939**

Table 4: Results for YM 20-20

Technique	MAE	GIMAE	GPIMAE	F1
MF [10]	0.7397	0.6077	0.5700	0.6698
2016_Hybrid AE [23]	0.7205	0.6008	0.7830	0.7578
2011_Liwei Liu [13]	0.6576	0.5054	0.6576	0.6828
2017_Learning [22]	0.8254	0.5958	0.8131	0.7544
2017_CCC [27]	0.6798	0.6095	0.7159	0.5585
2017_CCA [27]	0.6691	0.6042	0.6971	0.5641
2017_CIC [27]	0.7029	0.6218	0.7064	0.5677
Extended_SAE_3	0.5906	0.4959	0.6523	0.7973
Extended_SAE_5	**0.5798**	**0.4834**	**0.6306**	**0.8070**

with a sparsity of 99.8272%. Likewise, subsets of YM dataset such as YM 5-5, YM 10-10 and YM 20-20 are created [8]. To evaluate the effectiveness of the proposed approach, we use popular performance metrics such as Mean Absolute Error (MAE), F1 [6] and recently introduced Good Items MAE (GIMAE) and Good Predicted Items MAE (GPIMAE), [2].

5.2 Training Settings

We train the proposed stacked autoencoder with 3-hidden layers (Extended_SAE_3) and 5-hidden layers (Extended_SAE_5). We adopted two transfer functions *sigmoid* and *hyperbolic tangents*. For *sigmoid* transfer function, dataset is normalized to the range 0 and 1 and for *hyperbolic tangents*, dataset is normalized from -1 to 1. The network is optimized with *Gradient Descent Optimizer* and *Adam Optimizer* with minibatch of size 30 and weight decay is added for regularization. Dropout Regularization is added to each hidden layer with probability p=0.5. Experiments were conducted for different values of parameters and the best obtained parameter values are presented here. Out of all the rating instances, 90% are used for training and the remaining samples are used for testing the approach.

5.3 Results and Analysis

To compare the proposed approach with single rating and multi-criteria rating systems, we implemented our approach along with Matrix Factorization (MF) [10], 2016_Hybrid AE [23] and multi-criteria recommendation techniques: 2011_Liwei Liu [13], 2017_Learning [22], three approaches from [27] (2017_CCC, 2017_CCA, 2017_CIC). These methods are applied on all the working datasets and the results are reported in Table 1 - Table 3. Traditional Matrix Factorization (MF) technique incurred the highest loss in terms of MAE, GIMAE and GPIMAE with values 1.2077, 1.3055 and 0.8079, respectively on TA 5-5 dataset (Table 1). In terms of MAE and F1, 2017_Pref Learning performs better than existing single and multi-criteria rating techniques. However, the proposed approach (Extended_SAE) outperforms all the existing approaches across mentioned performance metrics (Table 1). Results on YM 5-5 dataset are reported in Table 2. It can be observed that MF incurred the highest loss and least F1. However, proposed extended stacked autoencoder outperforms all the approaches significantly in various evaluation metrics (Table 2). Similar trends are also found on the other datasets, YM 10-10 (Table 3) and YM 20-20 (Table 4). Due to lack of space, we omit analysis of the results on the above mentioned datasets.

6 CONCLUSION AND FUTURE WORK

In this paper, we proposed extended stacked autoencoders in multi-criteria rating scenario. The input layer of the traditional autoencoders is extended to accommodate the multi-criteria ratings. The performance of the proposed approach on real-world datasets is very promising to further focus in this direction. This work can be extended by incorporating side information (such as reviews) along with multi-criteria ratings to enhance the quality of recommendations.

7 ACKNOWLEDGEMENT

This work is partly funded by Media Lab Asia, Ministry of Electronics and Information Technology (MeitY), Govt. of India, Grant No. PhD-MLA/4(13)/2015-2016, dated: 24/07/2015. Partial travel grant for attending the conference is funded by electronic commerce company Flipkart Pvt Ltd.

REFERENCES

[1] Gediminas Adomavicius and YoungOk Kwon. 2007. New recommendation techniques for multicriteria rating systems. *IEEE Intelligent Systems* 22, 3 (2007).

[2] Fidel Cacheda, Víctor Carneiro, Diego Fernández, and Vreixo Formoso. 2011. Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Transactions on the Web (TWEB)* 5, 1 (2011), 2.

[3] Sanxing Cao, Nan Yang, and Zhengzheng Liu. 2017. Online news recommender based on stacked auto-encoder. In *Computer and Information Science (ICIS), 2017 IEEE/ACIS 16th International Conference on.* IEEE, 721–726.

[4] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. 2012. Multi-column deep neural networks for image classification. In *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on.* IEEE, 3642–3649.

[5] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on.* IEEE, 6645–6649.

[6] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 5–53.

[7] Geoffrey E Hinton and Richard S. Zemel. 1994. Autoencoders, Minimum Description Length and Helmholtz Free Energy. In *Advances in Neural Information Processing Systems 6*, J. D. Cowan, G. Tesauro, and J. Alspector (Eds.). Morgan-Kaufmann, 3–10.

[8] Dietmar Jannach, Zeynep Karakaya, and Fatih Gedikli. 2012. Accuracy improvements for multi-criteria recommender systems. In *Proceedings of the 13th ACM Conference on Electronic Commerce.* ACM, 674–689.

[9] Yehuda Koren. 2008. 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, 426–434.

[10] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009).

[11] Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995.* Elsevier, 331–339.

[12] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 7, 1 (2003), 76–80.

[13] Liwei Liu, Nikolay Mehandjiev, and Dong-Ling Xu. 2011. Multi-criteria service recommendation based on user criteria preferences. In *Proceedings of the fifth ACM conference on Recommender systems.* ACM, 77–84.

[14] Mehrbakhsh Nilashi, Othman bin Ibrahim, and Norafida Ithnin. 2014. Hybrid recommendation approaches for multi-criteria collaborative filtering. *Expert Systems with Applications* 41, 8 (2014), 3879–3900.

[15] Mehrbakhsh Nilashi, Othman bin Ibrahim, Norafida Ithnin, and Nor Haniza Sarmin. 2015. A multi-criteria collaborative filtering recommender system for the tourism domain using Expectation Maximization (EM) and PCA–ANFIS. *Electronic Commerce Research and Applications* 14, 6 (2015), 542–562.

[16] Mehrbakhsh Nilashi, Dietmar Jannach, Othman bin Ibrahim, and Norafida Ithnin. 2015. Clustering-and regression-based multi-criteria collaborative filtering with incremental updates. *Information Sciences* 293 (2015), 235–250.

[17] Michael Pazzani and Daniel Billsus. 1997. Learning and revising user profiles: The identification of interesting web sites. *Machine learning* 27, 3 (1997), 313–331.

[18] Steffen Rendle. 2010. Factorization machines. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on.* IEEE, 995–1000.

[19] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. Introduction to recommender systems handbook. In *Recommender systems handbook.* Springer, 1–35.

[20] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1988. Neurocomputing: Foundations of Research. MIT Press, Cambridge, MA, USA, Chapter Learning Representations by Back-propagating Errors, 696–699.

[21] Qusai Shambour and Jie Lu. 2011. A hybrid multi-criteria semantic-enhanced collaborative filtering approach for personalized recommendations. In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01.* IEEE Computer Society, 71–78.

[22] Rama Syamala Sreepada, Bidyut Kr Patra, and Antonio Hernando. 2017. Multi-criteria Recommendations through Preference Learning. In *Proceedings of the Fourth ACM IKDD Conferences on Data Sciences.* ACM, 1.

[23] Florian Strub, Romaric Gaudel, and Jérémie Mary. 2016. Hybrid recommender system based on autoencoders. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems.* ACM, 11–16.

[24] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* 11, Dec (2010), 3371–3408.

[25] Hongning Wang, Yue Lu, and ChengXiang Zhai. 2011. Latent aspect rating analysis without aspect keyword supervision. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 618–626.

[26] Yin Zhang, Yueting Zhuang, Jiangqin Wu, and Liang Zhang. 2009. Applying probabilistic latent semantic analysis to multi-criteria recommender system. *Ai Communications* 22, 2 (2009), 97–107.

[27] Yong Zheng. 2017. Criteria Chains: A Novel Multi-Criteria Recommendation Approach. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces.* ACM, 29–33.