International Journal of Computer Applications (0975 – 8887)
Volume 162 – No 10, March 2017

Deep Learning Methods on Recommender System: A Survey of State-of-the-art

Basiliyos Tilahun Betru
University of Yaoundé I
National advanced school of engineering and
Jimma University Institute of Technology

Charles Awono Onana
University of Yaoundé I
National Advanced School of Engineering

Bernabe Batchakui
University of Yaoundé I
National Advanced School of Engineering

ABSTRACT
The advancement in technology accelerated and opened availability of various alternatives to make a choice in every domain. In the era of big data it is a tedious and time consuming task to evaluate the features of a large amount of information provided to make a choice. One solution to ease this overload problem is building recommender system that can process a large amount of data and support users' decision making ability. In this paper different traditional recommendation techniques, deep learning approaches for recommender system and survey of deep learning techniques on recommender system are presented. A variety of techniques have been proposed to perform recommendation, including content based, collaborative and hybrid recommenders. Due to the limitation of the traditional recommendation methods in obtaining accurate result a deep learning approach is introduced both for collaborative and content based approaches that will enable the model to learn different features of users and items automatically to improve accuracy of recommendation. Even though deep learning poses a great impact in various areas, applying the model to a recommender systems have not been fully exploited. With the help of the advantage of deep learning in modeling different types of data, deep recommender systems can better understand users' demand to further improve quality of recommendation.

General Terms
Recommender system, deep learning

Keywords
Recommender system, deep learning, big data, decision making, collaborative filtering, hybrid recommender.

1. INTRODUCTION
Advancements of online technology accelerated and opened availability of various alternatives to make a choice based on user preferences. In the era of big data it is a tedious and time consuming task for users to evaluate features of a large amount of information provided to make a choice of their preferences. Recommender system have proven to be a valuable technique that can process a large amount of data and support users' decision making ability in finding optimal recommendations by reducing the number of alternatives to those the user will likely prefer. A recommender system is defined as [5] a personalized information filtering technology used to either predict whether a particular user will like a particular item or to identify a set of N items that will be of interest to a certain user. Nowadays, many people use recommender systems in their daily life to deal with information overload and provide personalized recommendations, content, and services to them[1, 28] examples of such applications include recommendation of product or service on E-commerce platforms and others. Usually, a recommender system recommends items by either predicting ratings or providing a ranked list of items for each user [1]. A variety of techniques have been proposed to perform recommendation, including content-based, collaborative and hybrid recommenders [6, 31]. This paper is organized as follows: Section 2 presents review of different traditional recommendation techniques. Deep learning approaches for recommender system are presented in section 3. In section 4 survey of deep learning techniques on recommender system are presented. Finally, conclusion remarks and outlook to future work are given in section 5.

2. TRADITIONAL RECOMMENDER SYSTEM

2.1 Collaborative filtering (CF)
Collaborative filtering technique utilizes past activities or a set of similar tastes of users without considering user or item content information [15, 42]. Collaborative filtering techniques can be either memory-based or model-based.

Memory-based collaborative filtering, as described by [24], it tries to find users that are similar to the active user and uses their preferences to predict ratings for the active user. There are several advantage of memory based collaborative filtering: it is scalable to large size of data, relatively simple to implement for any condition, easy to update the database thus new arrival data can be handled easily and it provides feedback to explain on how the recommender system works can be provided by explicitly listing content features or descriptions that caused an item to occur in the list of recommendations [31, 41]. However, several limitations are also existed in memory-based CF techniques: it is very slow as it uses the entire database every time it makes a prediction, makes unreliable and inaccurate prediction if the active user has no items in common with all users who have rated the item to be recommended [31, 41].

Similarity computation between items or users is a critical stage in memory-based collaborative filtering algorithms. Correlation-based similarity measures the similarity $w_{u,v}$ between two users u and v, or $w_{i,j}$ between two items i and j, is measured by computing the Pearson correlation or other correlation-based similarities. Pearson correlation measures the extent to which two variables linearly relate with each other [31 39, 50]. For the user based algorithm, the Pearson correlation between users' u and v is

$$W_{u,v} = \frac{\sum_{i\in I}(r_{u,i}-\overline{r_u})(r_{v,i}-\overline{r_v})}{\sqrt{\sum_{i\in I}(r_{u,i}-\overline{r_u})^2}\sqrt{\sum_{i\in I}(r_{v,i}-\overline{r_v})^2}}, \quad (1.1)$$

Where the i $\in I$ sums over the items that are rated by both user u and user v.

For the item-based algorithm, denote the set of users' $u \in U$ who rated both items i and j, then the Pearson Correlation will be

$$W_{i,j} = \frac{\sum_{u \in U}(r_{u,j} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U}(r_{u,j} - \bar{r}_i)^2}\sqrt{\sum_{u \in U}(r_{u,j} - \bar{r}_j)^2}}, \qquad (1.2)$$

Where $r_{u,i}$ is the rating of user u on item i, r_i is the average of the i^{th} item by those users.

Vector Cosine-based Similarity is adopted in neighbor-based collaborative filtering to compute the similarity across users or items. Vector Cosine-Based Similarity measures the similarity between two documents by treating each document as a vector of word frequencies and computing the cosine of the angle formed by the frequency vectors [14, 31, 39 and 50]. Vector cosine similarity between items i and j is given by,

$$W_{i,j} = \cos(\vec{\imath}, \vec{\jmath}) = \frac{\vec{\imath} \, . \, \vec{\jmath}}{\|\vec{\imath}\| * \|\vec{\jmath}\|} \quad , \qquad (1.3)$$

Where "• "denotes the dot product of two vectors.

Model-based collaborative filtering, Is machine learning or data mining based model which finds complex rating patterns in training data and then make intelligent predictions or recommendation for the user based on the learned model [7, 31 and 41]. There are several advantage of model-based CF algorithms: it improves prediction performance, it is scalable to the actual dataset and easily avoid over fittings. However, several limitations are also existed in model-based CF techniques: it suffer from the sparsity problem so that it is unable to generate reasonable recommendations for those users who provide no ratings and it is difficult to add data to model based systems due to inflexible.

Model-based CF methods, such as Bayesian models, clustering models, and dependency networks, have been explored to solve the shortcomings of memory-based CF algorithms. Usually, classification algorithms are unsupervised learning algorithms and designed to cluster objects into different categories. Clustering algorithm can be used as CF models if the user ratings are categorical, and regression models and Singular Value Decomposition (SVD) methods and be used for numerical ratings. Clustering algorithm, such as K-Means, is used to cluster users or items in groups. Then, the conditional probability of ratings for an item can be calculated based on their group information [7, 23, 31, 41 and 43]. Naive Bayes, matrix factorization that finds common factors that can be the underlying reasons of the ratings given by users, probabilistic matrix factorization which extends MF into the probabilistic framework are model-based CF models that make recommendations [6, 27, 31 and 55].

2.2 Content based recommender system

Content based recommender system make use of user profiles or item descriptions for recommendation [5] In order to provide relevant information to the user, a user profile has to be created using web usage mining or information retrieval methods with attributes and features of the items [37]. Content based recommendation system filters items based on the similarity of the contents the user is interested in. The utility u(c,s) of an item s for a user c is estimated based on the utilities u(c,s_i) assigned by user c to items $s_i \in S$ that are similar to item s that can be derived quantitatively based on the metadata of the item[26].

There are several advantage of content based recommender system: it provides users independence through exclusive ratings which are used by the active user to build their own profile, it provide transparency to their active user by giving explanation how recommender system works and it is adequate to recommend items not yet placed by any user. This will be advantageous for new user [41]. However, several limitations are also existed in content based recommender system like: it advocate the same types of items because of that it suffers from an overspecialization problem, it is harder to acquire feedback from users in CBF because users do not typically rank the items and therefore, it is not possible to determine whether the recommendation is correct and it is a difficult to generate attributes of items [31, 41].

2.3 Hybrid recommender system

Hybrid recommendation system is a method which combines content based with collaborative filtering techniques to gain better prediction or recommendation performance [1, 10, 35, and 47]. Content-based recommender systems make recommendations by analyzing descriptions of item, user profile, preferences and finding regularities in the content [5, 35]. A content-based recommender then uses heuristic methods or classification algorithms to make recommendations [35]. Content-based techniques have the start-up problem, in which they must have enough information to build a reliable classifier. Also, they are limited by the features explicitly associated with the objects they recommend [31] while collaborative filtering can make recommendations without user or item content information. Also, content-based techniques have the overspecialization problem, that is, they can only recommend items that score highly against a user's profile or his/her rating history [15, 34, 37, 42 and 44].

Hybrid approaches can be implemented in various ways: Implement collaborative and content-based methods individually and aggregate their predictions, Integrate some content-based characteristics into a collaborative approach, Comprise some collaborative characteristics into a content based approach, and Construct a general consolidative model that integrate both content based and collaborative characteristics [11, 41]. Cold start and the sparsity are common problems in recommender systems which are resolved by using hybrid methods.

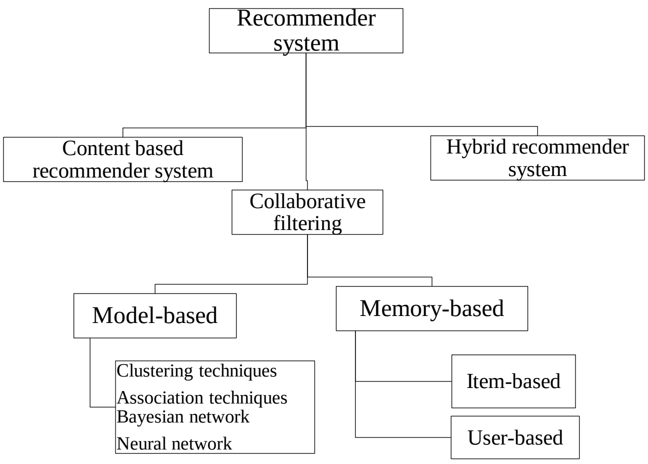

Fig 1.1 Summery of recommendation techniques

3. DEEP LEARNING APPROACHES FOR RECOMMENDER SYSTEM

Deep learning has emerged as a new area of machine learning and data mining research area [13, 52]. Deep learning can be trained either by supervised or unsupervised approaches [31,32] which consist of several layers of processing that form a hierarchy: each subsequent layer extracts a progressively more abstract representation of the input data and builds upon the representation from the previous layer, typically by computing a nonlinear transformation of its input. The parameters of these transformations are optimized by training the model on a dataset [31, 32]. The model experiencing a phase of rapid growth due to its strong performance in a number of domains such as computer vision and audio [17, 38 and 54], speech recognition [30] and Language Processing [4, 53]. Deep learning models have shown their effectiveness for various NLP task including semantic parsing [49], machine translation [12], sentence modeling [36], and various NLP tasks [9]. Generally speaking, deep architecture models consist of multiple layers and can learn a hierarchy of features from low level features to high level ones.

3.1 Basic terminologies of deep learning

Deep belief network (DBN): Probabilistic generative models composed of multiple layers of stochastic, hidden variables. The top two layers have undirected, symmetric connections between them. The lower layers receive top-down, directed connections from the layer above [31, 32].

Boltzmann machine (BM): A network of symmetrically connected, neuron-like units that make stochastic decisions about whether to be on or off [31, 32].

Restricted Boltzmann machine (RBM): A special BM consisting of a layer of visible units and a layer of hidden units with no visible-visible or hidden-hidden connections [31, 32].

Deep Boltzmann machine (DBM): A special BM where the hidden units are organized in a deep layered manner, only adjacent layers are connected, and there are no visible-visible or hidden-hidden connections within the same layer [31, 32].

Deep neural network (DNN): A multilayer network with many hidden layers, whose weights are fully connected and are often initialized (pre-trained) using stacked RBMs or DBN [31, 32].

Deep auto-encoder: A DNN whose output target is the data input itself, often pre-trained with DBN or using distorted training data to regularize the learning [31, 32].

Distributed representation: A representation of the observed data in such a way that they are modeled as being generated by the interactions of many hidden factors. A particular factor learned from configurations of other factors can often generalize well. Distributed representations form the basis of deep learning [31, 32].

3.2 Architectures of Deep Learning

Depending on how the architectures and techniques are intended for use deep learning can be categorized in to **Generative deep architectures,** which are intended to characterize the high-order correlation properties of the observed or visible data for pattern analysis or synthesis purposes, and/or characterize the joint statistical distributions of the visible data and their associated classes. In the latter case, the use of Bayes rule can turn this type of architecture into a discriminative one [3, 22, 29, 31 and 32].

Discriminative deep architectures, which are intended to directly provide discriminative power for pattern classification, often by characterizing the posterior distributions of classes conditioned on the visible data [3, 22, 29, 31 and 32].

Hybrid deep architectures, where the goal is discrimination but is assisted (often in a significant way) with the outcomes of generative architectures via better optimization or/and regularization, or discriminative criteria are used to learn the parameters in any of the deep generative models [3, 22, 29, 31 and 32].

3.3 Deep Neural Networks (DNN)

Deep neural network (DNN) is a multilayer percept network with many hidden layers, whose weights are fully connected and are often initialized using stacked RBMs or DBN [31, 32]. The success of DNN is that it can accommodate a larger hidden units and performs a better parameter initialization methods. A DNN with large number of hidden units can have better modeling power. Even the learned parameters of the DNN is a local optimal, the DNN can performs much better than those with less hidden units. However, in order to converge to a local optima, a DNN with large number of units also requires more training data and more computational power. This also explains why DNN becomes popular until recently [31, 32].

3.3.1 Deep Auto Encoder (DAE)

Deep auto-encoder (DAE) is a special type of DNN whose output target is the data input itself, often pre-trained with DBN or using distorted training data to regularize the learning. [48] Proposed a pre-training technique to learn a deep auto encoders with multiple layers. This technique involves treating each neighboring set of two layers as a restricted Boltzmann machine. In this manner, the pre-training procedure approximates a good parameter initialization. Then, they use a back-propagation technique to fine-tune the pre-trained model [31, 32].

3.4 Convolutional Neural Network (CNN)

Convolutional Neural Network is a type of deep learning model with each module consisting of a convolutional layer and a pooling layer. These modules are often stacked up with one on top of another, or with a DNN on top of it, to form a deep model. As shown in the figure below the convolutional layer shares many weights, and the pooling layer subsamples the output of the convolutional layer and reduces the data rate from the layer below. The weight sharing in the convolutional layer, together with properly chosen pooling schemes, endows the CNN with some invariance properties. CNN is the most successful deep models in the application of Computer Vision and are biologically-inspired variants of MLPs [31, 32].

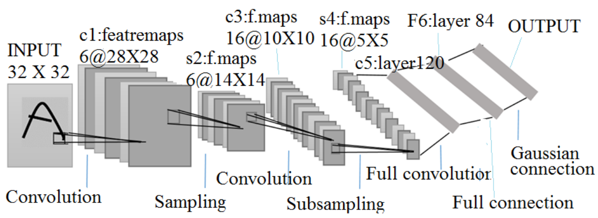

Figure 2.1: Architecture of CNN

As a class of deep models for learning features, the Convolutional Neural Networks (CNN) learns a hierarchy of increasingly complex features. Without building handcrafted features, these methods utilize layers with convolving filters that are applied on top of pre-trained word embedding. Moreover, in benefiting from the shared weights, CNNs have fewer parameters than traditional feed-forward neural networks [31, 32].

4. SURVEY OF DEEP LEARNING TECHNIQUES ON RECOMMENDER SYSTEM

Deep learning has recently been proposed in building a recommender systems both for collaborative and content based approaches [10, 40]. Some of the works are presented below:

4.1 Restricted Boltzmann Machines for Collaborative Filtering

Restricted Boltzmann machine (RBM) is a special Boltzmann machine (BM) which consisting of a layer of visible units and a layer of hidden units with no visible- visible or hidden-hidden connections [39]. The model outperforms Matrix Factorization [10, 41, and 43] even though the authors show that RBM can be successfully applied to the recommendation problem and slightly outperforms traditional Matrix Factorization, the proposed model is not deep enough and only consists of 2 layers. Learning deep models has been successfully applied in the domain of modeling temporal data [4, 31] and learning word embedding [25, 31 and 48]. Training a deeper RBM is helpful for capturing hierarchical latent factors of users and items and more accurately modeling ratings. RBM does not make use of content information, such as user profiles or review texts. RBM is typically used with rating data where most ratings are missing and take advantage of this fact for computational tractability. As a result, the proposed model cannot deal with the cold start problem [2, 31], where recommender systems are required to give recommendations to novel users who have no preference on any items, or recommending items that no user of the community has rated yet. However, content information is proved to be effective to reduce the cold start problem [2, 31].

4.2 Collaborative Deep Learning for Recommender Systems

To address the cold start problem, [21] introduces Collaborative Deep Learning to utilize review texts and ratings. [21] introduces Collaborative Deep Learning to integrate a Bayesian Stack De-noise Auto Encoder (SDAE) [40] and Collaborative Topic Regression (CTR) [8] learn latent factors of items from review texts and draw a latent user vector from Gaussian distribution. Collaborative Deep Learning is the first deep model to learn from review texts for recommender systems. It seemly integrates an Auto Encoder and CTR to model ratings. However, Collaborative Deep Learning still has several shortcomings. First, Collaborative Deep Learning only models item review texts. In recommender systems, user provides reviews to express their feelings. These review texts can be utilized to learn preference of users. Second, review texts are represented by using bag of words scheme. As we know, bag of words vectors only convey the frequency of words. If two reviews are semantically related but use different words, Collaborative Deep Learning, which uses bag-of-words, may not consider the two reviews to be similar. The vocabulary in English is very diverse and two reviews can be semantically similar even with low lexical overlap, so semantic meaning is especially important [31] However, semantic meanings, which are essential for reveal user attitudes and item properties, are lost in Collaborative Deep Learning. At last, in Collaborative Deep Learning, word order is ignored. However, in many text modeling applications, word order is extremely important [19]. To further improve the performance of Collaborative Deep Learning, word order should be taken into consideration while modeling review texts.

4.3 Deep Content-based Music Recommendation

It is well known that Collaborative Filtering can generally outperform content-based methods [20]. However, it is not valid when recommending items that have not been consumed before. To alleviate the cold start problem in music recommendation, some recommender systems recommend music based on metadata, such as genre, artist and album [51] but, the recommendation results are predictable and not useful. Recommender systems should recommend items that users are unknown of. A better approach is to analyze music signals to recommend similar songs users have previously listened to [31, 51]. To capture high level features from music signals, the [51] introduce a deep CNN model to learn latent factors from music signal that demonstrated effective features of music can be learned from music signals via deep models. However, there are still several potential flaws. First, the proposed model employs latent factors learned from WMF as ground truth to train a deep CNN. Although WMF is efficient to learn latent factors from implicit feedback, features approximated by WMF is still is not accurate enough to be used as ground truth to train CNN [51]. Meta data, such artists, genre or album, is not utilized. Although CNN can capture patterns existing in music signals, metadata is still informative and can be fused to learn item features [20, 31 and 51].

5. CONCLUSION

In this article, we presented traditional recommender system and techniques involved in deep learning recommender system. Then, a survey and critique of deep learning on recommender systems are provided. Based on the comparative study of traditional techniques of recommender system we can conclude that using hybrid approach would give better accuracy and performance. The review also reveals that a lot of improvement can still be done in content based and collaborative filtering recommendation techniques to obtaining a better results. Due to the limitation of the traditional recommendation methods in obtaining accurate result a deep learning approach is introduced that will enable the model to learn different features of users and items automatically to improve accuracy of recommendation. With the help of the advantage of deep learning in modeling different types of data, deep recommender systems can better understand users demand and further improve quality of recommendation. Even though deep learning poses a great impact in various areas, a lot of improvement can still be done in applying the model to a recommender systems to improve accuracy of recommendation. This paper is a preliminary study of the work optimal recommendation of business opportunity for young African entrepreneurs by applying hybrid deep architecture through enhanced optimization criteria.

6. ACKNOWLEDGMENTS

This work is supported by ERMIT Intra -ACP project

7. REFERENCES

[1] Alexander Tuzhilin, Gediminas Adomavicius. 2005. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions

[2] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. 2002. Methods and metrics for cold-start recommendations. ACM.

[3] Bengio Y. 2009. Learning deep architectures for AI. Foundations and Trends in Machine Learning

[4] Bengio, Y., Ducharme, R., Vincent, P. and Jauvin. 2000. A neural probabilistic language model. NIPS.

[5] Bigdeli, E., and Bahmani, Z. 2008. Comparing accuracy of cosine-based similarity and correlation-based similarity algorithms in tourism recommender systems. In Management of Innovation and Technology.

[6] Bin Shen, Xiaoyuan Su, Russell Greiner, Petr Musilek, and Corrine Cheng. 2003. Discriminative parameter learning of general bayesian network classifiers. In Tools with Artificial Intelligence, IEEE

[7] C. Basu, H. Hirsh, and W. Cohen. 1998. Recommendation as classification: using social and content-based information in recommendation. AAAI.

[8] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. International conference on Knowledge discovery and data mining. ACM.

[9] Cicero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks.

[10] D. Agarwal and B.-C. Chen. 2009. Regression-based latent factor models. In KDD.

[11] F.O. Isinkayea, Y.O. Folajimib, B.A. Ojokohc. 2015. Recommendation systems: Principles, methods and evaluation Egyptian Informatics Journal.

[12] Fandong Meng, Zhengdong Lu, Mingxuan Wang, Hang Li, Wenbin Jiang, and Qun Liu.2015. Encoding source language with convolutional neural network for machine translation.

[13] G. Hinton, S. Osindero, and Y. The. 2006. A fast learning algorithm for deep belief nets. Neural Computation.

[14] G. Salton and M. McGill. 1983. Introduction to Modern Information Retrieval, McGraw-Hill.

[15] Gediminas Adomavicius and Alexander Tuzhilin. 2005. toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans.

[16] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks.

[17] Goffery, Simon and Yee. 2006. A fast learning algorithm for DB nets

[18] Graham W Taylor, Geoffrey E Hinton, and Sam T Roweis. 2006. Modeling human motion using binary latent variables. In Advances in neural information processing systems,

[19] Hanna M Wallach.2006. Topic modeling: beyond bag-of words. ACM.

[20] Hanna M Wallach.2006. Topic modeling: beyond bag-of-words. International conference on Machine learning. ACM.

[21] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. ACM.

[22] Heigold, G., Ney, H., Lehnen, P., Gass, T., Schluter, R. 2011. Equivalence of generative and log-liner models. IEEE.

[23] J. Breese, D. Heckerman, and C. Kadie. 1998. Empirical analysis of predictive algorithms for collaborative filtering. Uncertainty in Artificial Intelligence.

[24] J.S. Breese, D.Heckerman, and C.Kadie. 1998. Empirical analysis of predictive algorithms for collaborative filtering. On Uncertainty in Artificial Intelligence.

[25] Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. Computational Linguistics.

[26] K. Lang 1995. Newsweeder learning to filter netnews. In ICML.

[27] Koji Miyahara and Michael J Pazzani. 2000. Collaborative filtering with the simple Bayesian classifier. In Topics in Artificial Intelligence.Springer.

[28] L. Hu, J. Cao, G. Xu, L. Cao, Z. Gu, and C. Zhu. 2013. Personalized recommendation via cross-domain triadic factorization. In WWW.

[29] LeCun Y., Chopra S., Ranzato, M., and Huang, F. 2007. Energy-based models in document recognition and computer vision. ICDAR.

[30] Lee, Roger,Pangenstl and dra. 2009. CDBN for scalable unsupervised learning of hierarchical representation.

[31] Lei zheng. 2016. A survey and critique of deep learning on recommender systems. Department of computer science.

[32] Li Deng, Dong Yu. 2014. Deep Learning: Methods and Applications. Foundations and Trends in Signal Processing

[33] M. J. Pazzani. 1999. A framework for collaborative, content-based and demographic filtering. Artificial Intelligence Review.

[34] M. K. Condiff, D. D. Lewis, D. Madigan, and C. Posse. 1999. Bayesian mixed-effects models for recommender systems. Recommender Systems: Algorithm and Evaluation.

[35] M. Pazzani and D. Billsus. 1997. Learning and revising user profiles: the identification of interesting web sites. Machine Learning.

[36] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences.

[37] Nymphia, SatishKumar. 2016. Survey on Content Based Recommendation System. Department of Information Technology. University of Mumbai PIIT.

[38] Oliver, N., Garg, A., and Horvitz, E. 2004. Layered Representations for Learning and Inferring Office Activity from Multiple Sensory Channels. Computer Vision and Image Understanding.

[39] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. 1994. An open architecture for collaborative filtering of netnews. ACM on Computer Supported Cooperative Work.

[40] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and PierreAntoine Manzagol. 2010. Stacked denoising autoencoders, learning useful representations in a deep network with a local denoising criterion. The Journal of Machine Learning Research.

[41] Poonam B. Thorat, R. M. Goudar, Sunita Barve. 2015. Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System, International Journal of Computer Applications.

[42] R. Salakhutdinov and A. Mnih. 2007. Probabilistic matrix factorization. In NIPS.

[43] Recommender systems A Computer Science Comprehensive Exercise Carleton College, Northfield, MN. http://www.cs.carleton.edu

[44] Robin Burke. 2002. Hybrid Recommender Systems: Survey and Experiments. DePaul University

[45] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. 2007. Restricted Boltzmann machines for collaborative filtering. ACM. Machine learning.

[46] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In UAI.

[47] T. Chen, W. Zhang, Q. Lu, K. Chen, Z. Zheng, and Y. Yu. 2012. Svd feature: a toolkit for feature-based collaborative filtering. JMLR.

[48] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean.2013. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems.

[49] Wen-tau Yih, Xiaodong He, and Christopher Meek. 2014. Semantic parsing for singlerelation question answering. In Proceedings of ACL, 2014.

[50] Xiaoyuan Su and Taghi M. Khoshgoftaar. 2009. A Survey of Collaborative Filtering Techniques. Department of Computer Science and Engineering, Florida Atlantic University.

[51] Xinxi Wang and Ye Wang. 2014. Improving content based and hybrid music recommendation using deep learning. ACM on Multimedia.

[52] Y. Bengio. 2009. Learning deep architectures for AI. Foundations and Trends in Machine Learning.

[53] Yann LeCun, Y-Lan, Jie, Fu. 2007. Unsupervised learning of invariant feature hierarchical with application to object recognition.

[54] Yann LeCun. 1998. Gradient based learning applied to document recognition

[55] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems.

[56] Yoshua Bengio. 2007. Learning deep architectures for AI. Technical report, Dept. IRO, Université de Montreal.