2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
Banff Center, Banff, Canada, October 5-8, 2017

E-Learning Recommendation Framework Based on Deep Learning

Xiao Wang*†, Yuanyuan Zhang‡, Shengnan Yu§, Xiwei Liu*¶ Yong Yuan*¶ and Fei-Yue Wang*
*State Key Laboratory of Managment and Control for Complex Systems
Institute of Automation, Chinese Academy of Sciences, Beijing,100080 China
Email:xiwei.liu@ia.ac.cn(Corresponding),yong.yuan@ia.ac.cn, feiyue.trans@gmail.com
†Department of Computer Science and Technology
Xi'an Jiaotong University, Xi'an, 710049 China
Email:gzzkdx867155@stu.xjtu.edu.cn
‡Academy of Computer Science
Sichuan University, Chengdu, 610065 China
Email:2014141462362@stu.scu.edu.cn
§Department of Computational Mathmatics
Ocean University of China, Qing'dao,266100 China,
Email:shengnanyu_ocu@126.com
¶Qingdao Academy of Intelligent Industries
Qing'dao,266109 China

***Abstract*—In the paper, considering the limitation of effective method in E-learning area, a recommendation framework for E-Learning based on deep learning is proposed. Our model is based on deep learning, which has strong capability to learn from large-scale data.It has some improvements than previous methods. First, it is based on the conventional K-Nearest Neighbor(KNN) method to train a model, thus its accuracy is guaranteed. Second, it can recommend the new item whose similarity can not be calculated. Third, it greatly reduces the heavy burden for a running system, which is useful in real practice of recommendation systems. In conclusion, the proposed framework can offer a new recommendation method for more personlized learning in the future.**

I. Introduction

In the recent 30 years, a lot of researches [1]–[7] have been carried for the E-learning area. Compared with the conventional learning method, E-learning offers an online platform education, which is not limited by space difference. Therefore,the spread of E-learning makes better education for worldwide students.Therefore, more and more web-based online learning systems [8]–[10] are developed and used widely. With time passed by, to make online education more effective,personlized education through e-learning gradually developed.New E-learning focus on the individual learning, hoping to offer a more personalized studying platform for learners. Therefore, Computerized Adaptive Testing (CAT) [11]–[14] and Computerized Adaptive Education (CAE) [15]–[17] always draw a lot of attention.

Although great achievements have been made by such models, there still exist problems waiting to be solved:

First, most adaptive learning systems [18]–[22] are based on the Solomon style quantitative table, which most of the parameters are calculated by a simple method. Most importantly, the parameters such as styles and learning habits may not be conclusive and incisive enough to capture the real features of learners. Since features of a person when he is learning is hard to capture a full picture of him to give exact recommendations, thus such recommendation systems usually recommend based on types instead of customizing for individual.

Second, some web-based learning systems [22]–[25] only focus on the combination of hypermedia and textbooks. Though it is true that hypermedia can help students to learn in a more vivid method, simple combinations of them can not really improve the students' ability in a maximum efficiency. In addition, the recommendation of such complex learning systems do not apply a very effective method, which means they are more likely a fixed learning sequence for students.

Third, some recommendation system based on similarity [26]–[29] also have their limitations. Since the recommendation is laid on similarity between users, the accurate recommendation should be based on the data from a large number of users. This leaves a problem for such systems. First of all, if there exist no records for some items in the system, the similarity can not calculate, which will result recommendation for the item will not carried on anymore. In addition, if there exists too many users in the system, the amount of calculation will increase rapidly, which leave a heavy burden for a running system.

To deal with such problems, we propose an e-learning recommendation framework based on deep learning. It can have some benefits than previous models:

1) It is based on the K-Nearest Neighbor(KNN) recommendation methods, thus its accuracy for most recommendations is reliable.

2) Through deep learning, when it is successfully trained, we can decide whether to recommend the item or not without calculate its similarity with others. This brings at least two

advantages. First, it will greatly reduce the burden of the system, since the method needs no amount of time and space consuming for similarity calculation. Second, it can recommend newly introduced item, whose similarity can not be calculated since no previous records can be traced.

3) Compared with conventional KNN method, the calculation amount is smaller than that of KNN, since KNN has to calculate the similarity at each time. Therefore, the heavy computation burden of KNN may not be suitable for a running system, but our model can react immediately after training finished.

To better present our framework,our paper is arranged as follows. Section II reviews the mechanism of conventional K-Nearest Neighbor (KNN) Collaborative Filtering Algorithm and how it used in our framework. Section III briefly reviews deep learning and explains the detail of implementing our deep learning recommendation framework, as well as how to test the performance of our framework.Section IV concludes the working procedure of our framework. Section V concludes the paper.

II. K-Nearest Neighbor (KNN) Collaborative Filtering Algorithm of Recommendation Systems

To discuss how K-Nearest Neighbor (KNN) works in our framework. First, we briefly review the mechanism and procedure of KNN in Section II-A. Then, how the KNN works in our framework is addressed in Section II-B

A. K-Nearest Neighbor(KNN) Algorithm Review

To recommend more precisely, a lot of recommendation systems [18], [26], [27], [30], [31] are raised and employed in different areas. Among all of them, K-Nearest Neighbor (KNN) algorithm [32], [33] is one of the successful methods which has made achievements at some point.

Compared with other algorithms for recommendation, KNN has some features as follows. First of all, its recommendation results are based on the nearest neighbors. Therefore, most of its recommendations are reliable. Second, since it is based on the similarity, its mechanism is easy to understand.

As discussed in [34], KNN has two main different methods. One is based on users, while the others are based on the items.We will then explain common KNN working procedure in the rest of this part.

First, we need to calculate the similarity among users and also among the items in our database. Among several different similarity methods, we adopt the cosine similarity method [35]. Considering the method can better measure two vectors' similarity by their angles, similarity of users and items is suitable for the method. The detail of the cosine similarity is explained in Eq.(1).

$$sim(i,j) = \frac{\sum_{k=1}^{N} X_k Y_k}{\sqrt{\sum_{k=1}^{N} (X_k)^2}\sqrt{\sum_{k=1}^{N} (Y_k)^2}} \tag{1}$$

where $sim(i,j)$ denotes the similarity between i and j, whose range is in [-1,1]; N denotes the number of features that V_1 and V_2 have in common; X is the vector formed by common feature shared by V_1 and V_2 from V_1;Y is the vector formed by common feature shared by V_1 and V_2 from V_2.Besides, V_1 is the feature vector of i, while V_2 is the feature vector of j.

Second, we will go through to calculate the similarity of each two objects. After that,for each item, we will pick up top k highest similarity objects as the neighbors of object.After this step finished, all the neighbors and similarities of each objects have been accessed.

Third, we can recommend as a sequence by the recommendation grade, which is calculated in Eq.(2) and Eq.(3), which is user-based and item-based KNN, respectively.In addition,the higher the score is,will we be more likely to recommend it.

$$\hat{r_{u,i}} = \frac{\sum_{v\in N(u;i)} sim(u,v) * r_{v,i}}{\sum_{v\in N(u;i)} sim(u,v)} \tag{2}$$

where $\hat{r_{u,i}}$ is the recommendation score calculated from KNN, $N(u;i)$ denotes the set that consists of neighbors of u who have connections with i; $sim(u,v)$ denotes the similarity u and v; $r_{v,i}$ denotes the score from v to i, whose calculation method is not the same as $\hat{r_{u,i}}$ and detail will be addressed in Section II-B.

$$\hat{r_{u,i}} = \frac{\sum_{j\in N(i;u)} sim(i,j) * r_{u,j}}{\sum_{j\in N(i;u)} sim(i,j)} \tag{3}$$

where $\hat{r_{u,i}}$ is the recommendation score calculated from KNN, $N(i;u)$ denotes the set that consists of neighbors of i who have connections with u; $sim(i,j)$ denotes the similarity i and j; $r_{u,j}$ denotes the score from u to j, whose values are not the same as $\hat{r_{u,i}}$ and detail will be addressed in Section II-B.

B. K-Nearest Neighbor(KNN) in Our Framework

As it addressed in Section II-A, the user-based and item-based KNN have been reviewed. But how they work in our framework needs to be further explained.

It is widely known that the KNN has its own drawbacks: when the item sets or the user sets become too large, the amount of calculation will be too enormous for a running system. Therefore, to avoid the shortcoming of KNN, we will not employ KNN as the instant recommendation method in our framework. Instead, it will be used in a relatively small initial dataset to offer the training set and aim set to train the deep learning neural networks. The detail of what the trainset is will be explained in III-B.

Further more, we will combine user-based and item-based KNN to a more effective KNN. Since the user-based KNN is based on the idea that similar users will tend to have same knowledge for most cases. Therefore, their shortcomings in the learning may be similar. User-based recommendation may be effective to help a student to grasp the most difficult knowledge point for most students in his level. For item-based KNN, it's based on the idea that knowledge point with similar user performances can help students' review their unfamiliar point. It is resulted from that items in the same level should be recommended to help students to improve him to a new level.

When it comes to the combination of two types of KNN, one fact we should notice is that the recommendation item of user-base method and item-based are not the same. But for most cases, they will have intersections. For the intersection part, we take the average score of two methods as its score. For the left part, the score will be calculated by two methods, respectively.

Finally, for $r_{v,i}$ in Eq.(2) and $r_{u,j}$ in Eq.(3), their calculation is based on the user performance. For example, if user v submit correct answer for item i, $r_{v,i}$ will be set as 0. Since he has grasped the point, such items will be less likely to recommend. Otherwise, if he can not answer right, similar items will be recommended to help him improve and grasp the knowledge point.

III. Deep Learning Recommendation Framework for E-Learning

In the section, we will first briefly review the deep learning method and choose our neural networks in Section III-A. Furthermore, we will explain the detailed settings for our deep learning neural network in Section III-B. In the end, we propose a method to test the performances of our model in Section III-C.

A. Deep Learning Method

Deep learning now receives a lot of researchers' interest, which is derived from the artificial neural network [36]–[39]. It can learn the most important features by a model of multiple hidden layers and trained from a large dataset. Compared with parameter models, it greatly improves accuracy.

For deep learning, there are a lot of neural networks being proposed and implemented. In this paper, we recommend to use the Gated Recurrent Unit (GRU) neural networks [40]. GRU is a kind of recurrent neural network (RNN) [36], [37]. As shown in Fig.1, recurrent neural networks allow connections in the neurons of the same layer. This creates an internal state of the network that enable memory effect in a neural network.

Long Short Term Memory (LSTM) neural network is one of RNN model, which is mostly wide known [39]. GRU is a simpler kind of LSTM neural network that shares many properties of LSTM. Compared with the LSTM, it is much easier to set parameters to obtain better performances.

Conventionally, introducing deep learning technique to a new area is to apply its strong capability of learning from empirical data to get a more accurate special function for some cases. More importantly, with the development of deep learning optimized algorithm, multi-layer neural networks get more and more accurate in its performance in different areas

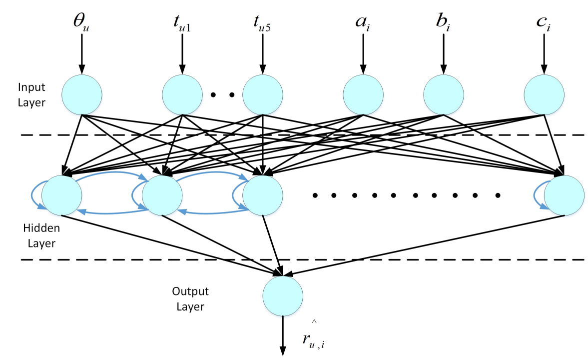

Fig. 1. The structure of GRU networks for recommendation in E-learning.

[41]–[43]. For our model, what we tend to learn the special function $f(\cdot)$ is shown in Eq.(4).

$$\hat{s}_{u,i} = f(\theta_u, t_{u1}, t_{u2}, t_{u3}, t_{u4}, t_{u5}, a_i, b_i, c_i) \tag{4}$$

where $\hat{s}_{u,i}$ denotes the recommendation score calculated by deep learning method; $f(\cdot)$ is the special function that deep learning should learn; θ_u is the evaluated ability for a person from Computerized Adaptive Testing (CAT), whose deep learning based method is proposed in [44]; $t_{u1}, t_{u2}, t_{u3}, t_{u4}, t_{u5}$ are the correct rate calculated from testing items which difficulty levels range from 1 to 5; a_i, b_i, c_i are three important parameters of each item, whose detail and calculated methods is explained in [14], [44].

The output of GRU is shown in Eq.(5), whose detail please refer to [40].

$$\begin{cases} z_{j,k,t} = \sigma(W_z \mathrm{I}_{j,t} + U_z \mathrm{O}_{j,t-1}) \\ r_{j,k,t} = \sigma(W_\mathrm{r} I_{j,k,t} + U_r \mathrm{O}_{j,k,t-1}) \\ \hat{h}_{j,k,t} = \tanh(W I_{j,k,t} + U(r_{j,t} \odot \mathrm{O}_{j,t-1})) \\ \mathrm{O}_{j,k,t} = (1 - z_{j,k,t})\mathrm{O}_{j,k,t-1} + z_{j,k,t}\hat{h}_{j,k,t} \end{cases} \tag{5}$$

B. Detail of Deep Learning Recommendation Framework

The training process of the neural network can be regarded as an optimization problem.First, we propose our measurement of GRU performance in Section III-B1.Second, some decision variables or parameters settings are discussed in Section III-B2. Third,algorithms applied in GRU is clearly listed in Section III-B3. Besides, the detail of the training dataset will be addressed in Section III-B4.

*1) **Performance Index**:* As suggested in [45], [46], Mean Squared Error(MSE) is widely adopted. We also use it to test the performance of our model. More precisely, the performance index compares the recommendation score evaluated by GRU neural network and KNN. Eq.(6) shows the definition of our performance index.

$$ming(w) = \frac{1}{L}\sum_{u=1}^{L}\frac{1}{M_u}\sum_{i=1}^{M_u}(\hat{r}_{u,i} - \hat{s}_{u,i})^2 \tag{6}$$

where $g(w)$ denotes the mean-squared-error in validation set; L denotes the number of students in the dataset; M_u denotes

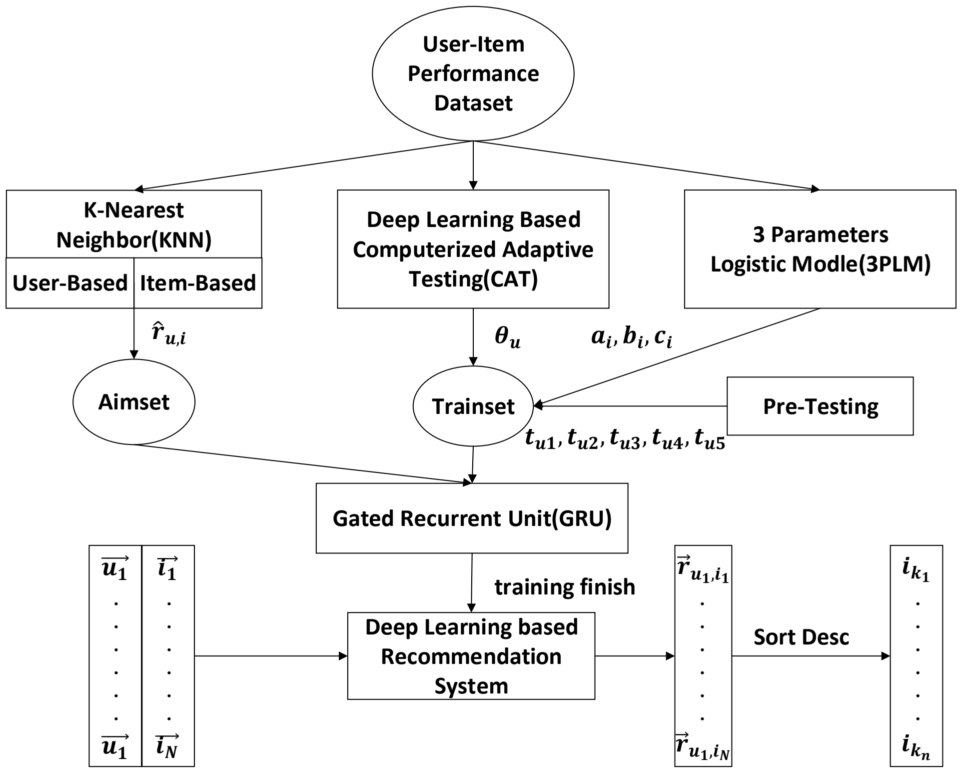

Fig. 2. The structure of our Framework

the number of items for user u; $\hat{r_{u,i}}$ denotes the recommendation score based on KNN; $\hat{s_{u,i}}$ denotes the recommendation score based on Deep Learning method.

*2) **Parameters Settings**:* For detail of our model,we set as follows.

(1) The type of transfer function.

As suggested in classical masterpiece [46], we choose sigmoid function, and all the outputs will be normalized into range [0, 1].

(2) The structure of neural networks

The structure of a neural network always plays an important role on the performance of model. Therefore, we suggest to decide the structure by careful testing of different structures.

(3) The weight coefficients whose values will be learned from training.

Because the final weight coefficients in the network will be decided by learning, so what we need to decided is the initial weight coefficients. As suggested in [46],initial weight coefficients of our network are randomly generated from a uniform distribution in the range [0, 1].

*3) **Training Algorithms**:* As we all know, updating coefficients while avoiding gradient vanish and find global optimal solution while avoiding local optimal solution are always the main concern for our choice of algorithms.

First, when it comes to updating weight coefficients, classical backpropagation algorithm [45], [46] is adopted and stochastic gradient descent algorithm [46] together with the adaptive learning rates trick proposed in [47] is applied in model training.

As for global minimum, cross-validation method [48] is an ideal choice. The method can make sure the error of validation set and training set will both decrease. Besides, we choose validation split as 0.3, which means nearly 30% of the training set is used for validation.

*4) **Training Dataset**:* To train our model,as shown in Eq.(4) and Fig.1, we need to get the 8 inputs and 1 output as data pair to train our models.

First, $\hat{r_{u,i}}$ in the aimset can be accessed by K-Nearset Neighbor(KNN), whose implementation has been addressed in Section II.

Second, θ_u is calculated by deep learning based on Computerized Adaptive Testing(CAT), whose detail is explained in [44].

Third, $t_{u1}, t_{u2}, t_{u3}, t_{u4}, t_{u5}$ are the correct rate of testing samples for students, whose difficulty level ranges from 1 to 5.

Fourth, a_i, b_i, c_i denote three parameters of item i by 3 parameters logistic model (3PLM), whose working mechanism has been careful explained in [44].

The detail of our training dataset can be viewed in our framework in Fig.2.

C. Performance Testing of Deep Learning Recommendation Framework

After a recommendation system is proposed, the main concern for it is how to judge the performance of the system. In addition, compared with conventional methods, the advantages of new method are always the focus. To answer such questions, we propose two methods to judge the performance of our framework: Offline Testing and A/B Testing.

First of all, Offiline Testing is explained. Before training, we need to split some users in known dataset as validation set.After training finished, since such users having been used to train the model, we can compare the Mean-Squared-Error(MSE) of recommendation score calculated from KNN and Deep Learning.Also,this testing can be calculated by Eq.(6).If the MSE is in an acceptable degree, it is reasonable to draw a conclusion that our model can recommend such items and users whose similarities are not calculated.

In addition, when the Offline Testing finished, A/B Testing should be made to make sure our model can really contribute to E-learning. The idea is that divide a group of students of the same level to two same groups. And they will take a test before they use the recommendation system. Then one group will use the system while the other just learn by a fixed sequence. After learing finished, all students will have a test which is the same difficulty level of the first test. we compare the average score and score distribution of two groups. If the average score increase of the group with recommendation system is larger than the other groups and the score distribution of the group converges more to higher scores, our model really contributes to personlized E-learning.

IV. Working Procedure of Our Framework

To demonstrate our framework more clearly, we discribe the working procedure of our framework in detail as follows.

First, our recommendation system is based on similar groups and similar items. Therefore, we should point out we should build neural networks for a similar group on similar items. In other words, we recommend two methods to make our system work more precise. First, we can filter users to similar groups by their information which are collected from their bevavior while using the system, such as their grades and so on. For different groups, different deep-learning based models should apply to them. Second, items with different knowledges should based on different deep learning based models, which will make the recommendation more effectively.

Second, students need to take a pre-testing,in which their performance on a special designed will be recorded and taken as their features for the recommendation.

Third, by the pre-testing and existed user-item performance, we can offer the training samples and aiming samples to train our neural networks. They can be divided to three main parts. First of all, we can offer the recommendation score of each users as aiming sample by combined K-Nearest Neighbor(KNN). The detail mechanism and implementation of the method have been clearly addressed in Section II. What's more, based on the dataset and deep learning based on Computerized Adaptive Testing(CAT) proposed in [44], the evaluated ability of different users will be given. The ability value is also an important feature of a user. Last of all, the parameters related to item can be given by conventional 3 Parameters Logistic Model, whose detail is given in [44]. This 3 parameters are the most important feature of an item in testing.

When the aimset and trainset preparing finish, we can train our model specificlly for a group and a set of items, whose implementation is addressed in Section III.After the training finished, when offering vector $\vec{u_1}$ of user 1, which includes $\theta_u, t_{u1}, t_{u2}, t_{u3}, t_{u4}, t_{u5}$; and vector $\vec{i_k}$, which includes a_i, b_i, c_i, our model can quickly calculate recommendation scores. Then, we can recommend items for user 1 by this scores in a descending order.

To learn the working procedure of our framework in a more clear way, Fig.2 shows the working procedure briefly.

V. Conclusion

In this paper, we propose a deep learning recommendation framework for E-learning, which is first applied in E-learning area. Through the testings and applications of our framework have not been carried. The advantages of our framework are clear.First, although the training process is time-consuming and complex, it is suitable for running system to react immediately instead of the conventional recommendation systems. Second, after training finished, it can recommend new items whose similarity are unknown. Meanwhile, we admit large-scale data set for KNN is necessary before deploy the deep learning framework.

Furthermore, based on previous proposed Computerized Adaptive Testing, we can develop a system where students can learn and test at the same time. When the system is successfully developed, we believe it would greatly contribute to the efficiency of students' learning compared with conventional E-learning. Also, this work can contribute to the idea of paralleled education by its fast speed and specialization.

Acknowledgment

This work was supported in part by National Natural Science Foundation of China (Grant No.61533019; Grant No.61233001; Grant No.71232006) and Science and Technology Innovation Program of Chinese Academy of Sciences (CAS).

References

[1] T. E. D. Mining, "Enhancing teaching and learning through educational data mining and learning analytics: An issue brief," in *Proceedings of conference on advanced technology for education*, 2012.

[2] L. Aroyo, P. Dolog, G.-J. Houben, M. Kravcik, A. Naeve, M. Nilsson, F. Wild *et al.*, "Interoperability in personalized adaptive learning," *Educational Technology & Society*, vol. 9, no. 2, pp. 4–18, 2006.

[3] M. J. Rosenberg, *E-learning: Strategies for delivering knowledge in the digital age*. McGraw-Hill New York, 2001, vol. 9.

[4] S. Downes, "E-learning 2.0." *Elearn magazine*, vol. 2005, no. 10, p. 1, 2005.

[5] D. R. Garrison, *E-learning in the 21st century: A framework for research and practice*. Taylor & Francis, 2011.

[6] A. T. Bates, *Technology, e-learning and distance education*. Routledge, 2005.

[7] P.-C. Sun, R. J. Tsai, G. Finger, Y.-Y. Chen, and D. Yeh, "What drives a successful e-learning? an empirical investigation of the critical factors influencing learner satisfaction," *Computers & education*, vol. 50, no. 4, pp. 1183–1202, 2008.

[8] P. Brusilovsky and C. Peylo, "Adaptive and intelligent web-based educational systems," *International Journal of Artificial Intelligence in Education (IJAIED)*, vol. 13, pp. 159–172, 2003.

[9] G. Weber and P. Brusilovsky, "Elm-art: An adaptive versatile system for web-based instruction," *International Journal of Artificial Intelligence in Education (IJAIED)*, vol. 12, pp. 351–384, 2001.

[10] C.-M. Chen, "Intelligent web-based learning system with personalized learning path guidance," *Computers & Education*, vol. 51, no. 2, pp. 787–814, 2008.

[11] H. Wainer, N. J. Dorans, R. Flaugher, B. F. Green, and R. J. Mislevy, *Computerized adaptive testing: A primer*. Routledge, 2000.

[12] H. Wainer, E. T. Bradlow, and Z. Du, "Testlet response theory: An analog for the 3pl model useful in testlet-based adaptive testing," in *Computerized adaptive testing: Theory and practice*. Springer, 2000, pp. 245–269.

[13] R. Conejo, E. Guzmán, E. Millán, M. Trella, J. L. Pérez-De-La-Cruz, and A. Ríos, "Siette: A web-based tool for adaptive testing," *International Journal of Artificial Intelligence in Education*, vol. 14, no. 1, pp. 29–61, 2004.

[14] S. E. Embretson and S. P. Reise, *Item response theory*. Psychology Press, 2013.

[15] L. Verschaffel, K. Luwel, J. Torbeyns, and W. Van Dooren, "Conceptualizing, investigating, and enhancing adaptive expertise in elementary mathematics education," *European Journal of Psychology of Education*, vol. 24, no. 3, p. 335, 2009.

[16] V. Jones and J. H. Jo, "Ubiquitous learning environment: An adaptive teaching system using ubiquitous technology," in *Beyond the comfort zone: Proceedings of the 21st ASCILITE Conference*, vol. 468. Perth, Western Australia, 2004, p. 474.

[17] A. Paramythis and S. Loidl-Reisinger, "Adaptive learning environments and e-learning standards," in *Second european conference on e-learning*, vol. 1, no. 2003, 2003, pp. 369–379.

[18] A. Klašnja-Milićević, B. Vesin, M. Ivanović, and Z. Budimac, "E-learning personalization based on hybrid recommendation strategy and learning style identification," *Computers & Education*, vol. 56, no. 3, pp. 885–899, 2011.

[19] C. Wolf, "iweaver: towards' learning style'-based e-learning in computer science education," in *Proceedings of the fifth Australasian conference on Computing education-Volume 20*. Australian Computer Society, Inc., 2003, pp. 273–279.

[20] N.-N. Manochehr *et al.*, "The influence of learning styles on learners in e-learning environments: An empirical study," *Computers in Higher Education Economics Review*, vol. 18, no. 1, pp. 10–14, 2006.

[21] P. Dolog, N. Henze, W. Nejdl, and M. Sintek, "Personalization in distributed e-learning environments," in *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*. ACM, 2004, pp. 170–179.

[22] Y. E. A. Mustafa and S. M. Sharif, "An approach to adaptive e-learning hypermedia system based on learning styles (aehs-ls): Implementation and evaluation," *International Journal of Library and Information Science*, vol. 3, no. 1, pp. 15–28, 2011.

[23] H. Chorfi and M. Jemni, "Perso: Towards an adaptive e-learning system," *Journal of Interactive Learning Research*, vol. 15, no. 4, p. 433, 2004.

[24] N. Henze, P. Dolog, W. Nejdl *et al.*, "Reasoning and ontologies for personalized e-learning in the semantic web." *Educational Technology & Society*, vol. 7, no. 4, pp. 82–97, 2004.

[25] E. Brown, A. Cristea, C. Stewart, and T. Brailsford, "Patterns in authoring of adaptive educational hypermedia: A taxonomy of learning styles," *Educational Technology & Society*, vol. 8, no. 3, pp. 77–90, 2005.

[26] A. Nakagawa and T. Ito, "An implementation of a knowledge recommendation system based on similarity among users' profiles," in *Sice 2002. Proceedings of the Sice Conference*, 2002, pp. 326–327 vol.1.

[27] A. Bozzon, G. Prandi, G. Valenzise, and M. Tagliasacchi, "A music recommendation system based on semantic audio segments similarity," in *Iasted International Conference on Internet and Multimedia Systems and Applications*, 2008, pp. 182–187.

[28] T. K. Quan, I. Fuyuki, and H. Shinichi, "Improving accuracy of recommender system by clustering items based on stability of user similarity," in *International Conference on Computational Intelligence for Modelling Control and Automation*, 2006, p. 61.

[29] M. Muozorganero, G. A. Ramezgonzlez, P. J. Muozmerino, and C. D. Kloos, "A collaborative recommender system based on space-time similarities," vol. 9, no. 3, pp. 81–87, 2010.

[30] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*. ACM, 2001, pp. 285–295.

[31] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 143–177, 2004.

[32] G. Wang, "Survey of personalized recommendation system," *Computer Engineering & Applications*, 2012.

[33] A. Albadvi and M. Shahbazi, "A hybrid recommendation technique based on product category attributes," *Expert Systems with Applications*, vol. 36, no. 9, pp. 11 480–11 488, 2009.

[34] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.

[35] A. F. R. Hernandez and N. Y. G. Garcia, "Distributed processing using cosine similarity for mapping big data in hadoop," *IEEE Latin America Transactions*, vol. 14, no. 6, pp. 2857–2861, 2016.

[36] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[37] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[38] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, p. 504, 2006.

[39] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, p. 1527, 2006.

[40] K. Cho, B. V. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," *Computer Science*, 2014.

[41] S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Briefings in Bioinformatics*, p. bbw068, 2016.

[42] R. Socher, Y. Bengio, and C. Manning, "Deep learning for NLP," *Tutorial at Association of Computational Logistics (ACL), 2012, and North American Chapter of the Association of Computational Linguistics (NAACL)*, 2013.

[43] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, and T. N. Sainath, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[44] X. Wang, Y. Zhang, S. Yu, X. Liu, and F.-Y. Wang, "CAT based on Deep Learning(submitted)," *IEEE Transactions on Learning Technologies*, 2017.

[45] P. J. Werbos, "The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting," 1994.

[46] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning representations by back-propagating errors*. MIT Press, 1986.

[47] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computer Science*, 2015.

[48] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *International Joint Conference on Artificial Intelligence*, Conference Proceedings, pp. 1137–1143.