LeCoRe: A Framework for Modeling Learner's preference

Kumar Abhinav
Accenture Labs
Bangalore, India
k.a.abhinav
@accenture.com

Venkatesh Subramanian
Accenture Labs
Bangalore, India
venkatesh.subramania
@accenture.com

Alpana Dubey
Accenture Labs
Bangalore, India
alpana.a.dubey
@accenture.com

Padmaraj Bhat
Accenture Labs
Bangalore, India
padmaraj.bhat
@accenture.com

Aditya D. Venkat
Accenture Labs
Bangalore, India
d.venkat.aditya
@accenture.com

ABSTRACT

Learning & development (L&D) is an important strategic factor for sustainable business growth of any organization. L&D has become an integral part of an organization, considering the fast-paced growth of industries. Success of learning and development program depends on how well it identifies the critical skill gaps in its workforce and bridges those gaps by considering individual learner's strengths and aspirations while recommending learning opportunities. Such recommendation requires a rich information about learners and learning opportunities. In this paper, we propose a framework that recommends learning opportunities to learners based on their preferences. We also propose a way to connect users of similar interests on the platform to improve course engagement. We developed a conversational AI agent that assist the learner in their journey. We evaluated our approach on the dataset consisting of 5,000 learners, and 49,202 courses. Our approach performed significantly better than the baseline approach.

Keywords

Learning & Development, Recommendation, Personalization

1. INTRODUCTION

Learning & Development (L&D) program plays a vital role in the overall talent management of any organization. The primary functions of the L&D program are to - 1. Identify the skills that are needed to achieve business goals, 2. Understand the skill gaps in its workforce, 3. Define a plan to close the gaps, and 4. Successfully deploy that plan. These are the crucial steps to create a strong pipeline of employees with appropriate skills required for current and future business needs. It also helps in understanding employees' learning needs based on their career aspiration and provides them a personalized learning plan. To provide a personalized learning plan, one needs to have access to variety of information such as employees' data, their goals, their position, and preferences. Along with this, one also needs to have an understanding of business requirements, courses available internally and externally, etc. With changing times, large employee base, global footprint, and varied profiles to train, monitoring and processing of data has become difficult. Large amount of learning content is available within the organization along with various other external sources of learning such as Massive Open Online Course (MOOC). Most of the content is not personalized and so the learners spend an inordinate amount of time in identifying the relevant content to leverage. We have used employee and learner interchangeably. A number of studies focus on recommending courses to learners [2] [6]. However, these existing recommendation systems have three major shortcomings - 1. They work on a limited set of information and do not utilize the rich set of information available about learners and courses. These information are dispersed at several enterprise systems making it difficult to consume. 2. They do not consider the scenario of a new learner whose course preference data is not available. 3. They do not utilize the social connections that learners may have for enhanced peer learning with learners of similar interests.

To address these challenges, we propose a recommendation framework that models the learner's behavior from the data available about them on different platforms. The framework is a part of the system "LeCoRe" that helps the learners in selecting the right learning content based on their history. The learners' preference model is built from the courses that learners have registered or completed in the past and their profile information captured through several internal as well as external platforms such as LinkedIn, Accenture People [1] etc.
The main contributions of the paper are as follows:

1. An ensemble based learning content recommender: We propose an ensemble based approach of content and collaborative filtering to evaluate the recommendation

[1]Accenture Internal portal to manage employee's details

framework. The results show a significant improvement over the baseline approach.

2. Conversation AI agent: We propose a conversational AI agent that assist the learner in their journey on the learning platform.

3. A system to connect learners with similar interests: The system also promotes an effective engagement among learners by establishing a connection with other similar learners within the platform.

The proposed system brings three main advantages. Firstly, the system improves the ease with which the learners select the learning content, matching with their interest. Secondly, the right content helps the learners in their career growth. Thirdly, the analytic techniques we have devised make use of much richer and contextual data available about learners. The remainder of this paper is structured as follows: Section 2 discusses the related work study in course recommendation. In subsequent section 3, we discuss the recommendation framework. In Section 4, we discuss the system architecture. We describe our dataset in Section 5 and discuss evaluation methodology and results in Section 6. Section 7 discusses the implementation as a tool. Finally, Section 8 concludes with summary of our findings.

2. RELATED WORK

Several studies have been performed in the area of recommending courses to learners. Aher et al. [6] combined clustering and association rule mining algorithms to recommend courses using historical data. Apaza et al. [3] proposed the course recommendation system based on the topic modeling technique. They computed the semantic similarity between the topics extracted from college course syllabus with the topics of MOOCs based courses and then applied content based approach to recommend relevant online courses to college students. However, it didn't consider the learner's history and lacked personalization. Fatiha et al. [7] applied Case Based Reasoning (CBR) based approach to find courses for learners that best fit their personal interests. They used Levenshtein distance to measure the similarity between the cases' attributes. Piao et al. [1] compared the three user modeling strategies based on job title, education and skills available on user's LinkedIn profiles, for personalized MOOC recommendations. They applied dot product similarity between user and course profiles, and then ranked the user's courses based on the similarity. Jiye et al. [5] conducted an experiment on edX platform to identify the factors that contribute to student enagagement in MOOC discussion forums. Jiezhong et al. [2] analyzed the key factors that influence users' engagement in MOOCs using the data collected from xuetangX, one of the largest MOOCs platform from China. Our approach can be differentiated with the state-of-the-art approaches along three dimensions. Firstly, we are the first to apply Deep Learning based approach to model learner's preference and recommending learning content. The system also addresses the problem of new learners with no learning history. Secondly, our platform provides bot assisted learner journey and also helps the learner to connect to social community to promote interaction among learners. Thirdly, our approach is much more comprehensive and models learner's preference over various dimensions of learner and course profiles.

3. FRAMEWORK

We propose a recommendation framework that models the learner's preference. We apply an ensemble based approach where we combine the predictions of Collaborative filtering and Content based techniques.

1. **Collaborative-based approach**: It is one of the most popular and powerful techniques used in recommendation systems. Collaborative Filtering approach (CF) builds the user's interest by collecting preferences of many other users [22]. We employ three popular collaborative filtering techniques:

    (a) **Singuar Value Decomposition**: It is one of the most popular Matrix factorization based techniques that involves decomposing a sparse user-item matrix into two low rank latent matrices that represents user factors and item factors. The missing ratings are then predicted from the inner product of these two factor matrices [22].

    (b) **Slope-One**: The Slope-One approach considers information from other users who rated the same item and from the other items rated by the same user [4].

    (c) **K-Nearest Neighbor**: K-Nearest Neighbor approach takes into account either the items or the users that are similar. This is captured using similarity metrics like Pearson Correlation, Euclidean Distance etc. It predicts rating of the item given by the user based on the weighted average of top-k similar users.

2. **Content-based approach**: The collaborative based approach considers the rating given by learners for different courses. However, it doesn't consider the learner's profile and the content of the courses. We employ content based approach that considers the personal characteristics of the learner and course information registered or completed by the learner. Personal characteristics of the learner includes Skills (skillset of the learner), Geography (geographical unit of the learner), Experience (years of experience) and Industry. Course information includes the course title, course description, course content type (such as web-based etc.). The title and description of the course is represented as topics vector using Topic Modeling techniques. Topic modeling [17] techniques are probabilistic model that have been used to identify topics within the text documents. Latent Dirichlet Allocation (LDA) [16], one of the popular topic modeling techniques, extracts topic information from unstructured text as probability distribution of words. LDA model is used as feature descriptor for course title, course description and profile description of the learner. We pose it as a regression problem. The algorithm predicts the learner's rating to a course using learner's profile and course information as the features. We apply Deep Neural Network based approach to predict the rating for the course. We use Multi-layer Perceptron [9] (also feed-forward neural network) model that consists of input layer, 6 hidden layers and an output layer. Multi-layer Perceptron is a supervised algorithm that learns

a non-linear function for classification or regression. It utilizes a backpropagation technique to optimize the weights so that the neural network can learn to map arbitrary inputs to outputs during training [15]. The predicted output of the network is compared to the expected output and an error is calculated. The error is then back propagated through the network, one layer at a time, and the weights are updated according to the amount contributed to the error. We use Dropout regularization technique to prevent neural networks from overfitting [8]. This is a technique where randomly selected neurons within the network are ignored while training the model. The dropout is applied after each hidden layers. The "Relu" activation function is applied to all the hidden layers. Activation functions convert an input signal of node to an output signal and introduce non-linear properties to neural network.

Many times, the system does not have much information about the new learner's preferences in order to make recommendations. This scenario is referred as "Cold Start", which is a classical problem in recommendation system. In order to build the profile of the new learner, we applied the concept of transfer learning where we identified the learners who are similar to the new learner and used their preferences. The concept of similar learner also helps in matching learners who are mutually interested, and likely to communicate with each other based on their profile characteristics and course enrollment. One of the main reasons for very high dropouts rate in MOOC is lack of engagement among the users [20]. Studies [18][21] have shown that collaboration among the learners promotes better engagement and reduces dropouts on MOOC platform . This would help in fostering the communication between the learners and forming social community of learners. We used the following similarity measures to compute the similarity between learners.

1. Similarity between the projects completed by the learners. We applied content matching approach "Latent Dirichlet Allocation" to find the similarity between their projects.
2. Similarity between their profile characteristics such as Profile Overview and skills
3. Similarity between the description of the courses that the learners have enrolled.

The steps for computing similarity between the learners is described in Algorithm 1. The algorithm computes the similarity between the learners i.e., the distance between a learner with every other learners based on the learner's history. This is computed offline and updated at certain intervals. We applied user-based Nearest Neighbor (K=5) approach to find the similar learners.

4. SYSTEM ARCHITECTURE

The framework of our approach named LeCoRe, shown in Figure 1. Our recommendation approach combines both content-based and collaborative filtering techniques. The proposed recommendation approach consists of four major phases:

Algorithm 1 Learner-Learner Similarity

Input: Learner's profile information

Output: Matrix representing the similarity score between learners

Initialize all the diagonal elements of matrix to 1 and rest 0

for $i \in \{1, ..., N\}$ **do**

  for $j \in \{i+1, ..., N\}$ **do**

    Apply LDA on description of projects completed by learners l_i and l_j

    Compute the cosine similarity between Project Description Topics vector of l_i and l_j

$$cos\_sim(PD_{l_i}, PD_{l_j}) = \frac{\vec{PD}_{l_i}.\vec{PD}_{l_j}}{||PD_{l_i}||.||PD_{l_j}||}$$

    Apply LDA on profile overview of learners l_i and l_j

    Compute the cosine similarity between Profile overview Topics vector of l_i and l_j as:

$$cos\_sim(PO_{l_i}, PO_{l_j}) = \frac{\vec{PO}_{l_i}.\vec{PO}_{l_j}}{||PO_{l_i}||.||PO_{l_j}||}$$

    Calculate the skill/concepts similarity between learners l_i and l_j as:

$$Skill\_similarity(S_{l_i}, S_{l_j}) = \frac{|S_{l_i} \cap S_{l_j}|}{|S_{l_i} \cup S_{l_j}|}$$

    where S_{l_i} and S_{l_j} is the set of skills possessed by learners l_i and l_j respectively.

    Apply LDA on description of courses enrolled by learners l_i and l_j

    Compute the cosine similarity between Course Description Topics vector of l_i and l_j as:

$$cos\_sim(CD_{l_i}, CD_{l_j}) = \frac{\vec{CD}_{l_i}.\vec{CD}_{l_j}}{||CD_{l_i}||.||CD_{l_j}||}$$

  end for

  Calculate learner-learner similarity score as:

$$L_{ij} = (cos\_sim(PD_{l_i}, PD_{l_j}) + cos\_sim(PO_{l_i}, PO_{l_j}) + Skill\_similarity(S_{l_i}, S_{l_j}) + cos\_sim(PO_{l_i}, PO_{l_j}))/4$$

end for

1. Learners Similarity: The system retrieves the learner's profile information which consists of individual characteristics of the learner such as profile overview, projects, etc. as well the learner's course history. The system utilizes the learners' profile to compute the similarity among the learner, as discussed in Algorithm 1. The output will be learner-learner similarity matrix which will have the similarity score of one learner with rest of the learners. The learner-learner similarity matrix will be stored in the database. These computations are performed offline and updated after certain intervals.

2. Data Filtering: The system retrieves the learner's profile and filter the features required for the collaborative filtering and content filtering. The layer also filters the

courses for which the learner has not provided any rating.

3. Feature Extraction: For content-based approach, we apply feature extraction techniques that unify numerical as well as text features. We apply LDA to extract the features from textual data - course title, course description, and profile description of the learner. These features are represented as vectors. The numerical features are then combined with the textual feature vectors and pass it to the content-based algorithm.

4. Learner Training: In this step we separately train collaborative filtering algorithms (such as Singular Value Decomposition) and content-based algorithms (Deep neural network model).

5. Content Prediction: Finally, we apply the trained model on test set, i.e. new courses posted on the platform. The trained model can be used to predict the rating that the learner will provide to the new courses. The system provides the top-3 recommendations to the learners sorted based on the decreasing order of the rating predicted by the trained model.

5. DATASET

We collected the dataset from Learning & Development team within Accenture through the REST-based services. The dataset consists of learner's profile information and the courses they have enrolled or completed. The dataset consists of 5,000 unique learners and 49,202 unique course content, resulting in total of 2,140,476 enrollments by all learners. The learners have enrolled for multiple courses.

6. EVALUATION AND RESULTS

In this section, we discuss the evaluation of our proposed framework. For collaborative filtering techniques, we considered tuples of $<Learner\ Id$, $Course\ Id$, $Rating>$ as features. In content-based technique, we considered tuples of $<Skills$, $Country$, $Experience$, $Industry$, $Title$, $Description$, $Rating$, $Category$, $Duration$, $Profile\ Overview$, $Content\ Type>$. The features considered for the study are explained in Table 2. These features are passed as an input to the Deep Neural Network model. We used 10-fold cross-validation set up in order to avoid overfitting. Deep neural network models are typically sensitive to the magnitude or scale of features. We apply feature scaling to all the features used as inputs in Deep neural network. The training process will run for a fixed number of iterations through the training dataset called epochs. We used 50 as epoch size. We specified batch size as 10. Batch size is the number of instances that are evaluated in the training set before the weights are updated in the neural network. We applied efficient Gradient Descent algorithm "Adam" [10], an optimizer used to search through different possible weights for the network that minimize loss. Multi-layer Perceptron model requires tuning a number of hyperparameters such as the number of hidden layers, number of neurons in each hidden layer, batch size, epochs, optimizer, activation function etc. We used GridSearch [14] techniques to find the best parameters for a prediction algorithm. It performs exhaustive search over specified parameters for any estimator object. The parameters of the estimator object are optimized by cross validated grid-search. We use "GridSearchCV" library in scikit-learn [12]. We also use trial and experimentation approach to arrive at the optimal number of neurons at each hidden layer that minimizes the overall error. The deep neural network model was implemented using Keras library [11]. In order to assess the effectiveness of the proposed hybrid recommendation model, we considered two evaluation metrics - Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) [13]. Mathematically, they are defined as:

$$MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y_i}| \tag{1}$$

$$RMSE(y, \hat{y}) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y_i}|^2} \tag{2}$$

where n is the number of samples, $\hat{y_i}$ is the predicted value of the i^{th} sample and y_i is the corresponding true value.

The performance measures of recommendation techniques are shown in Table 1. For evaluating the proposed framework, we compare the recommendations made by a baseline approach with that of the proposed framework. Learners' skillset is one of the most preferred modes of recommending the courses [6]. We considered a baseline approach that involves recommending courses based on the matching of the learners' skills with the course description. The MAE value of baseline algorithm is 1.853. The prediction made by Deep learning algorithm is better than the Collaborative as well as baseline approach. Deep learning algorithm performs better as it extracts more relevant features from the non-linear function. The recommendations made by the proposed framework performed significantly better as compared to the baseline approach.

7. SYSTEM IMPLEMENTATION

The framework is integrated within our internal web-based learning platform. The system recommends the right set of content to the learners based on their preferences captured either implicitly or explicitly. The platform is integrated with a bot which acts as a virtual buddy for the learner. The bot learns the learner's preference implicitly through their course enrollment or completion history. The bot also captures the learner's preference explicitly by asking the learner. We use DialogFlow [19] to build conversational interface (bot) which provides the natural language understanding services via intent identification. The recommendations are exposed as a REST-based services and integrated via webhook of DialogFlow. The system consists of two major components:

1. **Learning Content Recommendation:** The system recommends the relevant training content to the learners based on learner's history. The system provides the recommendations based on two considerations - learner's personal preference and the preference of other similar learners. The former is referred as "Suggested content based on your interests" and the latter as "Content trending among similar peers".

Table 1: Performance measures of Hybrid Recommendation techniques

Algorithms	MAE	RMSE
Singular Value Decomposition	0.61	1.00
Slope-One	0.68	1.05
K-Nearest Neighbor	0.613	1.03
Deep Neural Network	0.42	0.66

2. **Similar Learner Community:** The system also helps in finding the community of other similar learners and thus promotes peer learning. The learners can communicate with other similar learners and engage in meaningful discussions.

Due to space limitations we are not able to show screenshot of the system. However, screenshots can be accessed at [23]

8. CONCLUSION AND FUTURE WORK

In this work, we proposed hybrid recommendation framework to build learner's preference. The proposed approach solves the cold start problem often faced by new learners. We applied various predictive modeling techniques to evaluate our recommendation framework. We observed that the proposed framework is able to model the learner's preference quite well. As future work, we will include learner's career path preference for recommending learning content. We also plan to pilot the system to a set of users to evaluate the recommendations.

9. ACKNOWLEDGMENTS

The authors would like to thank Learning & development team for funding the research work and providing huge volumes of data in order to carry out this research work. The authors would also like to thank Tejaswini Pd for helping with the development.

10. REFERENCES

[1] Piao, Guangyuan, and John G. Breslin. "Analyzing MOOC entries of professionals on LinkedIn for user modeling and personalized MOOC recommendations." Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization. ACM, 2016.

[2] Qiu, Jiezhong, et al. "Modeling and predicting learning behavior in MOOCs." Proceedings of the Ninth ACM International Conference on Web Search and Data Mining. ACM, 2016.

[3] Apaza, Rel Guzman, et al. "Online Courses Recommendation based on LDA." SIMBig. 2014.

[4] Lemire, Daniel, and Anna Maclachlan. "Slope one predictors for online rating-based collaborative filtering." Proceedings of the 2005 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2005.

[5] Baek, Jiye, and Jesse Shore. "Promoting student engagement in MOOCs." Proceedings of the Third (2016) ACM Conference on Learning@ Scale. ACM, 2016.

[6] Aher, Sunita B., and L. M. R. J. Lobo. "Combination of machine learning algorithms for recommendation of courses in E-Learning System based on historical data." Knowledge-Based Systems 51 (2013): 1-14.

[7] Bousbahi, Fatiha, and Henda Chorfi. "MOOC-Rec: a case based recommender system for MOOCs." Procedia-Social and Behavioral Sciences 195 (2015): 1813-1822.

[8] Srivastava, Nitish, et al. "Dropout: A simple way to prevent neural networks from overfitting." The Journal of Machine Learning Research 15.1 (2014): 1929-1958.

[9] Nasrabadi, Nasser M. "Pattern recognition and machine learning." Journal of electronic imaging 16.4 (2007): 049901.

[10] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

[11] Chollet, FranÃğois. "Keras: Deep learning library for theano and tensorflow.(2015)."

[12] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." Journal of machine learning research 12.Oct (2011): 2825-2830.

[13] Willmott, Cort J., and Kenji Matsuura. "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance." Climate research 30.1 (2005): 79-82.

[14] Bergstra, James S., et al. "Algorithms for hyper-parameter optimization." Advances in neural information processing systems. 2011.

[15] Hecht-Nielsen, Robert. "Theory of the backpropagation neural network." Neural networks for perception. 1992. 65-93.

[16] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." Journal of machine Learning research 3.Jan (2003): 993-1022.

[17] Wallach, Hanna M. "Topic modeling: beyond bag-of-words." Proceedings of the 23rd international conference on Machine learning. ACM, 2006.

[18] Staubitz, Thomas, and Christoph Meinel. "Collaboration and Teamwork on a MOOC Platform: A Toolset." Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale. ACM, 2017.

[19] `https://dialogflow.com/`. [Online; accessed 04-Mar-2018].

[20] Onah, Daniel FO, Jane Sinclair, and Russell Boyatt. "Dropout rates of massive open online courses: behavioural patterns." EDULEARN14 proceedings (2014): 5825-5834.

[21] Zepke, Nick, and Linda Leach. "Improving student engagement: Ten proposals for action." Active learning in higher education 11.3 (2010): 167-177.

[22] Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." Computer 42.8 (2009).

[23] `https://github.com/kumarabhinav04/LeCoRe`.

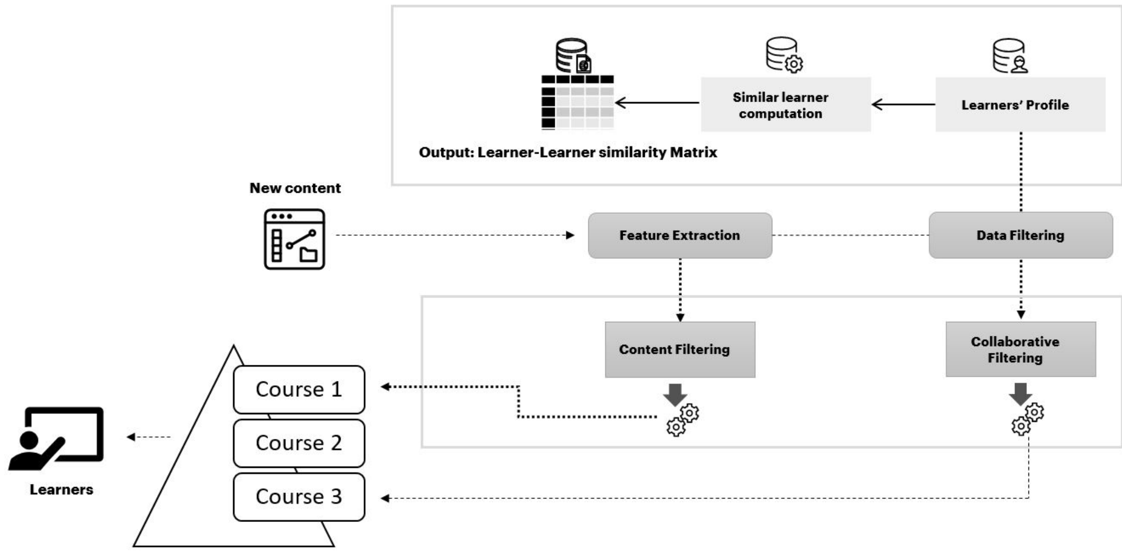

Figure 1: LeCoRe Framework

Table 2: Features considered for the study

Attributes	**Description**	**Entity**
Course Id	Unique Id of the course	Course
Title	Textual summary of the course	Course
Description	Textual description of the course	Course
Category	Category of the course e.g. Web development, Mobile development	Course
Duration	Total duration of the course	Course
Content Type	Type of the content e.g. web-based, classroom etc.	Course
Learner Id	Unique Id of the learner	Learner
Country	Country to which learners belong	Learner
Skills	Skills possessed by the learner	Learner
Education	Level of educational degree learner has. We considered five levels of education - High School, Diploma, Bachelor, Masters, and PhD	Learner
Experience	Total years of work experience learners possess	Learner
Profile Overview	Profile description of the learner	Learner
Industry	Industry group (Accenture Vertical) to which the learner belongs to e.g. Financial Services, Health & Public Service etc.	Learner
Project description	Textual description of the projects completed by the learner	Learner
Rating	Score provided by the learner to a course on a scale of 1-5	Learner-Course