Course Bundle Recommendation in Online Education Sites

Muhammad Hamza Bin Aqeel
LUMS
21100126@lums.edu.pk

Syed Abdul Hadi
LUMS
21100194@lums.edu.pk

Saad Bin Waqas
LUMS
21100241@lums.edu.pk

Iman Aleem
LUMS
21100245@lums.edu.pk

***Abstract*—Existing recommendation systems in online educational sites recommend individual courses to users, based on their course history. However no system exists that recommends bundles of courses to users following a sequential order. In this project, we propose a recommender system that generates bundles of courses to users of online learning sites based on the preferences of similar users and previously existing interactions. The system makes sure to recommend a bundle that consists of courses from different streams (e.g Chemistry, Media, Tourism etc.) and also take into consideration the difficulty level and order of these courses. Using feature engineering and collaborative filtering techniques, similarities are established between users and their preferences generated. Probability distributions are then used to rank courses and subsequently generate bundles for each user separately for a given snapshot of the weighted interactions' bipartite graph.**

1. Introduction

Existing recommendation systems recommend users of online educational sites individual courses from different streams (e.g Psychology, Human Resource etc.) but to the best of our knowledge, no model recommends an ordered bundle across different streams. Users with multiple stream preferences will, therefore, not be shown a sequential order of courses to choose from given that these courses can belong to different streams. The solution we propose recommends bundles of courses to users, taking into account the order/difficulty level of the courses.

1.1. Input

Two matrices will be considered as our inputs. One is the User-Book weighted adjacency matrix, which represents a weighted bipartite graph. Each weighted edge represents the rating of a course given by a user who has previously interaction with the course. The second input is the User-Feature matrix. This contains one hot encoded vectors representing the streams preferred by users in order.

1.2. Output

The output is an updated weighted adjacency matrix. Each edge weight represents the probability of a course being recommended to a user based on the users previous preferences and similarity to others.

2. Related Work

A Recommender System utilizes the opinions of members in a community to help new/existing participants of that community select items relevant to their needs [1]. It is a sub-class of information filtering systems, and capitalizes on calculated user preferences or ratings to determine if a user matches with an item. By extension, the Bundle Recommendation Problem aims to the find optimal bundle of items for users based on their collective preferences [2].

Neural networks consist of pools of simple processing units divided as input, output, and hidden units, which communicate by sending signals to each other over a number of weighted connections [3]. The learning paradigms where Neural Networks are used are supervised (pre-labelled inputs) and unsupervised (not including labels).

Graph Convolution Networks are used to input graphs into a neural network instead of structured data (such as images) directly to train weights and solve the required problem [4].

A bipartite graph is a graph whose vertices can be divided into two disjoint sets, such that each node in one set has an edge with at least one node in the other set. A heterogeneous graph is a graph having more than one type of nodes (such as user node and item node), and can be modeled as a bipartite graph if there are two types of nodes.

The link prediction problem is a problem outlined mainly for social networks, inferring which new interactions among members of a network are likely to occur given a snapshot of interactions [5]. Online education sites are websites offering free/paid course content from various institutions around the world, and a bundle of courses being recommended to a user based on own preferences and rating of alike users can be posed as a link prediction problem.

[6] makes use of a Contextual Hierarchy Tree to build a sequence of learning curriculum which is essential in course recommendation models, in order to ensure that users are only recommended courses which sequentially align with their qualification. Using this hierarchy tree, bundles of items are created as [7] has discussed the use

of a Bundle Generation Network (BGN) to balance quality and diversity in each bundle. A novel approach that has been adopted in our model is the recommendation (and creation) of bundles from a variety of different streams instead of recommending courses from a single stream as has previously been implemented in course recommendation algorithms. To ensure quality and relevance of each bundle generated for a user, [8] builds a Markov Chain seeing each user's history and interaction with items, the relevance of a bundle is quantified through the interaction of a user with items similar to those inside the bundle. Moreover, [9] uses a collaborative filtering approach to predict user preferences for a product by creating clusters of similar items and users to ensure quality of the bundles that are recommended to users. Additionally, [10] makes use of collaborative tagging to complement sparse information on user preferences and enhancing the quality and relevance of the individual items being recommended. This helps by making use of available personal information about users in addition to similarity with other users, and interaction with items.

[11] uses Sequential Pattern Mining (SPM) to statistically find relevant patterns in e-learning modules which are delivered in a discrete sequence that is known to be followed by users. Once the patterns are detected, they assist in the building of user-course interactions, user-bundle interactions and course-bundle affiliations which are the key elements of the Bundle Graph Convolutional Network (BCGN) used by [12] for Bundle Recommender Algorithms. [13] deploys a context-aware recommendation algorithm that uses time as contextual information to filter out the items that stand as less relevant among the items ranked by the recommender system. Such a mechanism improves the precision of the model by incorporating the contextual relevance of an item in addition to that established using the feature matrices of the users and items.

In the study conducted by [14], once a recommender algorithm is deployed, it can be set to constantly learn changes in user preferences by logging the history of user purchases and updating the feature matrix and the interactions used by the algorithm to continuously improve performance.

3. Techniques

The goal of the algorithm is to generate a probability distribution, where each value in the distribution corresponds to the likelihood that item i will constitute a bundle b, which will then be recommended to a user u.

3.1. User-User Similarity

Based on existing research work [10], we construct a user-user similarity matrix using feature engineering and collaborative filtering techniques, combined with dimensionality reduction and distance measures.

A user's interaction with all items $i \in I$ is represented as a one hot encoded vector V_1 with dimensions equalling the cardinality of the item set. Similarly, the most favorite item stream S of a user (determined by the number of items a user has previously consumed from a stream) is also represented as a one-hot encoded vector V_2. The dimensions d of user features, therefore, are given as follows:

$$d = |V_1| + |V_2|$$

For n users, the feature matrix for user similarity F_s is an $n \times d$ binary matrix, which undergoes dimensionality reduction using Single Value Decomposition, chosen over Principal Component Analysis and Matrix Factorization due to the sparsity in the matrix.

The $n \times n$ similarity matrix U_s is then calculated by using a standard euclidean distance measure on the reduced feature matrix, and normalized for better representation. U_s is used to predict the preference of a user for an item based on the behavior of similar users.

3.2. Nearest Neighbor Preference

Similar to [6], sequential alignment of items being recommended in a bundle to users is preserved. For example, since the algorithm pertains to recommendation of online courses, an individual who has taken upto a level 2 course in a particular stream will only be recommended a level 2 or above course from that stream.

The probability P_{ib} of an item being picked into a bundle then depends on the preference of users, which in turn is calculated using similarity measures with users who have interacted with that item before. P_{ib} is calculated as follows:

$$P_{ib} = \frac{U_{sk} \cdot R_{Ik}}{k}$$

where $U_{sk} = [Us_{i1}, Us_{i2}, ..., Us_{ik}]$, the vector of similarity of selected user with k nearest neighbors, and R_{Ik} is the vector of ratings assigned to a subset of I by the k nearest users.

Therefore, the normalized dot products give a probability distribution that is used for bundle generation.

3.3. Bundle Generation

For a bundle of size m, the composition of items is as follows:

- Courses belonging to the highest preference streams of the user, sequentially ordered;
- Courses belonging to streams usually chosen alongside the highest preferred one.

The sequential ordering is with respect to the overall courses in a stream; the generated bundle itself is an unordered collection of items. m streams are filtered out based on the criteria above, and the most relevant courses from these filtered streams based on the criteria above are selected and formulated into a bundle.

Based on updates on whether a user has interacted with an item (through a bundle or not), the item is flagged and is not recommended again. The recommendation of courses, therefore, is an acyclic process where the recommendation considers the difficulty of courses previously encountered by users. Therefore, the algorithm can be run on specified times to take into account fresh interactions and present the bundle with items having the greatest probability of selection, by solving the link prediction problem.

3.4. Space and Time Complexity

The execution of the algorithm requires a buffer to accommodate the following in each iteration:

1) The similarity of a user u with all n users who have taken a course;
2) The rating assigned by the n users to the course in question;
3) the $n \times m$ weighted adjacency matrix constructed, showing the probability of course recommendation;
4) the size of n bundles containing k courses, depending on the predefined size of courses.

Therefore, the total space complexity is $n(k + m + 2)$, and in its simplified form it is $O(nm)$.

The time complexity of the algorithm depends on the number courses which are being considered mainly, because the process of producing the probability distribution consists of constant ordered operations.

4. Experimental Evaluation and Comparisons

The real-world dataset we used for this project is the ***Goodreads Book Reviews*** dataset [15].

4.1. Dataset

This dataset was collected in 2017, with the review IDs and User IDs anonymized. Based on 2.36 million books, the dataset consists of 3 groups:

1) Meta-data of the books
2) User-book interactions (users' public shelves)
3) Users' detailed book reviews

For the purposes of our algorithm, we used four csv files; goodreads_interactions, book_id_map, user_id_map, and book_genres. Table 1 summarises the information about the dataset.

Items	1,561,465
Users	808,749
Interactions	225,394,930

TABLE 1: Dataset Summary

To relate this dataset with our work, we considered the book ratings as the difficulty level for a course (where 5 indicates a lower difficulty rating and 1 indicates a higher difficulty level) and the genre of the book as the stream. Furthermore, books highly rated by many users was considered as an easy course taken by many students.

4.2. Evaluation Metrics

We adopt a method to identify the goodness of a generated bundle. Based on the most favorite genres of a user, and other genres preferred by similar users, a subset of a genres B_s is extracted as a threshold to identify whether a bundle generated for a user at a particular snapshot of the data contains courses from that subset of genres or not. Furthermore, the difficulty level of each book that is shortlisted by the algorithm is noted and compared with the current standing of the user in terms of what the highest difficulty level of a book a user has read in that stream. If the recommended book is not upto that level, it adversely contributes to the goodness of the bundle. Furthermore, we evaluate the impact of increasing bundle sizes on the overall goodness of the generated bundle, to identify what the optimal size of the bundle can be. The error in the bundle generated for a user i is calculated as follows:

$$E_{Bi} = -log(\frac{booksfromB_s}{BundleSize})$$

4.3. Implementation

The model is evaluated amongst bundles of sizes 3, 5, 7, and 10 for each of the user. A total of 200 users were randomly selected and coupled in sets of 20, for better visualization of aggregated errors. There were a total of 10 resulting user sets, and the error of all bundles generated for the set for a particular bundle size was averaged to obtain a data point. Furthermore, The genre of books included in bundles and their corresponding probabilities (of a book being recommended) from the distribution were represented by being coupled in a key-value pair setting, where the key was the original ID of the book. These keys were used as identifiers of the books which were placed in the bundles recommended to the users.

4.4. Result and Analysis

The evaluation method implemented suggests that the error in the output of the model increases as the bundle size increases. From figure 1, we can appreciate the fact that smaller bundles are more relevant to a user's preference (preferred genre) whereas larger bundles give more flexibility for diversity in the generated bundles. However, a certain level of tolerance for non-preferred genres was maintained, unique to each bundle size, in order to keep the error rate from shooting up for mildly diverse bundles.

Existence of this tolerance limit suggests that larger bundles were generated using enough items that did not fall in

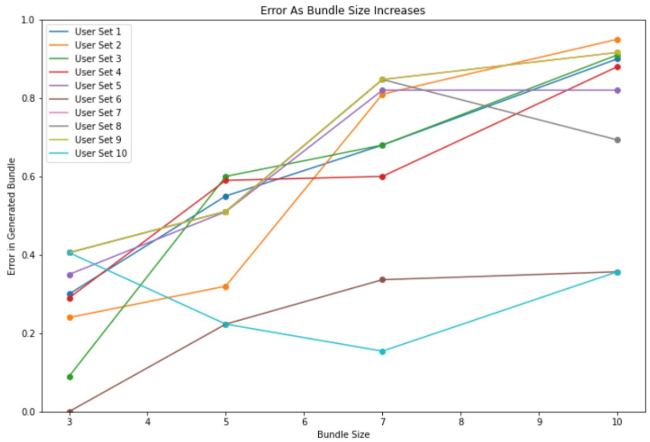

Figure 1: Performance on algorithm as bundle size increases

the preferred genre to drive up diversity such that it was compromising quality. [13] discusses an optimal balance between quality and diversity of bundles and the error rate from our model suggests that a bundle between 3 and 5 is optimal because the error rate of bundles generated for 9 out of 10 user sets significantly increases as the bundle size increases beyond 5.

5. Conclusion and Future Work

We aimed to propose an algorithm that recommends bundles of courses from different streams to a user, taking into account the difficulty level/order of the courses and the preferences of the user. Using feature engineering and collaborative filtering techniques, similarities between users were established, which were then used to generate user preferences. With the help of probability distributions, course bundles are generated, consisting of courses from preferred streams.

Future extensions of this work could work on recommending multiple bundles to a single user, instead of a single bundle only. This would provide the user with the liberty to choose from a variety of bundles, based on their choice and judgement.

In order to enhance its efficacy and reusability, the algorithm can be improved to ensure that if the interactions between users and items change,these interactions will be used to improve the efficiency of the algorithm.

References

[1] F. Ricci, L. Rokach, and B. Shapira, “Introduction to recommender systems handbook,” in *Recommender systems handbook*, pp. 1–35, Springer, 2011.

[2] T. Zhu, P. Harrington, J. Li, and L. Tang, “Bundle recommendation in ecommerce,” in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pp. 657–666, 2014.

[3] B. Kröse, B. Krose, P. van der Smagt, and P. Smagt, “An introduction to neural networks,” *anon*, 1993.

[4] L. Yao, C. Mao, and Y. Luo, “Graph convolutional networks for text classification,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 7370–7377, 2019.

[5] D. Liben-Nowell and J. Kleinberg, “The link-prediction problem for social networks,” *Journal of the American society for information science and technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[6] Y. Hou, P. Zhou, J. Xu, and D. O. Wu, “Course recommendation of mooc with big data support: A contextual online learning approach,” in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 106–111, 2018.

[7] J. Bai, C. Zhou, J. Song, X. Qu, W. An, Z. Li, and J. Gao, “Personalized bundle list recommendation,” in *The World Wide Web Conference*, pp. 60–71, 2019.

[8] A. Polyzou, A. N. Nikolakopoulos, and G. Karypis, “Scholars walk: A markov chain framework for course recommendation.,” *International Educational Data Mining Society*, 2019.

[9] anon, “A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques,” *Expert Systems with Applications*, vol. 92, pp. 507–520, 2018.

[10] Z.-K. Zhang, T. Zhou, and Y.-C. Zhang, “Personalized recommendation via integrated diffusion on user–item–tag tripartite graphs,” *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 1, pp. 179–186, 2010.

[11] J. K. Tarus, Z. Niu, and A. Yousif, “A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining,” *Future Generation Computer Systems*, vol. 72, pp. 37–48, 2017.

[12] J. Chang, C. Gao, X. He, D. Jin, and Y. Li, “Bundle recommendation with graph convolutional networks,” in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1673–1676, 2020.

[13] L. Cui, W. Huang, Q. Yan, F. R. Yu, Z. Wen, and N. Lu, “A novel context-aware recommendation algorithm with two-level svd in social networks,” *Future Generation Computer Systems*, vol. 86, pp. 1459–1470, 2018.

[14] K. Inuzuka, T. Hayashi, and T. Takagi, “Recommendation system based on prediction of user preference changes,” in *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pp. 192–199, IEEE, 2016.

[15] M. Wan and J. McAuley, “Item recommendation on monotonic behavior chains,” in *Proceedings of the 12th ACM Conference on Recommender Systems*, pp. 86–94, 2018.