LONDON JOURNAL OF SOCIAL SCIENCES

eISSN 2754-7671

A Novel Algorithm for Professor Recommendation in Higher Education

Umar Mohammad
umarmohammad0427@gmail.com

Yusuf Hamdan
yusufhamdan2023@gmail.com

Aarah Sardesai
sardesai.aarah@gmail.com

Merve Gokgol
mervekevser@gmail.com

Abstract

This paper introduces a novel professor-recommendation system designed specifically for community college and university courses. Building upon an existing algorithm for one-on-one teacher recommendations, we leveraged insights from the literature on Massive Open Online Course (MOOC) recommender algorithms. By analysing various approaches, we combined and refined ideas to develop an optimised system. Our approach utilises a tri-module framework that incorporates supervised and unsupervised learning techniques. The first module employs a Gradient-boosted Decision Tree algorithm, augmented with multiple factors and student dropout rates as ground truth, to generate a ranking score. The second module applies Apriori Association and Density-based Spatial Clustering of Applications with Noise (DBSCAN) algorithms to analyse these factors and identify professors with similar characteristics. In the third module, item-based collaborative filtering is employed, incorporating user ratings and the cosine similarity algorithm. The outputs from these three modules are subsequently integrated through a weighted average. This addition enables the system to prioritise opportunities for new professors, thereby ensuring a balanced recommendation approach. The resulting combined ranking score provides accurate recommendations for course instructors. This approach can be integrated into university course selection software for the benefit of both students and educational institutions.

Key words: Education, Machine Learning, GBDT (Gradient-boosted Decision Tree), Intelligent Recommendation System

https://doi.org/10.31039/ljss.2023.6.98

1. Introduction

Having a good professor can make a substantial difference in a student's learning experience and understanding of the subject. However, for many community college and university students, deciding on a professor to take a class from can be a difficult task. Although there are several websites and services where students can rate professors, oftentimes "the wisdom of the crowd" alone cannot cater perfectly to a student's preferences. Different students expect different qualities from their professors, and what may qualify as a good professor to one may mean the opposite to another. This conflict in student preferences leads to professor ratings being an inadequate measure to cater to students' personal preferences and learning styles. This reveals a major issue with using subjective, crowdsourced metrics for professor recommendations. However, both subjective and objective analysis have major issues that prevent them from being reliable metrics for gauging the relative quality of a professor:

- Subjective analysis results in data which is difficult to interpret due to the conflict between multiple students' preferences, learning styles, biases, etc.
- Objective analysis can give deterministic, interpretable data, but this data is meaningless if it does not effectively translate to individual student preferences.

This paper aims to supplement an objective approach with subjective elements to create a professor ranking model that can both give useful, interpretable data and also cater to individual student preferences. This is achieved by analysing the literature on teacher recommendation approaches for both one-on-one courses and Massive Open Online Courses (MOOCs). Each approach is analysed and compared to derive a single, complete recommendation system.

2. Review of existing recommendation system approaches

As a starting point for this paper, we decided to use Chen et al. (2021) due to its fully objective approach, making use of only quantifiable data and not any user ratings. Chen et al. (2021) outlines an algorithmic approach for teacher recommendation for one-on-one classes. The system makes use of a Gradient-boosted Decision Tree to make matches based on quantifiable teacher characteristics: "(1) demographic features: the demographic information of both students and teachers, such as gender, schools, etc. (2) in-class features: the class behavioural features from both students and teachers, such as lengths of talking time, the number of spoken sentences, etc. (3) historical features: the historical features aggregate each teacher's past teaching performance, which includes total numbers of courses and historical dropout rates, etc." (Chen et al., 2021:3). The network uses a pseudo-score based on student dropout rates as a "ground truth" to train the model.

In order to supplement this objective approach with a more subjective, rating-based one, we examined Verbert et al. (2011), which examined multiple collaborative filtering approaches for teacher recommendation for MOOCs. Not only does the study demonstrate the feasibility of collaborative filtering for teacher recommendation, but it also shows that for general conditions, the best method of collaborative filtering is to make use of item-based collaborative filtering, utilising Tanimoto as a similarity coefficient. Specifically, this combination performs with the highest accuracy on datasets with less user interaction, i.e., ratings.

Dai et al. (2016), on the other hand, makes use of user-based collaborative filtering instead of item-based. The paper proposes a system that makes use of LinkedIn profile data in order to find users with similar characteristics. The model generates tags based on a user's LinkedIn data in order to match users with courses that similar users enjoyed. This broad approach could be used as a general framework for a user-based collaborative filtering algorithm for professors.

Aher & Lobo (2013) makes use of an objective but personalised approach, making use of unsupervised techniques such as Apriori Association with a support percentage of 85% and K-means Clustering in order to recommend the user courses based on a multitude of factors. Although this approach is meant for recommending MOOCs specifically and not professors, it can be adapted to work with the same data points provided by Chen et al. (2021). Apriori Association can be used for non-numerical, limited data, while K-means clustering can be used for numerical data. This clusters teachers into possible groups, allowing the software to recommend teachers in the same cluster as teachers the student enjoyed previously.

Bousbahi & Chorfi (2015) takes an entirely different approach, making use of a case-based reasoning (CBR) approach to recommend MOOCs based on five main factors: course title, fees, availability, language, and location. The algorithm makes use of past circumstances to recommend courses. The paper mainly focuses on the workings of a recommender application which finds courses for the user based on whether they want to pay fees, their location, etc.

3. Evaluation of existing recommendation system approaches

Of the previous approaches, each one has certain aspects that make it more or less desirable for the use case of professor recommendation. In order to reduce algorithm complexity and computation time, inefficient or irrelevant approaches must be eliminated in order to create a streamlined system.

Chen et al.'s (2021) objective approach makes use of data points that are very relevant to the task of professor recommendation. Although Chen et al. (2021) is centred around recommending one-on-one courses, every single one of its data points is applicable to university courses as well. As a result, this approach should be included in the system.

Verbert et al. (2011)'s item-based collaborative filtering, on the other hand, is a broad approach that can be applied to both MOOCs and university courses. Since students often have certain traits they prefer in professors, item-based collaborative filtering allows personalisation in professor ratings. Since rating professors is already an existing concept, this approach can be well-integrated into the recommendation system. However, although Tanimoto similarity may perform the best for sparse datasets, it is not optimal for non-binary data such as user ratings, which may range from 1-5. As a result, to implement item-based collaborative filtering, the cosine correlation algorithm is much more optimal.

Dai et al. (2016)'s approach classifies users based on their LinkedIn profiles and suggests courses enjoyed by similar users. However, what must be considered is how much one's LinkedIn profile may suggest about their professorial preferences. Since Dai et al. (2016) originally dealt with MOOC recommendation, it is easy to understand how a LinkedIn profile may impact the subjects in which they are interested. For example, a user's education and bio may impact which subjects they are interested in. However, a LinkedIn profile alone cannot give meaningful data on a user's professorial preferences. As a result, this approach is irrelevant to our use case.

Aher & Lobo (2013)'s approach is unique yet broad, utilising an objective approach that is also personalised. The data points used for this approach can be almost anything, including the predefined data points of Chen et al. (2021). The fact that this makes use of objective data points means that it can provide interpretable data for the recommendation system. Its ability to personalise recommendations based on these factors also addresses the shortcomings of purely objective approaches like Chen et al. (2021). As a result, this approach can be integrated into the recommendation system. However, instead of using k-means clustering, it would be more optimal to use the Density-based Spatial Clustering of Applications with Noise (DBSCAN) clustering algorithm due to its better tolerance for outliers and lack of need to specify the number of clusters.

Finally, Bousbahi & Chorfi (2015)'s Case-based reasoning (CBR) approach is a broad one that can be applicable to multiple use cases. However, Bousbahi & Chorfi (2015) only makes use of five factors in its model, far less than the factors described in Chen et al. (2021). Due to this, it is doubtful whether CBR can reliably handle the amount of data required in this recommendation system. As a result, CBR is not a feasible addition to the recommendation system.

Of these 5 possible approaches, only Chen et al. (2021), Verbert et al. (2011), and Aher & Lobo (2013) will be used in designing the recommendation system due to their scalability, interpretable data outputs, and adequate personalisation.

4. Proposed system

In order to create the final system, we propose a tri-module system that makes use of both supervised and unsupervised learning techniques and both objective and subjective data. In order for the system to function properly, it must have access to detailed user rating information detailing how each user rated each of their professors (subjective data points). It must also have access to each of the data points detailed in Chen et al. (2021:3) for each professor: demographic features, in-class features, and historical features (objective data points). All numerical data points are normalised from 0 to 1 on a linear scale for range-limited data and an exponential scale otherwise.

4.1 - Module 1: Gradient-boosted decision tree

The first module makes use of a Gradient-boosted Decision Tree module very similar to the one used in Chen et al. The module is trained using all of the objective data points from Chen et al. (2021:3), using a pseudo-score based on student dropout rates as a "ground truth" in order to guide the training of the model. This pseudo-score is calculated by using equations sourced from Chen et al. (2021:2). The pseudo-score is calculated differently based on whether it is positive or negative:

Positive score: *For student s_i who has finished the class, let $\mathbf{T}_i$ be the set of all professors who have taught student s_i where t_j* represents *the jth professor in the set and p_i is the number of professors in the set. Let $M_i(t_j)$ be the number of class sessions taught by professor t_j. The positive pseudo score P(.,.) is defined as:*

$$P(s_i, t_j) = \frac{M_i(t_j)}{\sum_{n=1}^{p_i} M_i(t_n)}$$

Negative score: *For student s_k who has dropped the class, with similar notations as the Positive Score, the negative pseudo matching score N(.,.) is defined as:*

$$N(s_k, t_j) = -\exp(1 - M_k(t_j))$$

According to these definitions, the pseudo matching score of each professor ranges from -1 to 1. The score reaches a maximum of 1 when the student completes the entire class and never requests a change of professor. The minimum of -1 is reached when the student drops a class immediately after the first session. To calculate an individual professor's pseudo-score, the pseudo score for each student that the professor has taught is averaged.

4.2 - Module 2: DBSCAN clustering and apriori association

The second module makes use of Density-based Spatial Clustering of Applications with noise (DBSCAN) clustering and apriori association in order to group together professors with similar qualities and recommend professors who have similar characteristics to professors that the student has enjoyed in the past. Making use of the objective data points mentioned earlier, DBSCAN clustering is applied to numeric data types (lecture length, etc.), while apriori association is applied to non-numeric data types (gender, etc.).

Firstly, the module must compile a list of all professors, which is referred to as set $\mathbf{T}_i$, to whom the student has given a rating of 3 or more points out of a maximum of 5 points. Each professor is weighted simply by repeating their occurrence in the list based on their rating; a professor rated 3 points appears only once, while a professor rated 5 points appears three times.

4.2.1 - Apriori association sub-module

Making use of set $\mathbf{T}_i$, a classic apriori association or comparable implementation is used to mine association rules based on non-numerical data points of this dataset. Each association rule gives a score of either 1 if the professor meets the rule or -1 if the professor does not meet the rule. The professor's final score is calculated as follows:

Score calculation: *For professor t_i, let $\mathbf{R}_i$ be the set of professor t_i's score on each association rule, where r_j represents the jth score in the set and p_i is the number of scores in the set. Let $\mathbf{S}_i$ be a set of length p_i containing each association rule's support percentage from 0 to 1, where s_j represents the jth support in the set. The professor's overall score F(.) is defined as:*

$$F(t_i) = \frac{\sum_{j=1}^{p_i} r_j s_j}{\sum_{j=1}^{p_i} s_j}$$

This definition returns -1 if the professor does not fit into any association rules and 1 if the professor meets all association rules.

4.2.2 - DBSCAN clustering sub-module

The Density-based Spatial Clustering of Applications with Noise (DBSCAN) clustering algorithm is used to group similar professors into clusters. A professor's score is calculated differently based on whether it is positive or negative:

Base definition: *For professor t_i, let $\mathbf{C}_i$ be the set of all clusters which contain any member of set $\mathbf{T}_i$, where c_j represents the jth cluster in the set $\mathbf{C}_i$ and p_i is the number of clusters in the set $\mathbf{C}_i$. Let $D_i(t_i, c_j)$ be the Euclidean distance of professor t_i from the centroid of cluster c_j. Let $N_i(c_j)$ be the number of professors from set $\mathbf{T}_i$ contained within cluster c_j.*

Positive score: *Utilising the base definitions, for professor t_i who is contained within any of the clusters in $\mathbf{C}_i$, the positive score P(.) is defined as:*

$$P(t_i) = \left| \frac{\sum_{j=1}^{p_i} D_i(t_i, c_j) N_i(c_j)}{\sum_{j=1}^{p_i} N_i(c_j)} - 1 \right|$$

Negative score: *Utilising the base definitions, for professor t_i who is not contained within any of the clusters in $\mathbf{C}_i$, the negative score N(.) is defined as:*

$$P(t_i) = -\frac{\sum_{j=1}^{p_i} D_i(t_i, c_j) N_i(c_j)}{\sum_{j=1}^{p_i} N_i(c_j)}$$

This sub-module returns a minimum score of -1 if professor t_i is not within any of the clusters in $\mathbf{C}_i$ and has a Euclidean distance of 1 from the centroid of every single cluster in $\mathbf{C}_i$. The sub-module returns a maximum score of 1 if either there is only one cluster in $\mathbf{C}_i$ or all the clusters overlap completely, and t_i is the shared centroid of the overlapping cluster.

4.2.3 - Score combination

In order to create a final score for the module, the two scores from the two submodules are averaged, returning a score from -1 to 1.

4.3 - Module 3: Item-based collaborative filtering

The third module makes use of item-based collaborative filtering using the cosine similarity algorithm. The module calculates a professor's score as follows:

Score Calculation: *For student s_i and for professor t_i whom student s_i has not yet rated, let $\mathbf{S}_i$ be the set of all students who have participated in the rating system, and let $\mathbf{R}_i$ be the set of each student in set $\mathbf{S}_i$'s rating of professor t_i from a scale of 1 to 5, or 0 if the particular student has not rated professor t_i, and r_j represents the jth rating in the set. Let $\mathbf{T}_i$ be the set of all professors which student s_i has rated, where t_j represents the jth professor in the set and p_i is the length of the set. Let $\mathbf{N}_i$ be a set of length p_i containing student s_i's rating of each teacher in set $\mathbf{T}_i$ from a scale of 1-5, where n_j represents the jth rating in the set. Let $K_i(t_j)$ be the set of ratings given to professor t_j by every single student on a scale of 1-5 or 0 if the* *particular student has not rated professor t_j. Let $S_i(k_j)$ be the cosine similarity coefficient between set of ratings k_j and set* **R**$_i$*. The collaborative filtering score F(.) is defined as:*

$$F(t_i) = \frac{1}{2}\left|\frac{\sum_{j=1}^{p_i} S_i(K_i(t_j))n_j}{\sum_{j=1}^{p_i} S_i(K_i(t_j))} - 3\right|$$

This module returns a minimum of -1 when professor t_i is likely to receive a rating of 1/5 and a maximum of 1 when professor t_i is likely to receive a rating of 5/5.

4.4 - Module Combination

Once all three modules have returned a score, the scores are combined to create a singular, final score. Since modules two and three rely on user ratings, if the user has not rated any professors, the modules are not activated, and module 1 gives the final professor score. Otherwise, the average score of all three modules is the final score, ranging from a minimum of -1 to a maximum of 1.

Conclusion

In conclusion, utilising a tri-module approach to professor recommendation allows for the combination of both objective and subjective data, which allows for a more complete picture of each professor. This approach also tailors recommendations to each student's preferences, increasing the likelihood that students will find professors they enjoy working with. This algorithm can serve as a useful resource for both students and educational institutions, assisting students in finding the best professors and allowing educational institutions to ensure a higher quality of education. Some possible applications for this algorithm are the integration of the recommendation system and a rating system into a university’s course selection software, allowing students to choose professors that fit them better and increasing the number of ratings for each professor, allowing for better personalisation.

Bibliography

Aher, S. B., & Lobo, L. M. R. J. (2013). Combination of machine learning algorithms for recommendation of courses in E-Learning System based on historical data. Knowledge-Based Systems, 51, 1–14. doi:10.1016/j.knosys.2013.04.015

Bousbahi, F., & Chorfi, H. (2015). MOOC-Rec: A Case Based Recommender System for MOOCs. Procedia - Social and Behavioral Sciences, 195, 1813–1822. doi:10.1016/j.sbspro.2015.06.395

Chen, J., Li, H., Ding, W., & Liu, Z. (2021). An educational system for personalized teacher recommendation in K-12 online classrooms. Lecture Notes in Computer Science, 104–108. doi:10.1007/978-3-030-78270-2_18

Dai, K., Vilas, A. F., & Díaz Redondo, R. P. (2016). A New MOOCs' Recommendation Framework based on LinkedIn Data. Lecture Notes in Educational Technology, 19–22. doi:10.1007/978-981-10-2419-1_3

Fu, R., Tian, M., & Tang, Q. (2022). The Design of Personalized Education Resource Recommendation System under Big Data. Computational intelligence and neuroscience, 2022, 1359730. doi:10.1155/2022/1359730

Verbert, K., Drachsler, H., Manouselis, N., Wolpers, M., Vuorikari, R., & Duval, E. (2011). Dataset-driven research for improving recommender systems for learning. Proceedings of the 1st International Conference on Learning Analytics and Knowledge - LAK '11. doi:10.1145/2090116.2090122