International Journal of Engineering & Technology, 7 (4.32) (2018) 48-52

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

The Recommending Courses based on the Similarity of Students' Preferences

MoulayHachemAlaouiHarouni[1*], El-Kaber Hachem[1], Cherif Ziti[1], MustaphaBassiri[2], Joan Lu [3]

[1] Research Team EDP & Scientific Computing, Mathematics & Computer Department, Faculty of Sciences, Moulay Ismail University, Meknes, Morocco
[2] Laboratory of Physical Chemistry of Materials, Ben M'sik Faculty of Sciences, Hassan II University of Casablanca, Morocco
[3]Information and System Engineering, University of Huddersfield, Huddersfield, United Kingdom
** Corresponding author e-mail: harouni.alaoui@gmail.com*

Abstract

To keep students in touch, they need to be caught through proposing different courses similar to the ones they are interested in. The present paper is going to discuss the idea of recommending courses and how it will be applied into student profiles as the result of clustering the books or courses as well as the student preferences. It will illustrate the way the item and student profiles enable to recommend courses in our system.

***Keywords**: Recommending courses, clustering, student preferences, profiles*

1. Introduction

Recently, personalized techniques have been widely studied to automatically recommend or find customized information [1]. Personalized recommendation systems recommend an item to which a user prefers by using automatic information filtering method. Moreover, as mobile computing progresses, various resources can be available to model user preference
The users of YouTube are always in front of different videos recommended to be, firstly, similar to the previous videos the user has already seen, and to be, second, one of interesting video to the users. This idea which has been applied into our website helps to proposed different courses and shows to feed the desire of each student, and to benefit or enrich the background of the learner. Recommendation systems enable the students to find something interesting during the time of reading a particular articles or books. A single algorithm is usually responsible for a single row of recommendations, which means there may be different algorithms used for different categories. Different features are used to come up with items for the different genres like comedy or scientific books. Building recommendation system is quite challenging due to the large number of new students. However, collaborative filtering is taken into consideration to design the similarity between courses and books through data mining [2].

2. Collaborative Filtering Systems

An approach to design a recommendation system [3] is known as collaborative filtering [4], in which we are going to consider any student similar to the other learners to whom we have already recommended a particular course. This idea comes from the fact that students who are similar in terms of chosen courses would also like books that they have not seen but that a similar student has liked [5]. Let us consider the example of a student S and a set of other students who have read Animal farm by George Orwell.

It is basically proposed to that particular learner due to the similarity of other learners' preferences for the considered story. Each preference is recorded in the historical events of a profile of each student. Then, data mining helps to find similarity of two or more profiles.
That is to say, the number of courses that two learners have both opened and read might be small compared to the overall books they have seen. This makes it hard to do a recommendation, as there is not much information that we can base their decision on when considering the rating matrix[6]. Here clustering is used to improve the procedure. For example, our large collection of documents has different courses belonging to different books or lectures.
This way of clustering leads to a rating matrix that has less courses, as single article or course is replaced by famous books. Furthermore, the rating matrix is less sparse. Similarly, we can cluster students [7] using the preferred clustering algorithm and combine students to super learners and take the average among these students as the entry for the super student and books/video of lectures. Having done this, the process can be used in an iterated way to come up with an even smaller rating matrix by further clustering such that the rating matrix has filled in almost all blanks. Of course, in the end we would like to determine the entries for our original rating matrix. We can do this by using the entry of the clustered matrix [8] that corresponds to the considered student book/ course pair.
Explanation of content-based recommendation systems [9] is explored and made due to the similarity of student ratings for two items.

3. Making Recommendations

Traditional recommendation systems use collaborative filtering to predict the rating of a product to a particular user. The general idea behind collaborative filtering is that similar users vote similarly on similar items [10].
A rating matrix represents information about students and/or items. Rows represent students and their choices for items. Columns represent items and how student have chosen them [11]A rating matrix could look like the following table:

Table 1: Courses chosen by students

	A	B	C	D
S1	1	3	2	4
S2	2	2	3	3
S3	2	3	2	5

In table 1, we can see 3 students, S1, S2 and S3, who have chooses 4 items (A, B, C and D).
A: course 1
B: course 2
C: course 3
D: course 4
Each course contains several chapters, documents, exercises, links, etc. which oblige the student to visit it several times to complete it.
The following information refers to content covered Jaccard similarity and Cosine similarity, which both help to determine the similarity through the ratings of like and dislike. For example, if we record only likes of the students in our matrix, then we can work with a binary matrix:

Table 2: Courses liked by students

	A	B	C	D
S1	1	0	1	0
S2	1	1	1	1
S3	1	0	0	0

Here, we can see 3 students that have liked particular items (marked as 1). There is a 0 if no likes have been recorded. Students are similar if their rows in the rating matrix are similar according to a similarity measure, such as the Jaccard similarity [12](for binary matrices) or the Cosine similarity (for matrices with general numerical entries).
Recall that the Jaccard similarity for two sets X and Y is given by

$$\mathrm{JSim}(X, Y) = (X \cap Y)/(X \cup Y) \quad (1)$$

If we consider two students S1 and S2, then all items liked by a student S would belong to their set as follows:

Table 3: Courses liked by students

	A	B	C	D
S1	1	0	1	0
S2	1	1	1	1

Here, we have:

$$\mathrm{JSim}(S1, S2) = 2/4 = 1/2$$

Furthermore, recall that the Cosine similarity [13] of two vectors: $X = (X_1, \dots, X_n)$ and $Y = (Y_1, \dots, Y_n)$
The Cosine similarity is given by

$$\mathrm{CosSim}(X, Y) = \frac{\sum_{i=1}^{n} X_i . Y_i}{\sqrt{\sum_{i=1}^{n} X_i^{2}} \sqrt{\sum_{i=1}^{n} Y_i^{2}}} \quad (2)$$

The range of values that Cosine similarity can generate are within [−1, 1], where larger values refer to a larger similarity.

Table 4: Courses chosen by students

	A	B	C	D
S1	1	3	2	4
S2	2	2	3	3

For: $\mathrm{CosSim}(S1, S2) = 26/(30 \cdot 26) \sim 0.9309$

When there are blanks in the data, we may remove all entry combinations (X_k, Y_k) where either X_k or Y_k or both are blank (we denote a blank by "?").

Table 5: Courses chosen by students with blank

	A	B	C	D
S1	1	?	2	4
S2	2	2	?	3

We get $\mathrm{CosSim}(S3, S4) = \frac{14}{17 \cdot 13} \sim 0.9417$
So, we would regard S3 and S4 more similar than S1 and S2.
A recommendation for a student S is made based on students that are most similar to S and recommending items that these students liked. We can obtain an entry for a student/item (S, I) pair in two different ways. The first one is to look at students similar to the considered student S and determine the choice for (S, I) by taking the choice of the similar students into account. Another way, which is perhaps not that intuitive, is to look at items similar to item I. In this instance, we would only take items into account that have already been rated by students S. we can take the average rating of student S for these most similar items to obtain the score for (S, I).
One of the most prevalent problems in recommender systems is derived from the fact that for new students there are not any ratings for the items that could be offered. This problem is known as "Cold Start" [14] (Andrew et al. 2002). This is why recommendation systems ask new students for input on some items when using the system for the first time. Also, there is a requirement for a sufficient number of students who have rated an item before that item can be recommended to a learner.

4. Content-based Filtering Systems

A content-based filtering system selects items based on the correlation between item content and user preferences, as opposed to a collaborative filtering system that selects elements based on the correlation between people with similar preferences [15]. ASTEMOI [16]has a content-based filtering system. It makes recommendations by comparing a user profile with the content of each document in the collection. The student's profile is constructed by analysing the content of the documents that the student has found interesting. Documents that the learner finds interesting can be determined using explicit or implicit feedback. Explicit feedback requires the user to evaluate the documents reviewed on a scale. In an implicit return, the interests of the user are deduced by observing the actions of the user, which is more convenient for the user but more difficult to implement.If a student has read books, then the evaluation on them can serve as a good prediction for other books they have not read yet. In order to capture this a bit more formally, we would create for each item, in our case for every course/ book, an item profile that characterises the items. For books, the profile would usually include the author, genre, and information. We may add any two different types of features that are helpful to characterise our items. The first features are such as scientific magazine and the author's autobiography. The second feature is also to select the year when the book is published. Assume that we now have characterised each book by a profile. If a student has liked course/ book A, we would like to recommend books that are similar in terms of its item profile and the students' preferences.If our item profile only contains the first features above, we could use Jaccard similarity to measure the similarity of books. If there are also numerical variables, such the second feature involved, we can resort to cosine distance between the items.
The approach recommends books [17] based on a similarity measure and we would recommend courses that are close to a book the student likes or reads.

5. Features of items and Students

When characterizing items by features, we have to think about what could be well-suited features for our item class. For example, features of books could be the author, the publisher, the year when it is published, or the literary genre.This data can often be obtained from the description of a given book. By identifying well-suited features, we can create profiles for our item class. We could think of profiles as models for our items, using features as our parameters to help the recommendation systems work well.Some well-suited features could be to create profiles for books, courses and articles. Through these, we can apply what we have acquired from quantifying similarities and differences to these items so as to create profiles for students of different selected items.Firstly, Item profiles allow characterization of items such as book, its author, etc. in terms of their properties (features). An item profile consists of feature-value pairs.
Secondly, Student profiles characterize preferences that students have regarding particular courses. A student profile summarizes and records the preferences of each student.Let us consider a course profile as an example for an item profile. We can characterise a course by a set of authors (each course contains many documents and books for several authors) and its average recommendation score. We have k authors for all the documents and books in our database, where the average recommendation score that a course has obtained is also recorded.Then our course profile has k+1 entries (k Boolean and 1 numerical value for the score).An example: 1 0 1 0 1 1 4, where a 1 indicates that an author is part of the courses and the last entry gives the average recommendation. Now we can measure the similarity of the items (courses) by using the Cosine distance.For student profiles, we need to create vectors with the same components describing user preferences. We can characterize a student by the courses that they have liked. To summarize their preferences, we record for each author A the fraction of courses that the student S liked where A was an author. So, if we have k authors overall, the profile consists of k entries where each entry is a value in the range [0, 1]. An example for six authors is:0.2 0.1 0.3 0.05 0.1 0.01, Where the first entry indicates that author 1 has been the author in 20% of the courses that student S liked.Again, we can use Cosine distance to measure the similarity between students.

6. The Similarity of Students' Preferences on Choosing Courses

Data mining has great potential to improve the understanding of a student's behaviour. In our case, we are mostly breaking the course choices made by the students, which influences the presentation of the courses in our platform and which can bring significant appreciations to the learner.
Now, we would like to know which sets of courses our student usually read together. Because if we know it in advance, we can adapt the presentation page of our platform for these lesson packages on a regular basis, which will bring us more profits at the level of the students' need. Frequent courses should have a large fraction in common. We want to find frequent lesson packages. The following table illustrates the choices of eight students:

Table 6: Frequentcoursesselected by different student

Student	Courses
1	Course 1, Course 2, Course 3, Course 4
2	Course 4, Course 1, [redacted], Course 3, Course 5, Course 6
3	Course 1, Course 6, Course 2, Course 5, Course 7
4	Course 7, Course 1, Course 8, Course 9
5	[redacted], Course 8, Course 3, Course 10, Course 11
6	[redacted], Course 8, Course 1, Course 9
7	Course 3, Course 11, Course 1, Course 5, Course 2

To find all the similarities or the doubletons of courses based on the choice of students comes as a way to make recommendation system effectively. We start our search the similarities of double courses, and then we compare between the students who select the two.The following table illustrates the similarities of selected courses more than the text can express:

Table 7: The similarities of selected courses among different students

	Course 8	Course 5	Course 3	Course 2
Course 1	4,6	2,3,7	1,2,7	1,2,3,6,7
[redacted]	5,6	2,3,7	1,2,5,7	
[redacted]	5	2,7		
Course 5	—			

Here, we can see which courses students select -highlighted in blue- to show more highly chosen courses by the most students. These insights are important for presenting different courses based on their similarity of preferences in our platform.
Frequent itemsets: We approach one of the main families of data characterization techniques:
The discovery of frequent itemsets: This problem is often considered as the discovery of "association rules;" although the latter is a more complex characterization of the data, the discovery of which depends fundamentally on the discovery of frequent itemsets.
One of the popular applications for frequent itemsets would be a diversified medical application. For example, for the detection of drugs that causes particular side effects or protein structure analysis or DNA sequence analysis. In addition, an important medical application is cellular compartment and protein function

subtrees within the Gene Ontology [18]. A popular online store like amazon.com offers millions of different items for sale and has tens of millions of customers.
The motivation will be to know which items are bought together or vice versa, so the motivation will be to find all the similar customers.
The association rules: To identify the most important relationships between item sets we use the association rules as
a collection of ...rules[19]. Frequent itemsets tell us which courses are frequently studied together.
Let us take a look at our popular itemsets, course 1, course 2 and course 3. We know that these courses are studied together. If some students choose courses 1 and 2, they will probably choose course 3. This can be illustrated as a rule of association. The general form of an association rule is $X \Longrightarrow Y$ where X is a set of elements and Y is an element.[20]
The implication of the rule is that if all elements of X are chosen by a student, then Y is likely to be chosen by that student.
Let I be the set of all items and T= {t1, t2,…,tN} be the set of all transactions. Each transaction ti contains a subset of items chosen from I.
An important property of an item set is its support count which is the number of transactions that contain a particular item set; it can be defined as follows:

$$\sigma(X) = |\{t_i | X \subseteq t_i, t_i \in T\}| \quad (3)$$

In association rule X and Y are disjoint itemsets. It can be measured in terms of its support and confidence.
The support of an association rule determines how often a rule is applicable to a given data set. While confidence determines how frequently items in Y appear in transactions that contain X. So:
Support:

$$s(X \Longrightarrow Y) = \sigma(X \cup Y)/T \quad (4)$$

Confidence:

$$c(X \Longrightarrow Y) = \sigma(X \cup Y)/\sigma(X) \quad (5)$$

In our example above we consider the association rule {course 1, course 2} ⟹ { course 3}
The support count $\sigma(\text{course 1}, \text{course 2}, \text{course 3}) = 3$
The support: $s = \sigma(\text{course 1}, \text{course 2}, \text{course 3})/T = \frac{3}{7} = 0.42$
The confidence:$c = \sigma(\text{course 1}, \text{course 2}, \text{course 3})/\sigma(\text{course 1}, \text{course 2}) = \frac{3}{5} = 0.6$
Confidence of an association rule is an important measure because it gives us an insight into the reliability of the conclusion of the association rule. For a given rule, the higher the confidence, the more likely it is for course 3-here- to be present in transactions that contain course 1 and course 2.
A-priori algorithm. The APriori algorithm is a data mining algorithm developed in 1994 by Rakesh in the area of association rule learning[21].It is used to recognize properties that appear frequently in a dataset and to infer a categorization.
A key concept in the A-priori algorithm is the following observation. First, if a set of elements is common, all its subsets must also be frequent. Second, the support of an item set never exceeds the support of its subset.
It is also very interesting that our typical threshold, s,will be 1% of the total of the chosen items.
Here is the procedure to follow:

- *Counting up frequent items and their supports.*
- *Generating all 2-pairs of frequent items.*
- *Counting up the 2-pairs of frequent items and their supports.*
- *Generating all triples of frequent items.*

The A-priori algorithm takes advantage of the fact that any subset of a frequent item set is also a part of frequent item set.

7. Conclusion

Our paper contributes through a new collaboration recommendation system that is used Association Rules Algorithm to recommend courses to a targeted student based on what other similar students have chosen. The association rule is a desirable tool for making course recommendations, but the confidence of association rule has a great impact on performance. By choosing a relatively high confidence, we can achieve a better performance. We would like to have these systems worked effectively. That is why a lot of works can be done in the future; for example, doing the evaluation of effectiveness of our recommendation systems can take different forms.
In the ideal case, we would like to always recommend a book/ a course a student would like to read. When working with ratings of likes (form 1 to 5 likes), we could even be more extreme and ask for a recommendation system that always proposes books that a student would rate with five likes. All these procedures aim at evaluating the effectiveness of a recommendation system.

References

[1] Sarwar, B. 2001. Item-based collaborative filtering recommendation algorithms. In: WWW 10 Conference, pp. 285–295.
[2] Liu, D. and Y. Shih. 2005. Integrating AHP and Data Mining for Product Recommendation Based on Customer Lifetime Value, Information and Management 42(3), 387-400.
[3] Horozov, T., N., Narasimhan and V. Vasudevan. 2006. Using location for personalized POI recommendations in mobile environments. In: SAINT. pp. 124–129.
[4] Breese, J., D., Heckerman and C. Kadie. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In: Proc. of the 14th UAI.
[5] Yang, Li, B., Q. Yang and X. Xue. 2009. Transfer learning for collaborative filtering via a rating-matrix generative model. In: ICML, pp. 617–624.
[6] Marlin, B. Modeling user rating profiles for collaborative filtering. in NIPS'03
[7] Wang, F. and H. Shao. 2004. Effective Personalized Recommendation Based on Time-Framed Navigation Clustering and Association Mining. Expert Systems with Application 27(3), 365-377.
[8] Banerjee, A., D. Inderjit, G. Joydeep, M. Srujanaand S. Dharmendra. 2007. A generalized maximum entropy approach to Bergman co-clustering and matrix approximation. in KDD '04, pp. 509–514.
[9] Indulska, J., L.T. Yang, T. Ungererand J. Cao (Eds.). 2007. UIC 2007, LNCS 4611, pp. 1130–1139.Springer-Verlag Berlin Heidelberg.
[10] Ekstrand, M. D., J.T. Riedl and J. A. Konstan. 2011. Collaborative Filtering Recommender Systems. Foundations and Trends in Human–Computer Interaction 4(2), pp 81-173.
[11] Tung, H. and V. Soo. 2004. A personalized restaurant recommender agent for mobile e-service. In: IEEE International Conference on e-Technology. E-Commerce and e-Service. pp. 259–262.
[12] Fu, J. and J. Ye. 2017. Simplified neutrosophic exponential similarity measures for the initial evaluation/diagnosis of benign prostatic hyperplasia symptoms 9, 154.
[13] Korenius, T. J., Laurikkala and M. Juhola. 2007. On principal component analysis, cosine and Euclidean measures in information retrieval. Doi:10.1016/j.ins.2007.05.027
[14] Andrew, I. S., P. Alexandrin, H. U. Lyle and M.P. David. 2002. Methods and Metrics for Cold-Start Recommendations. Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York City, New York: ACM. pp. 253–260. ISBN 1-58113-561-0. Retrieved 2018-02-02.
[15] Lieberman, H. 1995. Letizia: An Agent that Assists Web Browsing. In proceedings of the 1995 International Joint Conference on Artificial Intelligence. Montreal, Canada.
[16] El Emary, I. and A. Brzozowska. 2017. Shaping the Future of ICT. Boca Raton: CRC Press. Chapter 6 pages Artificial Intelligence in E-Learning.

[17] Alaoui, H.H., E. Hachem and C. Ziti. 2016. Data mining for the service of intelligent tutoring system. Int.J.Mult.disc.scie 1(1): 61 - 65.
[18] Shahab. S.S., S. Jamshed and A. Rashid. 2017. A Novel Approach for Book Recommendation using Fuzzy based Aggregation, Indian Journal of Science and Technology 10(19), DOI: 10.17485/ijst/2017/v10i19/96787.
[19] Koyutürk, M., Y. Kim, S. Subramaniam. S. Wojciech and G. Ananth. 2006. Detecting conserved interaction patterns in biological networks. J ComputBiol 13:1299–322.
[20] Manda, P., S. Ozkan, H. Wang, F. McCarthy, and S.M. Bridges. 2012. Cross-ontology multi-level association rule mining in the gene ontology. PLoS One 7: e47411.
[21] Agrawal, R., T. Imielinski and A. Swami. 1993. Mining Association Rules Between Sets of Items in Large Databases. SIGMOD Conference. pp. 207-216.