"St. Kliment Ohridski" University - Bitola

# HORIZONS

# INTERNATIONAL SCIENTIFIC JOURNAL

## SERIES B
## Natural Sciences and Mathematics, Engineering and Technology, Biotechnology, Medicine and Health Sciences

Year X Volume 1 August 2014

For the publisher: Prof. Zlatko Zhoglev, PhD, Rector



**Editor:** Elena Kitanovska-Ristoska, MA

ISSN 1857- 8578
Print: AD Kiro Dandaro-Bitola, printing copies: 200

37.018.43:004

# USING RECOMMENDATION SYSTEMS FOR LEARNING MATERIALS[11]

**Josif Petrovski**
University St. Kliment Ohridski, FAMIS, PhD candidate
josif.petrovski@yahoo.com

**Abstract**

Recommendation systems are becoming more attractive today in electronic commerce, where algorithms are developed to determine with high precision the desires of customers. However, despite the rapid development of these systems and their use in the Internet environment, there is a small number of cases where these systems are adapted and used in the learning process. System recommendations complements the natural process of relying on friends, classmates, teachers and others who participated in the selection of learning materials. In this paper we review the aspects of creating a system for recommending documents, and one crucial question: how exactly to find learning materials that suit the needs of the student.

*Keywords: Learning, System recommendations, Utility matrix, Recommendation model;*

## INTRODUCTION

E-learning systems are becoming more popular in educational institutions. The rapid development of e-learning has changed the traditional approach to learning and created new challenges for educators and students. Educators have difficulty choosing the appropriate learning materials for the growing number of materials online. Students have a problem when they decide which material is adequate for them and their needs. Therefore, the educator need an automated way of getting feedback from students in order to better

[11] original scientific paper

guide their learning process. Students, on the other hand, would benefit when an electronic system could suggest activities, and in intelligent way to select and recommend materials and documents for learning, which would improve the knowledge of students.

First attempts to develop recommendation system are made and applied in the field of e-commerce. The basic principle of the system is to use feasibility of creating a list of recommended items and to verify that the user likes the recommended items. This validation may arise directly by customers, or caused by the use of data that represent the previous activities of users. Such systems are called Recommendation System (Ricci, Rokach & Shapira, 2011). Recommendation systems are using many different techniques that will be discussed later, and depending on the techniques they use there are two types of Recommendation System:

- *Collaborative filtering systems*, which focus on the relationships between users and items. The similarity of two items is determined by the familiarity of ratings that were set by users and evaluated both items.

- *Content-based systems*, where first come the characteristics of items. The similarity of objects is determined by comparing their features.

- *Hybrid systems*, combination of both systems above.

## COLLABORATIVE FILTERING SYSTEMS

The main goal of collaborative filtering is to predict or provide recommendation for products based on the activities of users who are like-minded. The assistance comes in the form of a top list where items are listed according to their importance for the buyer. Activities, opinions and ratings of the products in the system can be obtained solely by the users. Each user has a list of items where he has given his opinion. It is in form of rating, usually expressed on a numerical scale with values from 1 to 5, with 5 being the highest. For some items there may be no value. In this case the user is called *active user* and the system tends to set rating for that item in two ways:

- *Forecast*, a numerical value showing the forecast whether the active user would like the subject.

- *Recommendation*, Top N list with items that would be favor of the active user. This list contains items that have not been reviewed and evaluated by the user and is called Top recommendation.

The system consists of a crossed data for users and items represented in a matrix, called Utility matrix. Each data is in the form of rating, which represents the user's opinion on that subject. Some values can be 0 or blank, indicating the fact that the user is yet to rate that item.

Collaborative filtering systems do not always succeed in connecting items to users. In case of introduction of new users or new items a problem may appear called Cold Start, because there will be no data the system can process and bring the correct decision. Another problem that arises in these systems is Data Sparsity. In practice, many commercial systems have large databases. As a result, the Utility matrix will be large and mostly empty, thus creating obstacle for proper recommendation. This is often related to the previous problem, because the system tends to make recommendations based on past actions of the user, so for new users the system can’t make a wish-list.

## CONTENT-BASED SYSTEMS

First thing these systems do is to create a profile for each item that is offered. Profile is a record or set of records that represent main features of that item. Usually profile consists of features that are easily visible. For example, features of a textbook are *authors, publishers, year of publication* and *classes* where is used.

These systems are of great importance to our paper, because they can discover features of documents or learning materials that should be recommended.

There are items where features are not immediately visible. This is often the case with written documents, intended for students to read. Recommendation system can offer titles that we would have interest in, but how will find out which are the right ones. Unfortunately, when we have documents it is not easy to reveal their main features. As a substitute for the main features we identify words

that describe the topic of the document. For example, if an article is written about IT technology it will have words like computer, Internet, multimedia, data etc. Once a document is classified under the topic IT technology we will easily note that these terms, called *keywords*, appear frequently, but until the classification is done these words are not counted as keywords yet.

Classification begins with reviewing the documents and finding keywords. At first glance it appears that the most words in the text are the keywords. But this assumption is wrong. Most of the repeating words are conjunctions and prepositions (and, or, if, that) or other words that help to build idea, but have no relevance to the topic. These words are called Stop words and are rejected during document classification. In fact, the keywords are relatively rare words. On the other hand, not all rare words may be keywords. There are certain words that rarely appear in the text, and again belong to the Stop group. The difference between the rare words that have significance and those who don't is located in the concentration of useful words in only a few documents.

The formal measure for determining the concentration of a given keyword in a relatively small number of documents is called TF.IDF, short from Term Frequency times Inverse Document Frequency. Basically, TF.IDF determines the relative frequency of a particular word in the document compared with inverse proportion of that word in a set of documents. This calculation determines how relevant a given word is in the specified document. Words that are common in a document or a small group of papers tend to have a higher TF.IDF value versus general words, such as conjunctions and prepositions. It is calculated as follows. We assume that we have a set *D* of documents. Choose a document *d* from the group and word *Z* to compare:

$$Z_d = f_{Z,d} * log\left(\frac{D}{f_{Z,D}}\right)$$

where $f_{Z,d}$ represents the number of repetitions of the word *Z* in document *d* and $f_{Z,D}$ is the number of documents from the set where the word *Z* is repeating (Berger, 2000). Depending on the values of the variables we usually have two situations.

If we assume that |D|~$f_{z,D}$, i.e. the number of documents is approximately equal to the repetition of the word Z over all documents in D. Then $0 < log\left(\frac{|D|}{f_{z,D}}\right) < 1$, $Z_d$ will have lesser value then $f_{z,d}$ but still positive value. This indicates that the requested word, though often to be found in the documents, has significance. This would be the case when we search the set of documents associated with the topic *education* for the words "teacher" or "student." But these words are too common, unless the user specifies that the document should contain exactly those words. These common words lead to a low index of TF-IDF and the system might omit these words in the search.

On the other hand, we assume $f_{z,d}$ is high and $f_{z,D}$ is low. Then the value of $log\left(\frac{|D|}{f_{z,D}}\right)$ will be very high and so will the value of $Z_d$. This is our case because word with high $Z_d$ points to the fact that this word is keyword in the stated document and not just a common word. This word has high discrimination power (Ramos, 2003). When searching with this word, the user would be satisfied with the results offered.

Finally, when we leave out STOP words and perform classification with TF.IDF method, each document will get a set of keywords, by which will become recognizable and now Recommendation system can easily notice its main features and make an accurate recommendation.

Here we see that TF.IDF a simple but effective algorithm for determining keywords of written documents. But despite this TF.IDF has its limits. His algorithm is not developed enough to resolve the grammar of the language. First example are synonyms, where words with same meaning are considered as different. Also, when words change their structure in plural are classified as different. This might be a problem if we have large database of documents. However TF.IDF remains as the basis for development other algorithms that surpass its disadvantages.

## RECOMMENDATION SYSTEMS IN EDUCATION

Recommendation systems allow users to share their opinions and therefore to use the gained experience. They can be defined as "systems that provide individual recommendations as a result or lead the participant through a series of interesting and useful items separated from a large group of possible options" (Burke, 2002). These systems are made up primarily to support web users as support in decision making in certain situations, in terms of preparing the information that would be useful in those situations where the user does not have enough experience or knowledge of the environment (Adomavicius & Tuzhilin, 2005).

In the education process systems often recommend documents with specific content by keywords or according to the curriculum. Earlier we saw how a system can extract those keywords if they are not manually entered by the author. After efficient extraction of keywords, next step is to start the Recommendation system and offer Top-N list of documents that are in the interest of the student. This method compares the profile of the student with certain characteristics and predict ratings for papers that student hasn't assessed. We will be presented hybrid system, because it can recommend documents that have not yet received a rating (disadvantage of collaborative filtering), while allowing the recommendation of documents with different content (enrichment of content- based systems).

When a student selects a document to read, a group of related documents will be proposed. The user has the opportunity to evaluate the proffered documents by relevance or interest. On one hand the system for collaborative filtering examines the similarities between the students and their interests. On the other hand, the system for content-based filtering process similarities between the documents and the results are placed in the matrix. The system predicts ratings for all documents that are not rated by users and offers opportunity to be evaluated by the users. Results of the forecast are compared to actual ratings of the user that determines the accuracy of the assessment.

This system is a combination of several similar systems being used and still being updated. According to this model there are four basic matrices involved: User-User, Document–Document, User-Document and Rating-Difference. User-User matrix contains

similarities between users, Document-Document matrix contains similarities between documents, User-Document contains actual or anticipated user ratings and last matrix has the differences between actual and predicted values of the ratings.

Document-Document Matrix contains all the values of the similarity between documents and is filled after all documents are entered into the system. When a user give rating of a particular document, it will be saved in User-Document Matrix and at the same time the User-User Matrix will be updated. At that time predicted rating values will fill the User-Document Matrix. Once users give their rating to a document, it will replace the predicted rating. Then, the difference between actual and predicted ratings are stored in the Rating-Difference Matrix and the predicted values are analyzed again for the remaining documents.

To determine similarities between users technique of collaborative filtering applies, which calculates the similarities of users according to their assessment of documents. All values entered by users are considered as a vector of dimension N, where N is the number of documents in the system. First the algorithm calculates the distance between two users using Taxi geometry (Candillier and others, 2007) and then normalizes distance values between 0 and 1, thus limiting similarity to values from 0 to 1. This way the User-User Matrix is filled with values shown as an example in Table 1:

| | User1 | User2 | User3 | User4 | User5 |
|---|---|---|---|---|---|
| User1 | | 0,25 | 0,62 | 0,77 | 0,56 |
| User2 | | | 0,09 | 0,88 | 0,36 |
| User3 | | | | 0,21 | 0,12 |

***Table 1.*** *Similarity matrix for users*

The number of documents that are rated by many users have major impact on the accuracy of determining the similarity between users. The following equation calculates the similarity:

$$sim(user(i), user(j)) = \frac{(N * R_{\max}) - \sum_{x=1}^{N} | r_i(x) - r_j(x) |}{N * R_{\max}} \qquad [1]$$

where N is number of documents normally assessed, $R_{max}$ is the highest value (usually 5), $r_i(x)$ and $r_j(x)$ are values given for document x from users i and j.

For calculating the similarity between documents is used a method that not only compares keywords as main features of the documents, it takes their authors name and titles (Sarwar and others, 2001). It calculates the importance of keywords with TF/IDF and the similarity is determined by the equation:

$$sim(document(i), document(j)) = \sum_{x=1}^{N} w_x * \frac{a_x(i,j)}{b_x(i,j)} \qquad [2]$$

where N in number of attributes (three in this case – author, title, keywords), $w_x$ is significance of x, $a_x$ is number of common type of x attributes for documents i and j, $b_x$ is the lowest number of x attributes for documents i and j. By determining the values for each pair documents the Document-Document Matrix is filled with values. By adding a new document in the system, this method will compare it with all other documents and will re-enter the values in the Document-Document Matrix.

As previously said Recommendation system stores user ratings for documents in the User-Documents Matrix. This matrix has two types of ratings: real and predicted. The real rating is set by the user depending on how much he liked the document. The predicted rating is set by the system in places where the user has not yet assessed documentation, and it will automatically change if the user enter a rating at any time. The system tends not to leave empty fields in the matrix, because according to these ratings the system makes recommendation. To predict ratings for $User_i$ for $Document_j$ we follow these steps:

1. Generate Top-N list of users similar to $User_i$ who have set ratings for $Document_j$
2. Generate Top-N list of documents similar to $Document_j$ that have received a rating from $User_i$
3. Determine prediction by the similarity of users (Equation 3)
4. Determine prediction by the similarity of the content of the documents (Equation 4)
5. We determine final rating prediction with combining values obtained in steps 3 and 4 (equation 5)

When predicting user ratings, the value of similarity between users will be used as constant. The system uses collaborative filtering where Top-N similar neighbors to $User_i$ will be selected from User-User Matrix. The calculation of the predicted rating PR_user is calculated as follows:

$$PR\_user = \frac{\sum_{n=1}^{N} S_{i,n} * R_{j,n}}{\sum_{n=1}^{N} S_{i,n}} \quad [3]$$

where $S_{i,n}$ is a similarity between $User_i$ and $Neigbor_n$, $R_{j,n}$ is the rating of $Neigbor_n$ for $Document_j$.

Next step is to select top-N similar documents from Document-Document Matrix that are rated by users. The calculation is done as follows:

$$PR\_document = \frac{\sum_{n=1}^{N} S_{i,n} * R_{j,n}}{\sum_{n=1}^{N} S_{i,n}} \quad [4]$$

where $S_{i,n}$ is similarity between $Document_i$ and $Neigbor_n$ of the Document, $R_{j,n}$ is rating of $User_j$ for the $Neigbor_n$. By this equation the system gives a prediction based on content-based filtering. To make correct prediction according to the content, user must have rated at least one similar document. The system will update the forecast every time the user gives new rating for the document.

The last step is combination of results through ponder average. The number of neighbors is used as measurable factors. The equation for calculating prediction P is

$$P = \frac{PR_{user} * N_1 + PR_{document} * N_2}{N_1 + N_2} \quad [5]$$

where $N_1$ is number of selected neigbors of the user and $N_2$ number of selected neigbors of the document. This equation calculates the predicted rating for $User_i$ and $Document_j$ with combination of predicted values of ratings based on document contents and the similarity in the activities of users. All the ratings are stored in the

Rating – Difference matrix, as shown on table 2, so the system can use the values all over again.

| Real Rating | Predicted Rating | Difference | User | Document |
|---|---|---|---|---|
| R1 | P1 | R1-P1 | U1,4 | D1,5 |
| R2 | P2 | R2-P2 | U2,4 | D2,5 |

***Table 2.*** *Model of Rating – Difference matrix*

## DOCUMENT RECOMMENDATION

As mentioned before, the purpose of the system is, as the student reads a document, to recommend a series of similar documents. Recommendation should meet two conditions. First, the user must see the recommended document for the first time. Second, the recommended documents have to be related to the content of the students document. To enforce these conditions, the document is pulled from User- Document Matrix, according to the predicted values for rating, i.e. documents that have not received rating from the user, but are high on the Top-N list according to predicted rating. So, **the recommendation of documents is done only by predicted rating.**

## CONCLUSION

In this article a model was presented of a hybrid system for recommending documents to read. The aim was to describe Recommendation system that can be used in the educational process, a system that recommends learning materials. Searching for appropriate learning materials can be hard and long, and often without success. The combination of collaborative and content- based filtering helps students to improve their learning process. This system can not only be an improvement of the institution itself, but can also be the basis to create a Learning Management System which nowadays are becoming increasingly popular in educational environments.

These systems are further explored and are improved in order to increase their intelligence. And to achieve this they need to accurately identify students preferences, to follow their steps and fully

adapt to their needs. Therefore, this systems can use additional tools offered by the education itself. As educators use series of questions and tests to determine the level of knowledge of students, same way the system can obtain additional information that will help build a profile of each student. Next step would be connecting these systems with other similar systems on the Internet, making a global system with learning materials from many areas that would be easily available.

## REFERENCES


Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on, 17*(6), 734-749.

Berger, A et al (2000). Bridging the Lexical Chasm: Statistical Approaches to Answer Finding. In *Proc. Int. Conf. Research and Development in Information Retrieval*, 192-199.

Bobadilla, J., Ortega, F., Hernando, A., Alcalá, J. (2011) Improving collaborative filtering recommender system results and performance using genetic algorithms, *Knowledge Systems, 24*: 1310–1316.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction, 12*(4), 331-370.

Candillier, L., Meyer, F., & Boullé, M. (2007). Comparing state-of-the-art collaborative filtering systems. In *Machine Learning and Data Mining in Pattern Recognition* (pp. 548-562). Springer Berlin Heidelberg.

Emadzadeh, E., Nikfarjam, A., Gauth, K.I., Why, N.G. (2010) Learning materials recommendation using a hybrid recommender system with automated keyword extraction. *World Applied Science, 9(11)*: 1260–1271.

Rajaraman, A., & Ullman, J. D. (2012). *Mining of massive datasets*. Cambridge University Press.

Ramos, J. (2003). Using tf-idf to determine word relevance in document queries. In *Proceedings of the First Instructional Conference on Machine Learning*.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001, April). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web* (pp. 285-295). ACM.

Tang, T.Y., McCalla, G. (2003) Smart recommendation for an evolving e-learning system. In *Proceedings of the Workshop on Technologies for Electronic Documents for Supporting Learning, International Conference on Artificial Intelligence in Education, AIED 2003*.