Wong, L.-H. et al. (Eds.) (2013). Proceedings of the 21st International Conference on Computers in Education. Indonesia: Asia-Pacific Society for Computers in Education

A Hybrid Recommender System based on Material Concepts with Difficulty Levels

Guibing GUO[a*], Mojisola Helen ERDT[b] & Bu Sung LEE[a]
[a]*School of Computer Engineering, NTU, Singapore*
[b]*Multimedia Communications Lab (KOM), TU, Germany*
*gguo1@e.ntu.edu.sg

Abstract: Recommending learning materials for e-learning systems often encounters two issues: how to classify and organize learning materials and how to make effective recommendations. In this paper, we propose a new algorithm to handle these two problems. Specifically, we compile each learning material to concepts according to their relevance which is modeled as the length of a term-weight vector. Then recommendations are generated by taking into account the document's similarity with some good learning material, the personalized time-aware usefulness of the learning material, the concepts of the learning material as well as their difficulty levels. Experimental results based on a small sample demonstrate the effectiveness of our method in terms of knowledge gain obtained.

Keywords: e-learning systems, learning materials, material concepts, knowledge gain.

1. Introduction

The popularity of web-based learning environments have led to the creation of huge amounts of digital learning materials that are either used as mandatory or supporting materials during lectures or shared amongst learners. One of the challenges facing e-learning today is to provide effective and personalized recommendations to learners in order to overcome the *information overload* problem (Guo, Zhang and Thalmann, 2012; Guo, Zhang, Thalmann & Yorke-Smith, 2013). Another issue recognized is how to classify and organize learning materials effectively.

Many approaches have been proposed to enhance recommender systems in e-learning (Zhang, Tjhi, Lee, Vassileva & Cooi, 2010; Doan, Zhang, Tjhi and Lee, 2011). The research has shown that hybrid approaches (Ghauth and Abdullah, 2010) could generate more accurate recommendations than non-hybrid approaches, especially to alleviate some inherent issues of recommender systems such as *data sparsity* (Guo, Zhang and Thalmann, 2012; Guo, Zhang, Thalmann & Yorke-Smith, 2013). In particular, a hybrid method can recommend learning materials that well suit both the student preferences and the current learning context. Our work is inspired by Ghauth and Abdullah (2010) where learners are identified as similar to each other, then preferentially recommended what has been most useful to similar "good" learners. Good learners refer to the students who have already worked with these learning materials and have passed some tests effectively. Our work is also motivated by the Peer-based Intelligent Tutoring Model proposed by Champaign, Zhang and Cohen (2011) in which each learning object stores those students who experienced the object, together with their initial and final states of knowledge. Then, these interactions are used to reason about the most effective lessons to show future students based on their similarity to previous students. However, most previous works have not considered the concepts of learning materials, the difficulty levels and the time spent on learning materials simultaneously. The material concepts refer to the topics of a specific discipline or domain.

In this paper, we propose a hybrid approach to recommend effective learning materials in two steps: (1) compiling the learning materials to material concepts in terms of the relevance; (2) determining the most useful learning materials to recommend, according to the ratings given by students, the time that they spent on learning materials and the difficulty levels that they specify to different concepts. Therefore, we take into account the difficulty levels of each concept, the "content" (whose suitability will be determined by the first step of the algorithm) and the

"collaboration" (provided by the second step) to generate the most beneficial personalized learning materials. We have built a prototype of the system and performed simulated experiments on a sample dataset. The results demonstrate the effectiveness of our method in terms of knowledge gain.

2. Our Approach

The general structure of our approach is illustrated in Figure 1. On the one hand, tutors can upload the links of learning materials, the course names as well as the concept names to the e-learning systems, and determine the right courses for all learning materials which will be stored in the database, and after which an important learning material-concept relevance matrix will be set up. On the other hand, students are required to specify a difficulty level for each concept, to give their ratings to learning materials and to estimate the time that they spent in reading through each material. Finally, our algorithm will generate personalized recommendations by taking into account both material similarity and personalized usefulness of the promising "good" learning materials.

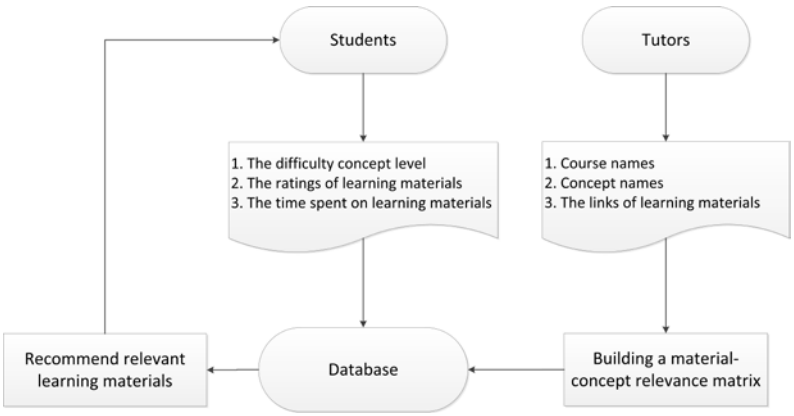

Figure 11. The general structure of our approach

2.1 Building a Learning Material-Concept Relevance Matrix

Assume that there are a set of *n* learning materials denoted as $M = \{m_1,\ldots,m_n\}$, and a set of *k* concepts denoted as $C = \{c_1,\ldots,c_k\}$. We form a learning material-concept relevance matrix L, where each entry $l_{i,j}$ equals 1 if a learning material m_i belongs to concept c_j; otherwise equals 0. For clarity, we keep the symbols u, i, j for indexes of students, learning materials and concepts, respectively. Each learning material m_i for concept c_j is represented as a term vector in *d*-dimension: $m_{i,j} = (t_{j,1},\ldots,t_{j,d})$, where each term $t_{j,p}(1 \le p \le d)$ is defined as a single word of the concept c_j and hence *d* is the number of words in that concept. For example, for the concept "software process model", it contains three words, namely "software", "process", and "model", i.e., $d = 3$. The weight of each term $w_{j,p}$ is computed using the well-known *tf-idf* method, reflecting the extent to which the term $t_{j,p}$ is important to the concept c_j. Thus, the term weight vector for each learning material with a specific concept can be represented by: $\vec{w}_{i,j} = (w_{j,1},\ldots,w_{j,d})$. We define the relevance of a learning material m_i to a specific concept c_j as the length of the term weight vector, i.e., $l_{i,j} = \|\vec{w}_{i,j}\|$, and $\vec{L} = (l_{i,1},\ldots,l_{i,k})$ as the relevance

vector across all concepts, and $\max L = \max_{l_{i,j}} \vec{L}$ as the maximum relevance in the vector. The learning material m_i will be classified to concept c_j if the following criterion is satisfied:

$$l_{i,j} > \theta_l \max L, \tag{1}$$

where $\theta_l \in [0,1]$ is a relevance threshold, and we empirically set its default value 0.8. Thus, a learning material may belong to multiple concepts if the above criterion for each concept is satisfied. In case none of the concepts meet the relevance criterion, we will relate the learning material to the concept of the highest relevance. Finally, the learning material-concept relevance matrix is built by setting the entry as 1 if a learning material belongs to the corresponding concept, or as 0 if not.

2.2 Determining the Usefulness of Learning Materials

After compiling each learning material to relevant concepts, a relevance matrix is constructed. On the other hand, students are asked to specify a difficulty level for each concept by issuing a rating from 1 to 5, where 1 means that the concept is the easiest and 5 the most difficult. Formally, suppose there are q students, and each of them $s_u (1 \le u \le q)$ rates a difficulty rating $d_{u,j}$ for each concept c_j. In addition, for each learning material m_i, students report their preference rating and the time that they spent in reading it, denoted as a couple $(p_{u,i}, t_{u,i})$. The rating $p_{u,i}$ indicates the usefulness of the learning material m_i relative to student s_u. It takes an integer value from the range [1, 5] where 1 means the least useful and 5 the most. The time $t_{u,i}$ is another indicator of the usefulness of a specific learning material. It is estimated by students in minutes such as 10 or 20 minutes. If a student has not used a learning material, the two ratings will become (0, 0).

Hence, in this work we compute user (i.e., student) similarity according to the material difficulty and preference ratings (time ratings will be used later). In particular, we denote $sim_d(u,v)$ and $sim_p(u,v)$ as the similarities computed based on difficulty ratings and preference ratings, respectively. Then user similarity is computed as the average of both types of similarities:

$$sim(u,v) = \frac{1}{2}(sim_d(u,v) + sim_p(u,v)), \tag{2}$$

where $sim(u,v) \in [0,1]$ is the overall similarity between users s_u and s_v. The difficulty similarity is defined as the differences between difficulty levels towards the common concepts rated by them:

$$sim_d(u,v) = 1 - \frac{1}{4k}\sum_j \left| d_{u,j} - d_{v,j} \right|, \tag{3}$$

where 4 is the maximum rating difference since the rating scale is in the range [1, 5].

The preference similarity is defined as the cosine value of angles between (the overlapping of) two rating vectors $\vec{r}_u$ and $\vec{r}_v$, where $\vec{r}_u = (r_{u,1}, \ldots, r_{u,n})$ is a preference rating vector for user s_u over all learning materials. Cosine similarity is a commonly used similarity measure:

$$sim_p(u,v) = \frac{\sum_{i \in M_{u,v}} r_{u,i} \cdot r_{v,i}}{\sqrt{\sum_{i \in M_{u,v}} r_{u,i}^2} \sqrt{\sum_{i \in M_{u,v}} r_{v,i}^2}}, \tag{4}$$

where $M_{u,v}$ denotes the set of learning materials that both users s_u and s_v have rated. However, as pointed out by Guo, Zhang and Yorke-Smith (2013), the cosine similarity suffers from the 'single-value' problem. That is, when the rating vector has only one element, the resultant cosine value will always be 1 regardless of the real rating values. To handle this problem, we similarly compute the preference similarity as the normalized differences between rating values:

$$sim_p(u,v) = 1 - \frac{1}{4}\left| r_{u,i} - r_{v,i} \right|, \tag{5}$$

where i indicates the only learning material that both users s_u and s_v have rated.

In our approach, two users are regarded as similar if their similarity is greater than a predefined threshold θ_s (by default, $\theta_s = 0.8$). Then a set $U_u = \{v \mid sim(u,v) > \theta_s, v \in U\}$ of similar users can be identified and hence recommendations can be made according to the ratings of similar users. However, for the users without sufficient rating information, known as the *cold users* in the recommender systems (Guo, Zhang and Thalmann, 2012), similar users are hard to be determined by similarity. To cope with this issue, we treat all other users as similar users of the cold users.

The time information is used to discount the preference (usefulness) ratings given by users, reflecting the efficiency and value of each learning material. We take into account this factor with the aim to recommend the most useful learning materials from which users can get the most benefits in as a short time as possible. Hence, the time-discounted preference rating is defined as:

$$r'_{u,i} = \frac{r_{u,i}}{t_{u,i}}. \tag{6}$$

Hence, the time-aware usefulness $p_{u,j}$ of a learning material m_j is computed as the average of time-discounted ratings given by the similar users:

$$p_{u,j} = \frac{1}{|U_u|} \sum_{v \in U_u} r'_{v,j}. \tag{7}$$

Note that for the learning materials rated by the active user herself, the usefulness computation will take into consideration her rating data as well.

2.3 Generating Recommendations

In this section, we proceed to determine the beneficial value of each learning material. Specifically, it is composed of both the similarity between the learning material in question and the 'good' learning material, and the computed usefulness. A good learning material (denoted as m') is defined as the material that receives the greatest usefulness within a specific concept. The similarity between a learning material m_i and the good material m' is computed in two cases. First, when the concept has only one term, the similarity is defined as the difference between the term weights:

$$sim(m_i, m') = \frac{1}{\max W} \left| w^i_{j,c} - w^{'}_{j,c} \right|, \tag{8}$$

where $\max W$ is the maximum difference between any two weights $w^i_{j,c}$ and $w^{'}_{j,c}$ toward a certain concept c. Second, when the concept has multiple terms, the cosine similarity is used:

$$sim(m_i, m') = \frac{\sum_c w^i_{j,c} \cdot w^{'}_{j,c}}{\sqrt{\sum_c {w^i_{j,c}}^2} \sqrt{\sum_c {w^g_{j,c}}^2}}. \tag{9}$$

Hence, the recommendation (beneficial) value is the combination of material similarity with the good learning material and the personalized usefulness of the learning material:

$$rec(u, m_i) = \frac{1}{5}\left(1 - \frac{1}{\sqrt{s+1}}\right) p_{u,i} + \frac{1}{\sqrt{s+1}} sim(m_i, m'), \tag{10}$$

where $rec(u, m_i)$ is the recommendation value for user s_u on target learning material m_i, and $s = |U_u|$ is the number of similar users. Therefore, a list of learning materials can be ranked and recommended according to the beneficial values in descending order.

3. Evaluation

To verify the effectiveness of our approach, we implemented a prototype based on a small sample of data. Specifically, we collected a number of learning materials (scoping in two concepts) from online knowledge systems (such as Wikipedia, Google search) and some tutorial web pages as shown in Figure 2. The two concepts are "software process model" and "API" and hence four terms are obtained. Of the ten collected materials, four are quite related with the first concept, three are somehow but not quite correlated with the first concept, two are highly associated with the second concept and the left one material is irrelevant to both concepts.

materialID	materialName	content	wordcount	link
1	software process model 1	A software process model is a simplified description ...	310	http://www.wiki.answers.com/Q/What_are_the_softw...
2	software process model 2	A software development process, also known as a s...	127	http://www.en.wikipedia.org/wiki/Software_developm...
3	software process model 3	Software process models.Software process is a set ...	570	http://www.css.freetonik.com/wiki/software_engineeri...
4	software process model 4	The software process.A structured set of activities r...	79	http://www.google.com.sg/url?sa=t&rct=j&q=software...
5	OSProcessNotSoRelated 5	Two-state process management model. The operati...	813	http://en.wikipedia.org/wiki/Process_management_(c...
6	OSProcessNotSoRelated 6	In computing, a process is an instance of a compute...	243	http://en.wikipedia.org/wiki/Process_(computing)
7	OSProcessNotSoRelated 7	Definition of Process.The notion of process is centra...	373	http://www.personal.kent.edu/~rmuhamma/OpSystem...
8	UI Design 8	User interface design or user interface engineering i...	183	http://www.en.wikipedia.org/wiki/User_interface_desi...
9	API 9	This library is called the API the Application Program...	645	http://www.dummies.com/how-to/content/getting-star...
10	API 10	What is Java API? Java API is not but a set of clas...	213	http://www.roseindia.net/java/javaapi/java-api.shtml

Figure 12. The original data collected and used in our experiments

Compiling Learning Materials to Concepts. For each learning material, we count the number and times of terms occurring in the documents, and compute the *tf-idf* weight for each concept term. Then a vector of term weights is obtained from which the relevance is computed as the length of the weight vector. Finally, we determine whether a learning material belongs to a specific concept based on Equation (1). The results show that the first four learning materials are correctly classified to the first concept. However, for concept "API", only the last material, namely "API 10" is correctly classified (but "API 9" is not). This may be due to the fact that the document "API 10" is much shorter than "API 9" (see the column "wordcount" in Figure 2), and hence the former document possesses a higher term frequency than the latter. For other learning materials which do not meet the requirement of Equation (1), the concept with the greatest relevance will be adopted. As a consequence, materials 5-7 are labeled by the first concept whereas "API 9" by the second concept. To sum up, all materials are correctly classified to proper concepts.

Recommending the Most Useful Materials. Different users often gain different benefits after reading even the same learning materials. Thus, we define the *knowledge gain* for each user as the benefits obtained via using a specific learning material according to her learning ability:

$$Gain(s_u, m_i) = LA_u \cdot rec(s_u, m_i), \tag{11}$$

where $Gain(s_u, m_i)$ denotes the knowledge gain obtained by user s_u using learning material m_i, LA_u denotes her learning ability, and $rec(s_u, m_i)$ is the personalized beneficial value given by our algorithm. For experiments, we randomly generate a learning ability in [0,1] for each user, where 1 means the active user can completely absorb all the benefits provided by a learning material whereas 0 indicates completely not. For simplicity, we keep the learning ability fixed, though it may vary in different contexts. For each experiment, we randomly simulate and group *n* users (each with a random learning ability) together, where *n* varies in the set {10, 20, 50, 100, 200, 500}. Then we calculate the average and the maximum learning ability of each group. To have more reliable values, we generate 50/100 groups each time and use the average and the maximum values across all groups. Thus, the users with the maximum learning ability are the best users in the groups. The objective of the experiments is to show whether the users with average learning ability, if they adopt our recommendations (denoted as AvgRec), can achieve the same as or even better knowledge gain than the best users using random materials (BestRand). The mean absolute errors (MAE) between the knowledge gain obtained in two cases is used to measure the quality of our recommendations:

$$MAE = \frac{1}{\kappa} \sum_u \sum_i \left| Gain(s_u, m_i) - Gain(s_{u'}, m_r) \right|, \tag{12}$$

where $s_{u'}$ represents the best user and m_r is a randomly selected learning material, and κ is a normalization factor. Thus, smaller MAE indicates better accuracy relative to the best users. The

knowledge gain and MAE on 50 groups and 100 groups are shown in Figures 3 and 4, respectively.

Figure 3 shows that consistently, as the number of users (i.e., students) increases, the knowledge gain obtained by the average users (who receive and adopt our recommendations) is equivalent or even better than the best users who pick random learning materials, regardless of the number of groups (used to determine the maximum and the average learning abilities). The variation of knowledge gains is due to the differences of initialized average and maximized learning ability. The results from Figure 4 show that the MAE remains low (smaller than 0.08) across different numbers of students, indicating that the differences of knowledge gain obtained by average users and best users are quite small. In conclusion, our method can provide users with personalized and useful learning materials from which they can gain good knowledge.

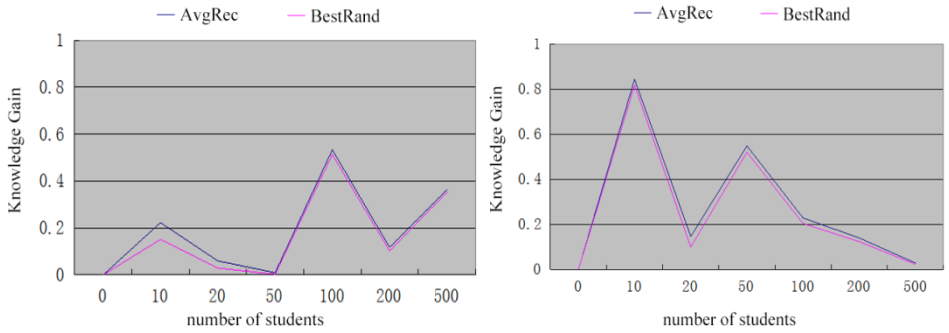

Figure 3. The knowledge gain obtained with 50 (left) and 100 (right) groups

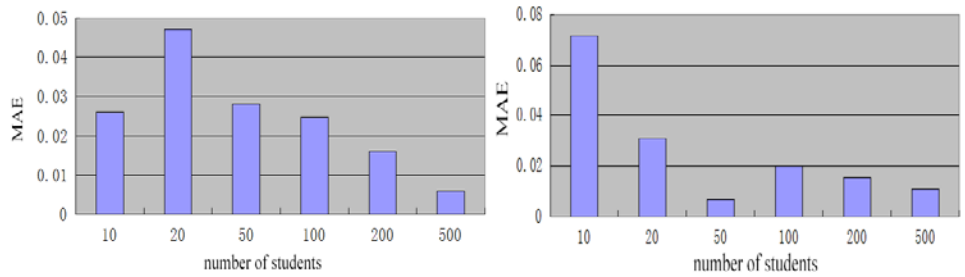

Figure 4. The mean absolute error (MAE) obtained with 50 (left) and 100 (right) groups

4. Conclusion and Future Work

In this paper, we proposed a new algorithm to produce effective and personalized learning material recommendations for students, aiming to (1) classify and organize the learning materials (uploaded by tutors) to different material concepts in terms of relevance; and (2) recommend students effective learning materials from which they can gain the most benefits in a short time, taking into account both the similarity between a learning material and a promising 'good' one, and the personalized usefulness of a learning material according to the time-aware ratings and difficulty levels reported by similar users. The experimental results based on our simulations show that our approach may work effectively to generate beneficial recommendations in terms of knowledge gain. In addition to the user-related features such as the learning ability, in the future we intend to incorporate other features, e.g., "education background" and "types of learners" to further improve our approach.

References

Champaign, J., Zhang, J. & Cohen, R. (2011). Coping with Poor Advice from Peers in Peer-Based Intelligent Tutoring: The Case of Avoiding Bad Annotations of Learning Objects. *Proceedings of the International Conference on User Modeling, Adaptation and Personalization.* UMAP.

Doan, T., Zhang, J., Tjhi, W.C. & Lee, B.S.. (2011). Analyzing Students' Usage of E-learning Systems in the Cloud for Course Management. *Proceedings of the 19th International Conference on Computers in Education.* ICCE.

Ghauth, K. I., & Abdullah, N. A. (2010). Learning materials recommendation using good learners' ratings and content-based filtering. *Educational technology research and development, 58*(6), 711-727.

Guo, G., Zhang, J. & Thalmann, D. (2012). A Simple but Effective Method to Incorporate Trusted Neighbors in Recommender Systems. *Proceedings of the 20th International Conference on User Modeling, Adaptation and Personalization.* UMAP.

Guo, G., Zhang, J., Thalmann, D. & Yorke-Smith, N. (2013). Prior Ratings: A New Information Source for Recommender Systems in E-Commerce. *Proceedings of the 7th ACM Conference on Recommender Systems.* RecSys.

Guo, G., Zhang, J. & Yorke-Smith, N. (2013). A Novel Bayesian Similarity Measure for Recommender Systems. *Proceedings of the 23rd International Conference on Artificial Intelligence.* IJCAI.

Zhang, J., Tjhi, W.C., Lee, B.S., Vassileva, J. & Cooi, K.C.. (2010). A Framework of User-Driven Data Analytics in the Cloud for Course Management. *Proceedings of the 18th International Conference on Computers in Education.* ICCE.