A Peer matching tool using social media

Floris Verburg Freek van Tienen Marijn Goedegebure Dragos Datcu Leon Rothkrantz

Abstract: *In distant learning students are remote in place and time. In case students are not familiar to each other, don't meet each other in real life at Universities, social media can provide a virtual meeting place. One of the problems is how to find best matching study partners in collaborative study activities. In this paper we describe a tool which recommends best matching students taking care of abilities and personal characteristics of students and requirements set by the teachers. The model of the matching tool will be described and the first results of user evaluation.*

Key words: *Distant learning, peer matching, social media, matching algorithm, supporting system, user test.*

INTRODUCTION

It is well known that many students come to the University to visit lectures because it provides them the opportunity to meet peers and to have discussions also about less academic topics. In e-learning the meeting place at the University is partly replaced by a virtual meeting place. The communication between teachers-students and students-students is facilitated by e-learning systems such as BlackBoard. But recently we can observe a huge rise of social Media. Students report about their daily activities, opinions, problems using social media such as FaceBook, Twitter , WhatsApp etc.

At this moment social media play a limited role in the academic process of teaching and learning. But social media offers great opportunities in e-learning, especially around the development of MOOCs (Massive Open Online Courses). In e-learning students, are remote in place and time. They follow the lectures using all sort of electronic devices.

Social media can enable communication and cooperation between students. It is possible to design social learning around specific courses. Students can use these networks as academic communication media, to discuss their study and learning activities, discuss topics from the lectures, put forward their questions and get help and support from fellow students. Students should feel part of a learning community of "friends" with a high feeling of presence [3,4].

In many courses students have to cooperate with fellow students in assignments and project work. Usually students make a selection out of their friends. Sometimes the teacher defines projects group at random or based on academic performance. In this paper we will not discuss the problem of optimal group formation. The goal is to form groups of students with their individual characteristics and abilities such that the group includes a minimum of abilities and characteristics to perform group assignments successfully. Our algorithms supports the choice of students but taking care of requirements and preferences set by teachers and students The question is how to find the best matching fellow students, especially students outside the university and distributed all over the world in case of MOOCs. After registration and enrolment in a course, students and their e-mail addresses are known. Inspecting the profiles from all these students via Facebook is too time consuming, therefore supporting matching tools are needed. At this moment there are many online dating sites trying to find the best matching life partner, our interest is finding the best matching study partners. A possible option is to extract features from student profiles via Facebook or LinkedIn, train classifiers or other matching technologies and take care of preferences and expert knowledge. In our case the application will recommend possible study partners and students have to make their choice. Our recommendation algorithm is custom designed. The algorithm uses the skills and personality characteristics of students, requirements and preferences defined by the teacher to find a best recommendation.

The outline of the paper is as follows. In the next section we will report a short literature review. Then we present our model of the application including the recommendation algorithm. After that we report some implementation details. Finally we present an evaluation of our application by users and end this paper with conclusions and future work.

LITERATURE SURVEY

In the e-learning community there is much interest in cooperative learning. Some researchers report about educational projects how students are able to communicate and cooperate via social media.

In [6] Knutas et al. states that benefits of collaborative learning and gamification methods have been used to motivate students towards achieving course goals in educational setting. However different users prefer different game elements and rewriting approaches. In his paper he presents an evidence-based method and a case study whre interaction analysis and k-means clustering is used to create gamification preference profiles. In another paper Knutas et al. [7] present in a case study an approach for using gamification elements to increase online student cooperation. An online discussion system was added and the actions in the discussion system were analysed and compared with user profiles and a student survey.

Dalsgaard et al. [3] discusses the potential of social networking within cooperative online education. He states that transparency is a unique feature of social networking. It provides students insight into each other's actions. The authors argue that cooperative learning can be supported by transparency. The authors consider transparency as means to promote affinity to learning community. They consider next to affinity, social presence as an important concept. It is important that students are visible and accessible. The learning system should suggest partners that make cooperation interesting. Dalsgaard argues that the pedagogical potential of social networking is the possibility to create awareness among students. Morten Flate Paulsen [4] researched in online education the combination of individual freedom and flexibility and meaningful cooperation and social unity. Purser et al. [5] report about peer-to-peer learning online, describing the role of cooperative student managed groupings in successful learn-by-MOOC experiences.

In his paper [1] Mortensen presents an equilibrium explanation of match formation and separation, based on the principle of voluntary pairing and competitive conditions. Search theoretic models that take account of meeting costs and initial uncertainty about match values have been developed to explain separation behaviour. McKinlay [2] researched online data-sites and discovered that one of these dating sites sorted people into profiles using the answers to thousands of questions posed by other users on the site. By creating fake profiles and writing programs to answer questions he discovered the underlying algorithm and was able to create successful matching profiles. He used collaborative filtering by collecting the preferences of many people, and grouping them into sets of similar users.

Based on the papers [8,9] there are roughly five kinds of algorithmic recommendation approaches:

- Demographic filtering
- Collaborative filtering
- Content based filtering
- Knowledge based filtering
- Hybrid filtering.

Different classes use different kinds of resources as is displayed in the next figure 1.

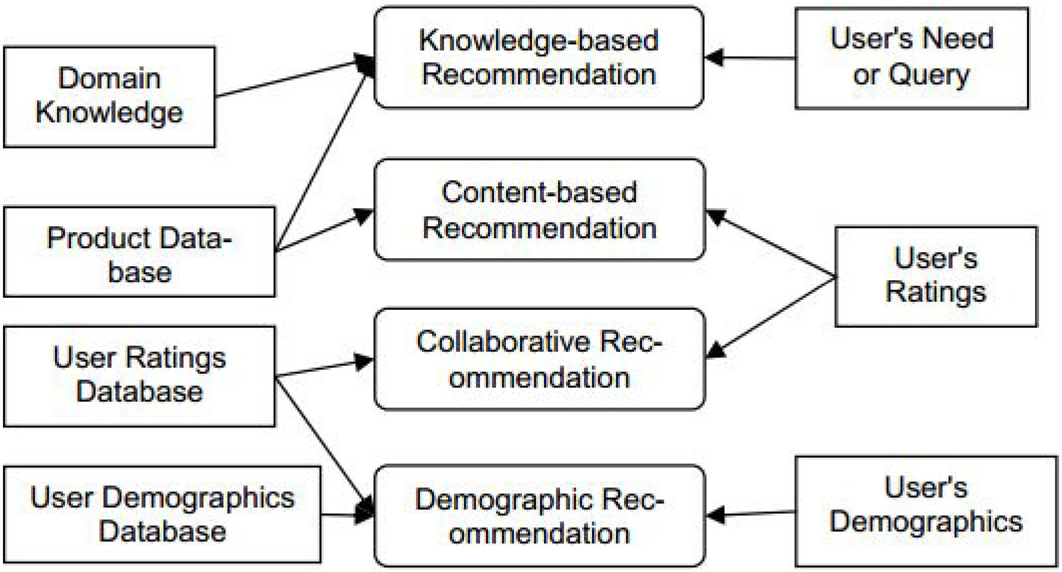

Figure 1. Overview of algorithmic recommendation systems, their resources

MODEL PEERMATCHING TOOL

Our peer-matching tool is centred around a specific e-learning course installed at a server. Students have access to the course and after enrolment the peer-matching tool is introduced to them. Students are able to download the Peer-matching App on their smart phone or other mobile devices. But before the matching algorithms starts, the teacher and its students have to provide some information. This is organised in several modules which will be discussed now:

Teacher module Peer-matching is focussed on practicals or group work. A group of students is supposed to complete some assignments. In a traditional classroom environments most students know each other and practical groups are formed spontaneously. At the end of the formation process there is usual a group of leftovers who decided to form a group. Another option is that the teacher leads the formation process. To complete assignments successfully composing students should have some abilities and personal characteristics. In case of a project around developing software, students should have some programming abilities, in case of data analysis some mathematical/statistical abilities. Every group is supposed to have one or more students with management capacity and writing abilities to write project reports. The teacher has to decide which abilities and which personal characteristics are needed in a group and defines some requirements. He is able to set some thresholds for average group abilities and/or requirements for individual students.

Student module Students are assumed to provide information about their abilities and personal characteristics. Abilities as programming, mathematics, writing reports etc can be defined by moving sliders on a scale from 1-100. In the current version these abilities are computed from the data from the academic record. Curriculae of computer science share common core courses from which abilities can be computed. Students have the freedom to change the automated computed scores.

To compute personal characteriscs a version of the wellknown big five personalites (http://www.123test.nl/bigfive/) is implemented . Students have to answer 5x4 questions on a 5-point Likert scale and based on the given answers the personal abilities scores are computed on the bigfive dimensions/personality characterstics **(**1) Extraversion, (2) Agreeableness, (3) Conscientiousness, (4) Emotional Stability, or (5) Intellect/Imagination in other words altruism, openminded, extraversion, neuroticuism, conscientiousness.

RECOMMENDATION ALGORITHM

Before we start explaining the algorithm we will first describe how to calculate the distance for a certain practical group. Calculating the distance of a practical group is done by going through the skills defined by the practical teacher. We will then calculate the average of the values of the skills of all the users in the practical group and the users' practical group combined. Then all the averages are multiplied with each other and then the root of the amount of skills in the practical is taken. This resulting number will be defined as the distance.

The matching algorithm starts with a practical and user. It will start with going through the different practical groups in the practical. To simplify the algorithm we chose to make sure every user is in a practical group. This means that when a user doesn't have any practical partners he is still in a practical group with only himself in it. While going through the different practical groups it will calculate the distance as described in the previous paragraph. After calculating the distance with every other practical group and saving this in a map, this result is sorted by descending distance. Then we start recommending practical groups to the user starting by the top of the list. The formal definition of this algorithm is visible in Figure 2.

To analyse the complexity we need the following measurements. We define *n* as the amount of practical groups in a practical. We define *m* as the amount of skills defined in a practical. We define *j* as the amount of users in a certain practical group plus the amount of users in the user's own practical group. Since in the first loop we go through the practical groups to calculate the distance we start with a time complexity of $O(n)$. Inside this loop we go through each of the skills defined in the practical to calculate the average which will give us $O(n*m)$. In order to calculate this average we need to go through all the users in our practical group and the practical group examined. This will lead to a time complexity of $O(n*m*j)$. As one of the last steps we still need to sort the list of the practical groups which will be done in a time complexity of $O(n*log(n))$. All this combined will lead into a total time complexity of $O(nmj + nlog(n))$ which for small *m* and *j* will lead to a total time complexity of $O(n*log(n))$.

function Recommend($practical$, $user$)
    $resultList \leftarrow$ Empty map of PracticalGroup and distance
    for all $practicalGroups$ in the $practical$ **do**
        $distance \leftarrow 0$
        for all $skills$ in the $practical$ **do**
            $average \leftarrow 0$
            for all $users$ in the $practical$ **do**
                $average \leftarrow average + SkillValue(user, skill)$
            end for
            for all $users$ in the $user$ practical group **do**
                $average \leftarrow average + SkillValue(user, skill)$
            end for
            $average \leftarrow average/(amount\_of\_users)$ · The amount of users in the practical
            $distance \leftarrow distance * (average - SkillValue(Practical, Skill))$
        end for
        $distance \leftarrow distance$(to the power 1/($amount_of_$skills) · The amount of skills in the practical
        $resultList \leftarrow resultList + (practicalGroup, distance)$
    end for
    Sort the $resultList$ by descending $distance$
    return $resultList$
end function

Figure 2. The recommendation algorithm

USER TEST

The purpose of the user test was to test the system for its robustness and usability in the different kinds of input a user can generate. The test also requested the user to make extensive use of the interface that we provided. The user was asked to test different functionalities :

- Registration
- Logging in
- Register to practical using url
- Viewing the practical that they registered to
- Invite other students to join their practical group.

In total 32 third year Computer Science students took part in the user test. After the test the students were requested to fill in a questionnaire with 22 questions, 18 closed questions and 4 open questions. We take a sample of the questions and discuss the results:

a. Did you successfully register, login?
b. Did you successfully enter your skills?
c. Do you know what a MOOC is?
d. What do you think about the recommendations?
e. Did you successfully invite other students?

Ad a. The register and login success rates were very high (more than 90%). This is not a big surprise. All the respondents were advanced Computer Science students, very familiar with these topics. In case non technical students will be tested we expect a drop of the success rates.

Ad b. Students had a more optimistic view of their abilities than the results from the academic record. The abilities were automated computed but corrected by 35% of the students. Even more students had their doubt about the usefulness of personal characteristics. The online big five personality test was interesting, but many students didn't understand the dimensions and the usefulness for project work. Next future the dimensions have to be translated to more relevant dimensions as leadership, communication and writing abilities.

Ad c. It proves that about 40% of the respondents know what a MOOC is and tested one. The rest 60% of the students had no idea what a MOOC is and didn'tread about it. This is a disappointing result for a University offering 10 MOOC's with a high success rate and user satisfaction. Regular students are evidently not impressed.

Ad d. It proves that 21% of the students scored "very useful", 43% scored "moderately useful and 36 % scored the recommendations as not useful. This result is rather disappointing. But reading the comment it proofs that students with a negative score prefers to choose the common method, choosing their friends as practical partners. This a common finding in experiments about the use of social media if students have the option to use the old familiar methods. But we have to realise that in case of MOOCs with thousands of participants the common method doesn't work because students are not familiar to each other.

Ad e. It proves that half of the students tried to invite their friends. Other students reported that they like the objective peer matching tool because usually they belong to the dropouts.

It would be very interesting to analyse the existing methods of peer matching. Unfortunately because of privacy rules the data of the matching results are not available for research.

CONCLUSIONS AND FUTURE WORK

In this paper we presented a peer matching algorithm, which can be used to form small groups of students to do practicals/group assignments. The algorithm is useful in case students are not familiar to each other and don't have the option to meet each other in real life. The target applications are MOOCs. Students have to provide personal characteristics and a description of abilities. The developed peer matching algorithm suggests partner students taking care of the requirements of the teachers. The students make the final choice.

It proves that teachers have different opinions about the requirements in the group formation process. The modular approach of the developed system enables easy adaptation. In the next release a weighted sum of abilities will be available. The relation personal characteristics computed via the big five model and practical or group work was not clear. In the next release personal characteristics will be defined directly related to the cooperative work in groups, such as leadership, flexibility, writing and presentation abilities. At this moment the system has been tested on a small group of students. Next future it will be tested on one of the developed MOOCs at Delft University of Technology. A start up company will develop the current prototype to a full operational system.

REFERENCES

[1] Mortensen, D. Matching: finding a partner for life or otherwise, inaugural lecture at the University of Chicago, America Journal of Sociology, vol. 94, 1988.

[2] McKinlay, C. Optimal Cupid: Mastering the Hidden Logic of OkCupid dissertation, University of California, 2014.

[3] Dalsgaard, Paulsen, Transparency in Cooperative Online Education, The International Review of Research in Open and Distance Learning, Vol. 10, No 3, 2009.

[4] Paulsen, Morton, Flate, Cooperative freedom and transparency in online education, Proceedings ICICTE, 2012.

[5] Purser, E., A. Towndrow,. A. Aranguiz . Realising the Potential of peer-to-peer learning: Taming a MOOC with social media. E-Learning Papers, vol. 33, 2012.

[6] Knutas, A., Ikonen, J.,Maggiorini, D., Ripomonti L., Porras J. Creating Software engineering student interaction profiles for discovering gamification approaches to improve collaboration. CompSysTech, 2014.

[7] Knutas, A., J. Ikonen, U. Nikula, J.Porras, Increasing collaborative communications in a programming course with gamification: a case study. CompSysTech, 2014.

[8] P. Brusilovsky, P., E. Millán, The Adaptive Web, Springer, 2007.

[9] R. Burke, R., Knowledge-based recommender systems, Encyclopedia of library and information systems 69, 175 , 2000.

ABOUT THE AUTHORS

Floris Verburg BSc, Freek van Tienen BSc, Marijn Goedgebure, BSc, Department of Computer Science, Delft University of Technology

Dragos Datcu, PhD, Faculty of Technology, Policy and Management, Delft University of Technology, Jaffalaan 5, 2628BX Delft, The Netherlands, E-mail: D.Datcu@tudelft.nl

Prof. Leon Rothkrantz, Department of Intelligent Interaction, Delft University of Technology, Phone +31 15 278 7504, E-mail l.j.m.rothkrantz@tudelft.nl

The paper has been reviewed.