A Recommender System for On-line Course Enrolment: An Initial Study

Michael P. O'Mahony
School of Computer Science and Informatics
University College Dublin
Belfield, Dublin 4
Ireland
michael.p.omahony@ucd.ie

Barry Smyth
School of Computer Science and Informatics
University College Dublin
Belfield, Dublin 4
Ireland
barry.smyth@ucd.ie

ABSTRACT

In this paper we report on our work to date concerning the development of a course recommender system for University College Dublin's on-line enrolment application. We outline the factors that influence student choices and propose solutions to address some of the key considerations that are identified. We empirically evaluate our approach using historical student enrolment data and show that promising performance is achieved with our initial design.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering

General Terms

Algorithms, Experimentation, Measurement

Keywords

Course Recommender System, Collaborative Filtering, Content-based Filtering, Academic Advising

1. INTRODUCTION

The structure and delivery of degree programmes within University College Dublin (UCD) has recently been reorganised and is now delivered on the basis of a fully modular, semesterised and credit-based curriculum. While this new approach provides increased choice and flexibility and allows students the ability to personalise their studies, challenges have arisen with regard to enabling students to appreciate the range and diversity of modules (courses) that are available to them. In particular, the current enrolment system makes it difficult for students to locate course options that might best fit their individual niche interests.

In other domains, the benefits of deploying recommendation technology to assist users in finding relevant items is well understood [6, 7]. More recently, research has been conducted into developing such technology for course recommender systems. For example, CourseAgent employs a social navigation approach to deliver recommendations for courses based on students' assessment of their particular career goals [3]. AACORN, which applies a case-based reasoning approach to course recommendation, has been proposed in [10], while the RARE system combines association rules together with user preference data to recommend relevant courses [1].

In this paper, we report on our work on developing collaborative recommendation technology for integration into UCD's existing on-line enrolment application. Given the challenges and constraints of integrating this technology into an existing live environment, this work is in its initial stages but the domain offers great potential and scope for future research and development. Here, we describe our current progress and present some encouraging preliminary evaluation results.

The paper is organised as follows. Section 2 provides an overview of enrolment requirements for UCD students. Section 3 describes recommendation algorithms to facilitate the enrolment process. These algorithms are empirically evaluated in Section 4 and conclusions are presented in Section 5.

2. MODULE ENROLMENT OVERVIEW

In UCD, students normally study and gain credits for 12 modules in each academic year. Of these, 10 modules are from within their core area of study (e.g. Computer Science, Economics etc.). In addition, students choose 2 *elective modules* which may be selected from the broader curriculum. Students have therefore the opportunity to deepen their knowledge by focusing on modules from within their subject area or broadening their knowledge by taking modules from elsewhere in the curriculum. For example, Science students may select Business modules, Medical students may choose Philosophy modules, Law students may select additional Law modules etc. In an Irish context, this approach represents a significant change to the more limited study options that are typically on offer.

Thus, once students have selected their core area of study, they are also presented with a wide variety of elective modules from which to choose. A number of factors can be identified which are likely to influence such choices:

- *Interests and academic knowledge*: students will naturally vary in these regards and base module choices accordingly.

RecSys'07, October 19–20, 2007, Minneapolis, Minnesota, USA.

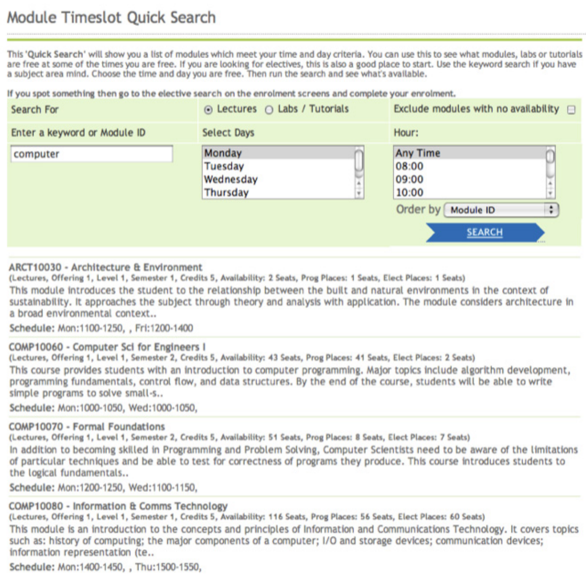


Figure 1: A screenshot of the current elective module search interface, showing elective modules associated with the keyword "computer" that are available at any time on Mondays.

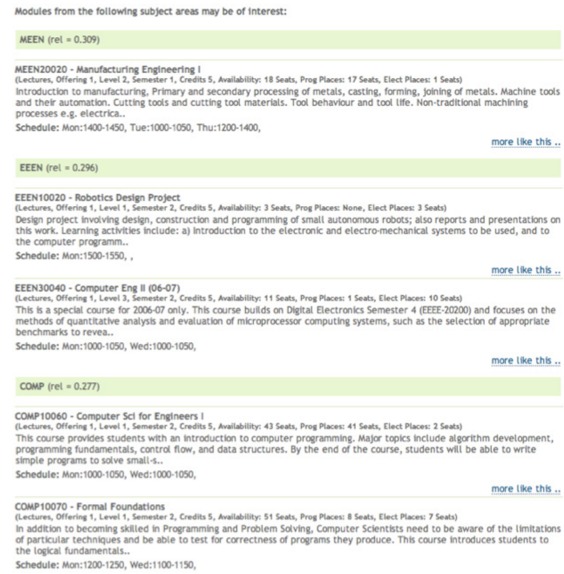


Figure 2: A screenshot of output from the collaborative recommender, showing elective modules grouped and ranked by subject code.

- *Career goals*: students may focus within their core area of study or opt for more diverse knowledge.

- *Module pre-requisites and co-requisites*: students may choose modules strategically, e.g. by selecting a module that offers a wide flexibility of choice in subsequent years.

- *Ability to progress study*: students may be less inclined to make particular choices if higher-level modules are not offered in subsequent years.

- *Difficulty and format of module*: the complexity of course material, the degree of practical work involved, the assessment criteria applied etc. may influence choices.

- *Awareness of options*: many students and, in particular, new entrants to the university, may be unaware of the range of choice and opportunity that is available.

- *Availability of places and timetable clashes*: in such cases, related or similar modules should be easily identifiable.

The University conducted a formal review of module enrolment in 2005. A survey of 1st year students elicited 820 responses, which corresponded to approximately 20% of the total 1st year student body. The survey revealed that 47% of students selected elective modules outside of their core degree, with a view to studying "something interesting", or because of a desire to study a specific subject outside of their core degree.

It was found that 24% of students did not get a place in any of their preferred elective modules, while 14% did not get a place on some of their preferred electives. 20% of students attributed not getting their preferred elective choices to timetable clashes, while 10% of students believed the reason they did not get their preferred electives was due to a lack of places.

Further, there is concern that, given the limitations of the existing module search interface, students are simply unaware of the full range of options available, resulting in poorly-informed choices being made.

These findings are noteworthy in the sense that they reveal a clear desire on the part of many students to select diverse and interesting courses, while also highlighting the need to better facilitate module search. In the next section, we describe the elective module search interface offered by the current enrolment system and describe our initial work on introducing recommendation technology into the system.

3. MODULE RECOMMENDATION

The existing elective module search interface is shown in Figure 1. Search can be refined by inserting a keyword or a specific module ID (if known). In addition, particular days and time-slots can also be specified. The output is a list of elective modules which match the search criteria. Output is ranked either alphabetically by module ID, by module level (i.e. 1st, 2nd, ... year module) or by availability. There are a number of limitations associated with this form of search:

- Search output is non-personalised; thus the same results are presented to all students, irrespective of the core area of study.

- Search output can be filtered by inserting keywords or by specifying module ID's; such filtering, however, necessarily requires a degree of fore-knowledge on the part of students.

- A "more like this" feature is not provided for the returned results; such a feature would be useful in the event of timetable clashes occurring or when no places are available for particular modules.

Core Modules (C)

COMP10	COMP20	ECON10	ECON20	ECON30	MATH10	MATH20
1	0	1	1	0	0	0

$$\Downarrow$$

Core Subject Codes (C^*)

COMP	ECON	MATH
1	2	0

Figure 3: Abstracting a student profile from the core module to the core subject code level. The elective module component of a profile is similarly abstracted.

To address the above limitations, we have implemented a collaborative-filtering style recommender system which suggests elective modules based on the past choices of like-minded students. In addition, we have developed a simple content-based recommender which recommends similar modules based on keyword similarity (i.e. a "more like this" feature). These approaches address some of the key factors that were identified in Section 2 and, in particular, those factors that were frequently raised in student feedback.

3.1 Collaborative Recommender

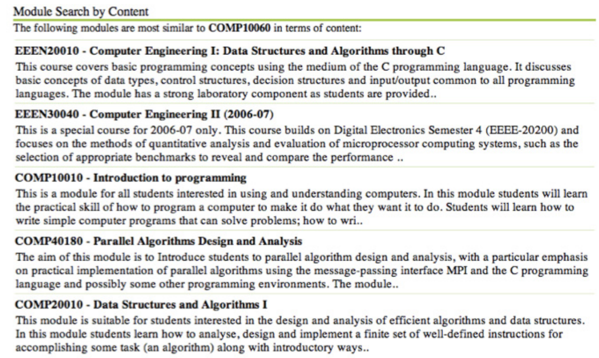
We propose a variation on the widely-used item-based collaborative filtering algorithm described in [4]. The objective of our module recommender system is to recommend *elective* modules to students based on the *core* modules that they have selected. Thus, we construct student profiles using two matrices, C and E, which record the core and elective modules that are selected by students, respectively. In each case, the $(i, j)^{th}$ entry is set to 1 if student i has selected module j and 0 otherwise.

The next step in the algorithm is to abstract each student's profile to the *subject code* level. Every module is associated with a unique identifier – e.g. `MATH10060`, where `MATH` is the subject code and `10060` is the course number. We construct two additional matrices, C^* and E^*, where the $(i, k)^{th}$ entry in each matrix represents the number of times that student i has selected modules of type subject code k. Figure 3 shows a simplified example of the abstraction process.

The pairwise vector similarity [4] between the columns of matrices C^* and E^* is now computed, resulting in an $m \times n$ similarity matrix S, where m and n are the number of core and elective subject codes in the system, respectively, and $S_{i,j}$, gives the similarity between (core) subject code i and (elective) subject code j. Note that with this approach the similarity matrix is not symmetric, i.e. $S_{i.j} \neq S_{j,i}$.

Recommendations are made for a particular student as follows. For each elective subject code j, compute the total similarity between all core subject codes selected by the student and j. Individual elective modules are then output, *grouped* and *ranked* by subject code. An example of such output is shown in Figure 2.

One rationale for profile abstraction is we believe that the relatively *course* ranking of recommendations by subject code is generally more useful and informative to students rather than ranking by specific module ID.


Module Search by Content

The following modules are most similar to **COMP10060** in terms of content:

EEEN20010 - Computer Engineering I: Data Structures and Algorithms through C

This course covers basic programming concepts using the medium of the C programming language. It discusses basic concepts of data types, control structures, decision structures and input/output common to all programming languages. The module has a strong laboratory component as students are provided..

EEEN30040 - Computer Engineering II (2006-07)

This is a special course for 2006-07 only. This course builds on Digital Electronics Semester 4 (EEEE-20200) and focuses on the methods of quantitative analysis and evaluation of microprocessor computing systems, such as the selection of appropriate benchmarks to reveal and compare the performance ..

COMP10010 - Introduction to programming

This is a module for all students interested in using and understanding computers. In this module students will learn the practical skill of how to program a computer to make it do what they want it to do. Students will learn how to write simple computer programs that can solve problems; how to wri..

COMP40180 - Parallel Algorithms Design and Analysis

The aim of this module is to Introduce students to parallel algorithm design and analysis, with a particular emphasis on practical implementation of parallel algorithms using the message-passing interface MPI and the C programming language and possibly some other programming environments. The module..

COMP20010 - Data Structures and Algorithms I

This module is suitable for students interested in the design and analysis of efficient algorithms and data structures. In this module students learn how to analyse, design and implement a finite set of well-defined instructions for accomplishing some task (an algorithm) along with introductory ways..

Figure 4: A screenshot of the "more like this" recommendation feature, in this case showing modules that are similar in terms of content to Computer Science module `COMP10060`.

3.2 "More Like This" Recommender

The key motivation for the "more like this" recommender is to facilitate students to find similar modules to those that are listed, particularly in cases where no places are available or where timetable clashes occur. The following approach is employed. Each module in the system has associated text fields detailing the module description and learning outcomes. After the removal of stop words and the application of suffix stripping [8], module-module similarities are computed using the well-studied information retrieval algorithm described in [9].

A "more like this" link is associated with each recommended module, as shown in Figure 2. The result of selecting this link for module `COMP10060` (an introductory course to computer programming for Engineering students) is shown in Figure 4, where similar modules are listed and ranked in terms of content-based similarity.

4. EVALUATION

In this section, we evaluate the collaborative recommender algorithm described in Section 3.1[1]. The evaluation dataset consisted of modules selected by 1st year students from all degree programmes in the 2006/07 academic year. Dataset statistics are shown in Table 1.

Test and training sets consisting of randomly selected student profiles were drawn from the dataset and a 10-fold cross validation was performed. Test sets were of size 10%. In all cases, recommendations were made using training set data only. To evaluate the algorithm, we use *recall* – the percentage of elective subject codes selected by test set students that are present in top-N recommended lists – and *coverage* – the percentage of test cases where the algorithm is able to make recommendations.

For top-10 recommendation lists, the recall achieved by the algorithm was 66%. This result means that, on average, 1.33 out of the 2 elective modules that were actually selected by students belonged to the top-10 recommended subject code groups. This compares very favourably to a recall value of only 10% provided by a benchmark non-personalised approach, where modules were simply grouped and ranked by

[1]We leave to future work an evaluation of the content-based "more like this" recommender.

Table 1: Dataset Statistics

# Students	2,597
# Transactions	30,422
# Modules	2,088
# Core Subject Codes	71
# Elective Subject Codes	79

subject code in alphabetical order (as is the case with the current search interface).

The algorithm achieved 100% coverage, meaning that it was able to compute non-zero similarities between all test set profiles and at least 10 (as required to fill top-10 recommendation lists) elective subject codes. On average, 60 out of a possible 79 similarities computed for test profiles were non-zero, with a standard deviation of 23. This strong performance was due to the abstraction of student profiles to the subject code level; since there were less than 80 unique subject codes as opposed to 2,088 unique course modules in the system (see Table 1), the sparsity of the data on which the algorithm operated was significantly reduced. Figure 5 shows a histogram of the number of non-zero similarities that were computable for each core subject code. All 71 core subject codes had a non-zero similarity with 1 or more elective subject codes, while 85% had in excess of 10 non-zero similarities. At the higher end, more than 70 non-zero similarities (out of 79) were computable for 3 core subject codes. Consequently, the density of the similarity matrix in this domain was much higher (35%) compared to that typically found in other application scenarios (1% or less), and thus the algorithm was able to deliver high quality recommendations as reported above.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented an overview of the enrolment requirements for UCD students and we have identified factors which influence module selection. We have described our initial work on developing recommendation technology to facilitate and enhance the on-line module selection process. When our collaborative algorithm was evaluated using historical enrolment data, it was found to provide very encouraging performance in terms of both recall and coverage.

There is rich scope for further work on recommendation in this particular domain. For example, a diverse range of alternative recommendation algorithms have been proposed in the literature (e.g. association-rule based algorithms [1, 5], hybrid approaches [2] etc.). In addition, the domain provides an environment for developing rich user models – for example, modeling students' interests and goals, analysing module selection patterns over multiple years and constructing models designed to suit the structure and complexities of different degree programmes etc.

Our recommendation algorithms will be in place prior to student enrolment at the beginning of the next academic year. We await the student feedback that will be forthcoming and plan to develop algorithms with more advanced features in time for the next release of UCD's enrolment application.

6. ACKNOWLEDGMENTS

This work was carried out with the support of Science Foundation Ireland under Grant No. 03/IN.3/I361, which is gratefully acknowledged. We would also like to thank our reviewers for their valuable comments and suggestions.

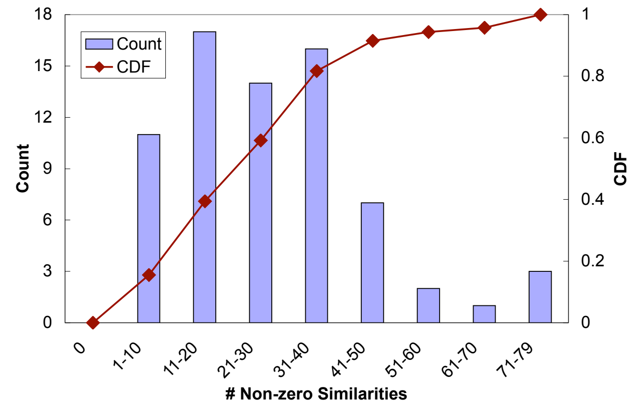


Figure 5: Histogram and cumulative distribution function (CDF) of the number of non-zero similarities that were computable for core subject codes.

7. REFERENCES

[1] N. Bendakir and E. Aimeur. Using association rules for course recommendation. *In Proceedings of the AAAI Workshop on Educational Data Mining*, pages 31–40, July 16–17 2006.

[2] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User–Adapted Interaction*, 12(4):331–370, November 2002.

[3] R. Farzan and P. Brusilovsky. Social navigation support in a course recommender system. *In Proceedings of the 4th International Conference on Adaptive Hypermedia and Apadtive Web-based Systems*, pages 91–100, June 21–23 2006.

[4] G. Karypis. Evaluation of item–based top–N recommendation algorithms. *In Proceedings of the Tenth ACM International Conference on Information and Knowledge Management (CIKM'01)*, pages 247–254, November 5–10 2001.

[5] W. Lin, S. Alvarez, and C. Ruiz. Efficient adaptive-support association rule mining for recommender systems. *In Data Mining and Knowledge Discovery*, 6:83–105, 2002.

[6] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *In IEEE Internet Computing*, 7(1):76–80, 2003.

[7] B. Miller, I. Albert, S. Lam, J. Konstan, and J. Riedl. Movielens unplugged: Experiences with a recommender system on four mobile devices. *In Proceedings of the 17th Annual Human-Computer Interaction Conference*, September 8–12 2003.

[8] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[9] S. E. Robertson and K. Spark-Jones. Simple, proven approaches to text retrieval. *Technical Report TR356, Cambridge University Computer Laboratory*, 1997.

[10] J. Sandvig and R. Burke. Aacorn: A CBR recommender for academic advising. *Technical Report TR05-015, DePaul University*, 2005.