

Recap

In Assignment 1, I outlined my interest in applying technology to the field of advising and gave a brief summary of the non-technological bases of that field. I described the many ways advising is carried out in traditional programs and further touched on the Google+ community and the UOCRSS – the unofficial OMSCS course review spreadsheet – which are the two main tools for advising in the OMSCS program. I argued for focusing on the OMSCS as an ideal setting for experiments with advising technology, because of the deficiencies in those tools but also because of both the scale and the technology-mediated nature of the program.

In this assignment, I'd like to focus on current applications of technology to advising, of which I have identified two major groups: *degree audit* software, which seeks to help advisees or advisors ensure that a particular student is making appropriate progress towards his or her degree, and *community* software, which seeks to streamline communication from advisors to advisees and perhaps stimulate discussion among advisees as well.

I'd like to close by touching on some technologies that are widely applied in other areas but haven't seen much application in the field of academic advising – yet.

Degree audits

A *degree audit package* is a software tool that allows students, advisors, or both, to determine what unfulfilled degree requirements prevent a student from graduating.

Many different software packages offer degree audit (and related) functionality: Degree Audit Reporting System (DARS) Ellucian Degree Works[1], u.achieve[2], AgileGrad[3], PeopleSoft Campus Solutions (by Oracle)[4], CAMS Enterprise[5], and likely several others which are not as widely-used. Most of these DAPs support both self-service and advisor-oriented scenarios. Common use cases include exploring potential major or degree program changes, and the applicability of transfer credits; they may also be data-mined for institute-level decision-making, such as planning course offerings.

Generally these systems will look at the student's existing transcript and transfer credits and list remaining unfulfilled requirements. They may also allow specifying individual "planned" courses for future terms, and verifying that those courses will cumulatively be sufficient for graduation. They do not suggest courses to take – this remains the prerogative of the human advisor.

To see this illustrated, let's explore Degree Works – it is representative of the genre, and is also the system used at Georgia Tech (thus a natural point of comparison for any automated advising solution we create for OMSCS students).[6] Degree Works is available to all Georgia Tech students and their advisors, although at present it is

only *relevant* to undergraduates [see below]. Degree data was entered using a custom Ellucian tool called *Scribe*. It includes an eponymous domain-specific language for describing curricula that allows encoding all requirements and restrictions for every degree, major and minor offered by the school. This encoding process is performed at the institute level and does not need to be repeated for individual students. Ellucian has begun offering a “scribing service” and it is not clear whether individual customers are still permitted to scribe their own degree descriptions, so this procedure may be somewhat outdated.[7]

The Georgia Tech Degree Works installation integrates with T-Square (Georgia Tech’s flavor of the Sakai learning management system) to automatically load transcript data for each student. The basic worksheet provided allows the student to perform a degree audit based on their past completed courses, their transferred credits, and their courses in progress. The “What If” feature allows the student to see which degree requirements would remain if they changed majors or degrees. The “Look Ahead” feature allows the student to add planned courses to the default (transcript-based) list for purposes of either of the above two worksheets. Taken together, these interfaces support all of the common DAP use cases.

Degree Works shares the same limitation as other DAPs: it only allows the student to evaluate a prospective course of study, without offering any suggestions about how to piece it together in the first place. For our purposes, there’s another crucial limitation: the OMSCS degree requirements haven’t been scribed! This means that no OMSCS student can use any planning features of Degree Works.

Community management

A *community management package*, in the context of advising, is a software tool that combines and centralizes several channels of communication between advisors and advisees.

The current “state of the art” for advising community management is using a *course-management system* (often one already employed for the college or university’s actual classes) to serve as a combined forum/ mailing list and distribution channel for advising updates. Many CMSes have been repurposed in this way at various institutions, including Desire2Learn at Western Illinois University[9], a custom Blackboard module at Troy University[10], and Moodle at UOPeople[11] (until recently – see below). A few universities, mainly those with predominantly- or solely-online offerings, are also building advising discussion communities on Moodle or Piazza.

One interesting case study is the advising process at the University of the People (UOPeople). As a fully-online institution, one might expect UOPeople to be on the cutting edge of technology-assisted or mediated advising, and as a large not-for-profit institution, keeping costs down and leveraging the size of the student body is likely to be a key concern (as it is for OMSCS).

Until recently, UOPeople offered students three channels for advising discussion: a confidential e-mail address allowed for discussion solely with the official academic advising team, an Academic Advising Virtual Office (AAVO) Moodle forum allowed for mixed discussion between peers and advisors, and a Yammer forum for purely unofficial discussion among students. However, on March 12, 2015, the mixed discussion forums were shut down for reasons that were not publicly shared.[12] To date, I haven't been able to deduce anything useful about the closure; I'm continuing to pursue this angle because it could offer important cautions for other student-to-student advising platforms.

And now for something completely different: recommendation systems

Recommendation systems are automated tools, usually but not always based on machine learning, that consider a set of users and a set of items and predict which items each user will like.

Recommendation systems are traditionally broken into two categories¹:

- *Content-based filtering* is a set of techniques that suggest appropriate items for a user based on a profile of their preferences. This profile may be filled in explicitly by the user, or it may be inferred by the system, but it always takes the form of a set of specific item attributes and weights mapping how likely a user is to like or dislike an item with each trait.
The advantage of content-based filtering is that it can encapsulate pre-existing domain knowledge (for instance, my awareness that “language” is an important feature of a book and my ability to look up the release date of any album produced in the United States after 1920). That advantage is also its weakness, however; any features whose importance is not predicted by the designer lose their predictive power.
- *Collaborative filtering* is a set of techniques that suggest appropriate items for a user based on similarity to other users. In classical collaborative filtering, no explicit conclusions are drawn about the user's preferences; instead, recommendations are made by looking at the items the user has rated highly so far, finding other users who also rated those items highly, and then suggesting other items those users have liked that the focal user has not

¹ Actually, we often recognize additional categories, such as *search-based filtering*, in which the user explicitly inputs criteria for a recommendation request, *critique-based filtering*, in which the system presents a series of candidates and elicits user responses explaining why each is unsuitable, and *demographic filtering*, in which the system makes recommendations per demographic and a demographic profile is supplied for each user. However, content-based filtering and collaborative filtering are the most universal forms in tech today (though demographic filtering often comes up in traditional retail); I would argue that this is because they can be implemented based only on data inferred from system interactions the user will have regardless, whereas others tend to require large amounts of prior knowledge (demographics) or specific recommendation-focused user input (searches and critiques).

This is a truly fascinating domain, and it's not feasible to cover it in depth within this assignment; if you, the peer reviewer, are curious about the non-simplified picture, my highest recommendation goes to the Wikipedia page on the subject (https://en.wikipedia.org/wiki/Recommender_system). It is an excellent starting point with many wonderful links.

yet rated. Collaborative filtering is the basis for arguably the most famous recommendation system in the world: *customers who bought this item also bought....*

The major advantage of content-based filtering is that it can dynamically adapt to *any* domain of items; the major disadvantage is that there is a bootstrapping problem for both new users and new items, since neither starts with any ratings that can be used for prediction.

The most successful recommendation systems hybridize both models, as the strengths of each tend to cancel out the weaknesses of the other.

When a well-designed recommendation system is available, everybody wins: retailers, content providers or developers are rewarded with additional purchases or engagement, and the customer, audience member or user is rewarded with a wealth of new products that are ideally suited to his needs.

Because of this mutual benefit, recommendation systems are used in every major Internet-based application today. Some companies, such as Pandora and Facebook, are virtually impossible to imagine without a system for surfacing suggested content to their users – indeed, in extreme cases a recommendation engine can be a company’s entire business model. Some companies, such as Amazon.com and Apple [through its iTunes Store], would merely lose quite a bit of revenue without the recommendations that invite their users to make additional purchases. So ubiquitous are recommendation systems that even traditional physical retailers, such as grocery stores, have seen their power and begun using their loyalty programs to offer tailored coupons and other “light recommendations” at the checkout.

However, there are a few key areas where recommendation systems are still underutilized, and education is one of these.² To the extent that recommenders have appeared here it has been in the form of *intracourse* recommendations – for instance, in the service of adaptive/intelligent tutors (to choose which content should be covered next), or to provide students with suggestions for supplemental resources to address problem concepts or skills (Okoye et al., 2012). None of these systems has been widely commercialized.

Meanwhile, the only known proof-of-concept for recommending *courses* to students was completed 15 years ago (Geyer-Schulz et al., 2000)³, and is now defunct. One additional related system was found which used data mining to predict student performance (Vialardi et al., 2009); this does not represent a complete recommendation system but could easily serve as the basis for one. The major MOOC providers offer “recommendations,” but simplistically-generated ones: edX

² The others that spring to mind, like stocks and securities trading, present much more obvious and fundamental obstacles to a recommendation-based approach.

³ I didn’t discuss this case study in detail because, maddeningly, it appears that not only the recommender system but also the entire program that it served is defunct.

offers a digest of its most-popular courses (by enrollment); Udacity offers a set of pre-set curricula suggestions in the form of “nanodegrees”; Coursera’s recommendation algorithm is not public knowledge, but anecdotally their e-mails appear to suggest courses which share keywords with courses the user’s browsed recently.

No official analyses exist, but I can speculate on a few reasons that this application hasn’t taken off yet:

Adoption - a system that requires approval at the institutional level will face significant challenges. Development overhead is incurred in integrating with existing systems, and in adhering to FERPA and other regulations restricting the use of educational data.

Scale – to be successful, recommendations systems (particularly collaborative ones) must have access to a large data set, with some number of ratings for each course. Traditional institutions (which average 6,154 undergraduates split across dozens of degrees and majors) may not have a large enough pool of students to bootstrap an effective system. The problem is exacerbated when we leave the realm of accredited degrees and consider MOOCs; the “long tail” of MOOCs means that interested students are split across thousands of courses, and lower student engagement/investment means that individual students are less likely to take time to rate classes.⁴

Technological literacy – community colleges, which are otherwise excellent candidates for course recommendation systems because of their relatively limited numbers of available courses and degree objectives, support a disproportionately high number of adult or economically disadvantaged students. These students may have reduced access to and/or facility with computers and smartphones.

Competition with established practices – established programs by definition have existing ways of doing advising. If those time-honored methods achieve good results, why change?

The good news is: none of these is necessarily an obstacle to an OMSCS-focused technological advising solution. In the OMSCS, advising is driven from the student level (e.g. the Google+ group and the UOCRSS), so institutional approval is not needed for a new system as long as we opt not to integrate it with existing software. The very large scale of the OMSCS (almost 3,000 students in the same degree program, with only minor variations among specializations) means that even if only a minority supports a new system, it will still likely have enough data per course to offer useful recommendations. The fact that OMSCS is an internet-mediated program strongly suggests a high level of tech-savviness in its students, and the

⁴ This has not been formally studied but has anecdotal support; for instance, consider www.class-central.com, a MOOC indexing site that also allows users to wishlist classes and submit ratings and reviews. Fewer than 50% of their computer science courses and social science have any ratings listed. For 5 randomly-selected courses with ratings, fewer than 50% of ratings also included a text review.

youth of the program means that very few practices have become so entrenched that students and staff will hesitate to try new alternatives. Taken together, these facts tend to confirm the claim made in the previous assignment: the OMSCS is an ideal setting for experiments with advising technology.

Next steps

1. Following up with DegreeWorks to find out why scribing didn't go forward for graduate programs. If the problem is simple lack of manpower, volunteering to do it myself might be a great first step to improving the OMSCS advising situation.
2. Finding out why UOPeople shut down its AAVO forums. Knowing what problems they encountered would help to avoid the same pitfalls in any analogous system.
3. Reading many, many outstanding papers, most notably everything in Educational Recommender Systems and Technologies: Practices and Challenges, to see if there are any overlooked examples.
4. Revisiting the advisor requirements (<https://github.com/Arkaaito/omscs-advisor/blob/master/requirements.md>) in light of what I've learned over the last two weeks, and trimming them into something that can be completed in a single term.

(Informal) References

Note: the numbered references are very unimportant! They are here purely to serve as pointers to the commercial products and/or specific projects referenced. The more significant, research-focused references are listed below in standard style.

- [1] Ellucian Degree Works. Retrieved from <http://www.ellucian.com/Software/Ellucian-Degree-Works/>
- [2] u.achieve: A comprehensive degree audit and academic planning solution. Retrieved from <http://www.collegesource.com/products/u-achieve/>
- [3] Hobson's AgileGrad Datasheet. Retrieved from http://www.hobsons.com/uploads/hobsons_agilegrad_datasheet.pdf
- [4] PeopleSoft Campus Solutions: Innovating for the Global Campus. Retrieved from <http://www.oracle.com/us/products/applications/peoplesoft-enterprise/campus-solutions/overview/index.html>
- [5] Help students stay on track with CAMS Enterprise's Degree Audit software module. Retrieved from <https://www.threeriverssystems.com/software/cams-enterprise/degree-audit/>
- [6] Georgia Tech DegreeWorks. Retrieved from <http://www.degreeworks.gatech.edu/>.
- [7] Ellucian Degree Works Scribe Service. (2015). Retrieved from <http://www.ellucian.com/Solution-Sheets/Ellucian-Degree-Works---Scribe-Service/>
- [8] DegreeWorks Project Status. Retrieved from <http://www.degreeworks.gatech.edu/general/status.php>.
- [9] Hemphill, L., Hemphill, H., and Biswell, C. L. (2013). Example of an Online Graduate Advising Site: Lessons Learned. *Proceedings of the Association for*

Educational Communications and Technology 2013. Retrieved from

http://www.aect.org/pdf/proceedings13/2013/13_10.pdf

[10] Waldner, L., McDaniel, D., Esteves, T., & Anderson, T. (2012, Feb. 22). The eQuad: A Next-Generation eAdvising Tool to Build Community and Retain Students.

Penn State Mentor. Retrieved from <https://dus.psu.edu/mentor/2012/10/equad-eadvising-tool-build-community-retain-students/>

[11] University of the People Student Handbook [2013-14]. Retrieved from

http://uopeople.edu/files/Pdf/2013_uopeople_student_handbook.pdf.

[12] AAVO Update from Student Affairs. *UOPeople Students' Blog*. Retrieved from

<https://uopeopleb.wordpress.com/2015/03/12/aavo-update-from-student-affairs/>.

Geyer-Schulz, A., Hahsler, M., Jahn, M. (2000). Educational and Scientific Recommender Systems: Designing the Information Channels of the Virtual University. *International Journal of Engineering Education* Vol. 17 no. 2 pp. 153-163.

Retrieved from <http://www.ijee.ie/articles/Vol17-2/IJEE1209.pdf>.

Geyer-Schulz, A., Hahsler, M., Jahn, M. (2002). Recommendations for Virtual Universities from Observed User Behavior. *Classification, Automation, and New Media*, pp. 273-280.

Okoye, I., Maull, K., Foster, J., and Sumner, T (2012). Educational Recommendation in an Informal Intentional Learning System. *Educational Recommender Systems and Technologies: Practices and Challenges*.

Vialardi, C., Bravo, J., Shafti, L. Ortigosa, A. (2009). Recommendation in Higher Education using Data Mining Techniques. *Proceedings of the Second International Conference on Educational Data Mining*. Retrieved from <http://www.educationaldatamining.org/EDM2009/uploads/proceedings/vialardi.pdf>.