Recommender Systems

Course Syllabus for CS 4379Q/5369Q (Spring 2016)

Overview

This course covers the basic concepts of recommender systems, including personalization algorithms, evaluation tools, and user experiences. We will discuss how recommender systems are deployed in e-commerce sites, social networks, and many other online systems. We will also review current research in the field.

Course Logistics

Course Title	CS 4379Q: Introduction to Recommender Systems
	CS 5369Q: Recommender Systems
Credits	3
Schedule	Tuedays and Thursdays, 9:30–10:50 AM in Derrick 235
Prerequisites	CS 3358 (Data Structures) with a C or higher
Readings	*Introduction to Recommender Systems* (online materials)
	Recommender Systems: An Introduction (optional textbook)
	Collaborative Filtering Recommender Systems
	Additional research papers and online articles
Web site	TRACS
Software	LensKit, Java

Instructor

Michael Ekstrand

Office	Comal 307F
E-mail	ekstrand@txstate.edu
Phone	(512) 245-7523
Skype	mdekstrand
Office Hours	MW 11AM-noon, Tu 1-2PM, Th 1-3PM
	Other times by appointment (in-person or remotely)

RESOURCES AND READINGS

Primary Material and Textbook

Our primary source material will be the videos and readings in *Introduction to Recommender Systems*, a free online course on Coursera. Think of this as an electronic video textbook.

Coursera will nag you to purchase a Verified Certificate; that is **not required** for this class. You do not need to pay anything to make use of the material, you only need to register a Coursera account. We also will not be relying on the online assignments or exams; you may still find them useful for your own study.

If you wish to purchase a printed textbook, the (optional) traditional text for the course is the following (available electronically via the library):

> *Recommender Systems: An Introduction* by Jannach, Zanker, Felfernig, and Friedrich (Cambridge, ISBN 978-0-52-149336-9)

Finally, there is a survey article that you will also likely find useful to read:

> *Collaborative Filtering Recommender Systems* by Ekstrand, Riedl, and Konstan (now publishers; Foundations and Trends in Human-Computer Interaction **4**(2)). Available free from my web site.

Research Readings

Throughout the course, we will be reading and discussing a number of research papers. The currently-planned list of papers is in the appendix of this syllabus; links to each paper will be provided in TRACS.

Software Resources

This course contains several programming assignments which will be in Java using the LensKit framework. We will spend some time in class on Java, but if you do not yet know it, I recommend the following book:

> Niemeyer, Patrick, and Daniel Leuck. 2013. *Learning Java*. Fourth edition. Sebastopol, Calif: O'Reilly. ISBN 978-1-4493-1924-3.

It is available from the library in both physical and electronic forms. For further study on the principles of good Java application design and coding practices, I recommend the following:

Joshua Bloch. 2008. *Effective Java* 2nd ed., Upper Saddle River, NJ: Addison-Wesley. ISBN 978-0-201-31005-4.

In addition, you will likely need to consult documentation for LensKit and other Java libraries; further links to these resources will be provided in TRACS.

COURSE STRUCTURE

This course is built on the content of the *Introduction to Recommender Systems* course on Coursera, using that instead of a traditional textbook. You will need a Coursera account to access the online course materials. There is no charge for this account.

I have organized the class material into *modules*, following the structure of the Coursera material. We are compressing a couple of the early modules to leave more room at the end for finishing up research projects.

The online videos are the primary means of delivering core content. **You are responsible for watching all videos.** You are also encouraged to complete the assignments and exams on Coursera to practice and better understand the material, but they will not factor into your grade. In class, we will further discuss and clarify the materials in the videos and readings, but this discussion will be in reference to the video material.

Course Components

The work in this course falls into several categories:

- **Reading reports** ask you to briefly summarize a research paper from the readings.
- **Assignments** to give you experience building, testing, and analyzing recommender systems.
- A **research project** to understand, communicate, and extend recent recommender systems research.
- Midterm and final exams.

Your final grade will be computed from these components as follows:

Category	*4379Q %*	*5369Q %*
Reading Reports	5	5

Analysis Assignments	10	10
Programming Assignments	50	50
Exams	20	20
Research Project	15	15

The standard 70/80/90 scale determines the minimum grade you will receive (that is, if you have 80 total course points, you will receive at least a B).

Readings and Reports

The Coursera course has a number of research papers and other articles to read; I encourage you to read those papers as well. However, details only found in one of those papers will not appear on an exam unless I have specifically indicated that paper as a required reading in class or via communications on TRACS.

In addition, there is a selection of research papers in the detailed course schedule. Some of these overlap with the Coursera readings, so it is not an entirely new set of readings. We will discuss some of these readings in class.

These readings will also form the basis of the Reading Reports. Each reading has an associated Reading Report assignment. For the reading report, write 2–3 paragraphs briefly addressing the following:

- What question is the paper trying to answer, or what problem is it trying to solve?
- What is the key idea of the solution or experimental method?
- What are the main lessons from the paper?

In addition to the assigned course readings, I expect graduate students to find and read some additional research papers. Graduate students must submit **12 reading reports**: 10 from the assigned readings, and 2 from additional research papers.

Undergraduate students need to submit **8 reading reports** on assigned readings.

Do not neglect the reading reports. The research project will be *much* easier if you are faithful in doing the readings and their reports.

I also encourage you to discuss the readings with your fellow students in advance of class; you may even do this before submitting the reading reports.

Analysis Assignments

There will be two analysis assignments, in which you will either analyze existing recommender services (the first) or develop a design brief for for a new service (the second). You may work in groups of up to 3 on these assignments.

Programming Assignments

Each module, except the first and last, will have a programming assignment (6 assignments). The assignment will be available at or before the date we begin the module. The assignments will involve programming and/or testing recommender systems using the LensKit toolkit.

Each assignment is due at **9AM on Tuesday** of the beginning of the next module. The lowest assignment grade will be dropped.

You may work in groups of up to 2 on the assignments. If you work with a group, only submit the assignment under **one group member**, and list all members of your group in your assignment submission.

I suggest that you watch the videos early in the module so you have time to work on the assignment.

Research Project

The final component of this class is the research project. For graduate students there are two options:

- A survey and research proposal, surveying and summarizing at least 4 research papers and proposing a new experiment to extend their results or answer new questions they raise.
- A replication study, where you attempt to reproduce a recent recommender systems algorithm or experiment paper.

Undergraduate students may do one of the graduate student options if they wish. However, they also can complete the requirement by selecting a research paper that we have not discussed in class, writing an extended review of it (similar to the reading reports but more in-depth), and presenting it to the class.

You may work with a partner on this project. Each student or group will present on its research project at the end of the semester. More details will be available in the research project description.

COURSE POLICIES

Attendance

I encourage you to attend all class sessions, but ordinary lecture attendance will not directly affect your grade. You do need to be present for all exams as well as the presentation days.

Some lectures will be on Java programming to prepare you for the assignments; I will announce in advance those lectures, so that you can skip them if you already know Java. They will most likely be in weeks 2–4.

Late Work

For the **programming assignments**, you may resubmit assignments on a batch system. Each assignment will have a nominal due date, by which you must submit at least a partial solution. You may resubmit assignments 1–3 by the end of spring break, and 4–6 by the beginning of finals week (May 2nd). The later submission will actually be graded. If your only submission is before original due date, you will receive 1% extra credit for each on-time submission (for a total of 6% possible extra credit).

Analysis assignments must be submitted on time. There will be sufficient lead time on them that you can plan ahead and do them early, which I highly recommend. Further details are in the assignment descriptions.

For the **project deliverables**, each deliverable must be submitted on time. Deliverables will be accepted up to 24 hours late with a 25% grade penalty.

Late reading reports are not accepted, but only the required number of reading reports will affect your grade.

The mid-term and final exams will be at the published times. Makeup exams will only be given in exceptional circumstances.

Cheating and Academic Integrity

As both a programmer and a student, you are expected to do your own work, attribute sources, and respect the legal and moral rights of others with respect to their work; as a student, you are also required to abide by the University Honor Code. While I aim to allow you to make reasonable use of resources, cheating (including copying code, using unauthorized resources during tests, etc.) will not

be tolerated. If you are found to be cheating, the penalty may range from an F on the assignment to an F on the course, and will also be reported to the university.

External Resources

You may consult external resources such as other books and web sites for understanding how to solve homework problems or portions of the project. In your assignment solution, list all external resources you used; if they are available online, provide the URL. You do not need to cite the textbook, or the official documentation for the software we are using (Java, LensKit, and the libraries used by LensKit).

Besides the course forum, you may ask questions related to the course material and concepts required to complete the work on publicly accessible discussion forums such as Stack Overflow, newsgroups, or publicly-archived mailing lists. To qualify as publicly-accessible, a site must provide access to complete discussions without requiring payment or registration. Provide URLs to the forum discussion on the relevant web site or archive (Google Groups works well for newsgroup archives) with your project deliverable submission. When you ask a question for one of the assignments, mention that it is for a course project and that your instructor permits you to make reasonable use of discussion forums. You may also freely use the LensKit Gitter chat room.

Restrict your questions to questions about how to go about a particular sub-portion of the problem, how something works, why something you are trying doesn't work, or other specific difficulties. Do not ask "how do I solve <the problem description>?", or similarly direct translations of the project requirements, or for specific code. Questions should be written to fill in a gap in your understanding that will then enable you to continue your work, not to get a solution to the assignment.

Conduct

You are expected to behave in a civil, respectful manner in all class interactions, both in official meetings such as lectures and out-of-classroom activities such as project group meetings and study sessions, and to contribute to a constructive learning environment.

Texas State policy (PPS 4.02) describes general behaviors that are disruptive. In addition, the Hacker School Social Rules are a good source of guidance on how to maintain a constructive and educational environment.

If you experience or witness harassment of any form, please let me know.

Disability Accommodations

If you need particular accommodations to be able to fully participate in this course, please talk with me as soon as possible. I may ask that you provide documentation from the Office of Disability Services, so if you have such documentation please bring it.

SCHEDULE

Following is an approximate schedule. I may adjust it as we progress through the semester. **Bold** items are key dates for the project and exams.

Week	Date	Mod.	Topic	Assignment
1	1/18	1	Introduction	
2	1/25	2	Non-Personalized Recommendations	A1
3	2/1	3	Content-Based Recommendations	A2
4	2/8			
5	2/15	4	User-Based Collaborative Filtering	A3
6	2/22			
7	2/29	**5**	Evaluation	A4
8	3/7		**Midterm** (Thursday)	*Analysis 1*
	3/14		*Spring break*	
9	3/21	6	Item-Based Collaborative Filtering	A5
10	3/28			*Research Proposal*
11	4/4	7	Matrix Factorization	A6
12	4/11			
13	4/18	8	Advanced Topics	*Draft Report*
14	4/25		**Research Presentations**	*Final Report*
F1	5/2		*Study & Finish Research Writing*	*Analysis 2*
F2	5/9		**Final Exam** (Tue, May 10 8–10:30 AM)	

Assignments (A*n*) are listed in the day they are issued; they are typically due at the beginning of the *following* module. Other deliverables (in *italics*) are listed the week they are due (they are due at the end of the week).

APPENDIX A: RESEARCH READINGS

The list of readings is also available as a Zotero collection. You may need to be on campus to access full papers from the links below; links that work off-campus will be available on TRACS.

Week 1

Hill, Will, Larry Stead, Mark Rosenstein, and George Furnas. 1995. "Recommending and Evaluating Choices in a Virtual Community of Use." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 194–201. CHI '95. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co. doi:10.1145/223904.223929.

Week 2

Amatriain, Xavier, Josep M. Pujol, and Nuria Oliver. 2009. "I Like It... I Like It Not: Evaluating User Ratings Noise in Recommender Systems." In *User Modeling, Adaptation, and Personalization*, edited by Geert-Jan Houben, Gord McCalla, Fabio Pianesi, and Massimo Zancanaro, 247–58. Lecture Notes in Computer Science 5535. Springer Berlin Heidelberg. doi:10.1007/978-3-642-02247-0_24.

Cosley, Dan, Shyong K. Lam, Istvan Albert, Joseph A. Konstan, and John Riedl. 2003. "Is Seeing Believing?: How Recommender System Interfaces Affect Users' Opinions." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 585–92. CHI '03. New York, NY, USA: ACM. doi:10.1145/642611.642713.

Kluver, Daniel, Tien T. Nguyen, Michael Ekstrand, Shilad Sen, and John Riedl. 2012. "How Many Bits Per Rating?" In *Proceedings of the Sixth ACM Conference on Recommender Systems*, 99–106. RecSys '12. New York, NY, USA: ACM. doi:10.1145/2365952.2365974.

Week 3

Phelan, Owen, Kevin McCarthy, Mike Bennett, and Barry Smyth. 2011. "Terms of a Feather: Content-Based News Recommendation and Discovery Using Twitter." In *Advances in Information Retrieval*, edited by Paul Clough, Colum Foley, Cathal Gurrin, Gareth J. F. Jones, Wessel Kraaij, Hyowon Lee, and Vanessa Mudoch, 448–59. Lecture Notes in

Computer Science 6611. Springer Berlin Heidelberg. doi:10.1007/978-3-642-20161-5_44.

Week 4

Pera, Maria Soledad, and Yiu-Kai Ng. 2014. "Automating Readers' Advisory to Make Book Recommendations for K-12 Readers." In *Proceedings of the 8th ACM Conference on Recommender Systems*, 9–16. RecSys '14. New York, NY, USA: ACM. doi:10.1145/2645710.2645721.

Week 5

Resnick, Paul, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. "GroupLens: An Open Architecture for Collaborative Filtering of Netnews." In *ACM CSCW '94*, 175–86. ACM. doi:10.1145/192844.192905.

Week 6

Golbeck, Jennifer. 2006. "Generating Predictive Movie Recommendations from Trust in Social Networks." In *Trust Management*, edited by Ketil Stølen, William H. Winsborough, Fabio Martinelli, and Fabio Massacci, 93–104. Lecture Notes in Computer Science 3986. Springer Berlin Heidelberg. doi:10.1007/11755593_8.

Week 7

Said, Alan, Ben Fields, Brijnesh J. Jain, and Sahin Albayrak. 2013. "User-Centric Evaluation of a K-Furthest Neighbor Collaborative Filtering Recommender Algorithm." In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, 1399–1408. CSCW '13. New York, NY, USA: ACM. doi:10.1145/2441776.2441933.

Week 8

Bellogin, Alejandro, Pablo Castells, and Ivan Cantador. 2011. "Precision-Oriented Evaluation of Recommender Systems: An Algorithmic Comparison." In *Proceedings of the Fifth ACM Conference on Recommender Systems*, 333–36. RecSys '11. New York, NY, USA: ACM. doi:10.1145/2043932.2043996.

Week 9

Karypis, George. 2001. "Evaluation of Item-Based Top-N Recommendation Algorithms." In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, 247–54. CIKM '01. New York, NY, USA: ACM. doi:10.1145/502585.502627.

Week 10

Bollen, Dirk, Bart P. Knijnenburg, Martijn C. Willemsen, and Mark Graus. 2010. "Understanding Choice Overload in Recommender Systems." In *Proceedings of the Fourth ACM Conference on Recommender Systems*, 63–70. RecSys '10. New York, NY, USA: ACM. doi:10.1145/1864708.1864724.

Week 11

Linden, G., B. Smith, and J. York. 2003. "Amazon.com Recommendations: Item-to-Item Collaborative Filtering." *IEEE Internet Computing* 7 (1): 76–80. doi:10.1109/MIC.2003.1167344.

Week 12

Burke, Robin. 2002. "Hybrid Recommender Systems: Survey and Experiments." *User Modeling and User-Adapted Interaction* 12 (4): 331–70. doi:10.1023/A:1021240730564.

Week 13

Koren, Yehuda. 2008. "Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model." In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 426–34. KDD '08. New York, NY, USA: ACM. doi:10.1145/1401890.1401944.

Week 14

Hariri, Negar, Bamshad Mobasher, and Robin Burke. 2014. "Context Adaptation in Interactive Recommender Systems." In *Proceedings of the 8th ACM Conference on Recommender Systems*, 41–48. RecSys '14. New York, NY, USA: ACM. doi:10.1145/2645710.2645753.