1-2: Welcome to the Course!

Goals for Today

- To understand the scope of the course.
- To understand the two tracks and the prerequisites for each.
- To understand the dual MOOC/U of M course structure and its implications
- To review and understand basic course concepts including assignments, exams, expectations, grading, academic standards, communication and feedback, and more …

Introductions

- Joseph A. Konstan
- Michael Ekstrand

Dedication

- John Riedl
  - Pioneer in Recommender Systems

Course Scope

- Broad Overview of Recommendation Techniques
  - Focused around algorithms
    - Non-personalized, content-based, collaborative
    - Emphasis on personalized, collaborative
  - Added topics/enrichment around interaction and design or recommender systems
  - Goal is to provide solid algorithmic core with extensive awareness of related topics
  - We include interviews with many leaders in field

Programming Challenge I

- Understanding recommender systems is useful even to people who can't program them.
- Approach: two tracks
  - Concepts track: expected to complete non-programming assignments, exams
  - Programming track: expected to have significant Java programming skill

Recommended Background …

- All Students
  - College-level algebra
  - Basic computing concepts and skills
- Programming Track
  - Java programming (extensive)
  - Data structures
  - Ability to install/manage open source software tools and libraries

Programming Challenge II

- Building quality recommender algorithms is hard – and often the details are best re-used rather than re-created
- Focus on using LensKit toolkit (and extending/adapting where possible) as a way to explore more algorithms with less programming effort
  - Big benefit: can then use LensKit afterwards (open source Recommender toolkit)

Mix of Coursera and Credit Students

- This course is being taken simultaneously by U of M undergrad/graduate students
  - U of M students all must be in the programming track
  - U of M students will participate in all online activities (including online grading)
- U of M students will have regular face-to-face sessions
  - *Question-driven sessions will be recorded and posted for Coursera students to view (optional)*

Interaction … the Class Forums

- There are a lot of you (almost 8000 one month before launch!)
- We will not be taking questions directly – all questions must come via the class forums
  - Organized by topic (technical, programming, course modules, assignments)
  - Be sure to vote up questions you feel most deserve answers
  - We will post replies to top vote-getters at least twice a week

Course Requirements and Grading

- Grades based on assignments and exams
- Two exams (multiple choice/short answer)
- “Written” and “Programming” assignments
  - Six pairs of assignments (one per two-week module), plus “assignment zero” this week
  - Variety of grading techniques (automatic, peer)
  - Get an early start (final deadlines are real)

Assignment grading

- We are grading results, not process
- You are generally asked to submit a file (usually csv format) and we'll provide a sample or specifications
- Some programming assignments come with code to submit the file themselves
- For many assignments, you'll get a personal set of test cases to submit

Written Assignment Zero …

- Use the following website to rate a set of movies … this will build a class dataset for use in our later assignments.

  http://mooc.grouplens.org/ratemovies/

- When you're done, you'll receive a code to enter into Coursera – it's that easy!
- You can use the code (or add your own identifier) to follow your profile through the course.

Statements of Accomplishment

- Standard: 50% of possible points (approx. 80% of non-programming points)
- Distinguished: 80% of possible points (requires completing programming assts.)

Note: U of M registered students will get regular grades

Academic Standards

- Academic integrity is essential
- Honor code online
  - U of M academic standards for credit students
- All written assignments and exams must be your own work
- Programming assignments may be completed in pairs, but each student must generate own output file; collaborators must be identified as part of online submission

Feedback and Surveys

- We're all still learning, and we will be studying how this course goes both to make mid-course corrections (where possible) and to shape future offerings
- Please participate in surveys and provide feedback
  - We have partnered with education researchers to analyze data and learn from the experience
- We will also have a "general suggestions" topic in the class forum; vote up good ideas!

Final Thoughts

- We’re glad you’re here
- Own your educational experience
  - No stigma associated with view-only
- But commit enough time to get value
  - Programming track will take 10+ hours a week for many students

1-2: Welcome to the Course!