

COMP3208 Social Computing Techniques

Stuart E. Middleton, sem03@soton.ac.uk

Updated: 27th January 2021

Deliverables and deadlines

Deliverable	Deadline	Feedback	Marking Scheme	Weight
submission.csv	Optional preliminary submissions timetable week 25-30 Mandatory final submission timetable week 30 Last submission is the final one	Automated email of MSE score	The MSE score of the benchmark code will be made available to students. The MSE score will be compared to several different benchmark algorithm scores and then a mark awarded. These benchmarks and the scores they produce are hidden.	15 marks (37.5%)
sourcecode.zip	Mandatory final submission timetable week 32	None	Source code used to compute the final submission.csv No marks are awarded for the code, but it can be used to check for academic integrity issues	0 marks (0%)
report.pdf	Mandatory final submission timetable week 32	4 weeks after submission	Design and description of recommender system 10 marks. Evaluation and discussion of recommender system 15 marks	25 marks (62.5%)

Task

The aim of the coursework is to build and evaluate a recommender system. The coursework should be done in pairs or (exceptionally) in groups of 3. You will be given a training set which consists of a set of users and items, and user ratings for these items. In addition, you will be given a test set with only users and items for which you need to predict the user's ratings.

The recommender system can be implemented in any language you choose as a group. You are required to implement your own recommender system. However, you can make use of some libraries such as Math libraries and code snippets, but you will need to acknowledge these in your report (see below). If in doubt, ask the module leader. Failing to comply with these requirements will be considered plagiarism and could result in failing the coursework or even module, and university procedures will be applied.

Deliverables (one of each per team)

(OPTIONAL) Multiple preliminary submissions.csv files can be submitted with preliminary predicted ratings from the test set. Submission of the preliminary ratings is optional, but you are strongly encouraged to submit them. Your submitted ratings will be automatically compared to the true ratings, and you will be emailed the mean squared error (MSE). This will give you feedback about your results and an indication of how far your current ratings are from the true ratings, and whether you need to improve them. Up to 10 submissions in total (including the final one) are allowed.

(MANDATORY) A submissions.csv file with the final predicted ratings from the test set. The mark for the final ratings will be assessed based on the mean squared error (MSE) compared to the true ratings (same measure used for the preliminary predicted ratings). Your last submission.csv file will be taken as your final one and used to compute your final mark.

(MANDATORY) A sourcecode.zip file with the source code used to produce the final submission.csv file. This can be used to check for academic integrity issues.

(MANDATORY) A report.pdf where you explain how your recommender system works (i.e. the algorithms used), including how you deal with any cold start problems. The report should be written in the style of a journal or conference paper and provide sufficient details to allow someone else to reproduce your results, without providing the actual implementation. You should use a formal approach to explain your algorithm(s) including the use of pseudocode and equations, in the same style of a research paper. In addition, you should evaluate your recommender system by (1) using appropriate metrics, (2) tuning your system by trying out different parameter settings such as neighbourhood size and/or variations of the recommender algorithm, (3) comparing it to various simple benchmarks at very least, and (4) discussing how it could be improved. The evaluation should be detailed in the report and it is expected that the report contains several graphs and/or tables with results. For teams of 3 people, the students should implement a second recommender system and compare the results with the one used for the submitted ratings. The report should contain references to papers, material and/or packages/libraries/software used to support your system. The group report should be no more than 10 pages including title, abstract, introduction, conclusions, references, appendices, etc.

Notes and Restrictions

You are not allowed to use existing implementations for your main recommender system. The recommender system you use to submit your ratings needs to be your own code. While you can use code snippets from e.g. StackOverflow, these must be clearly marked in the code itself and acknowledged in the report. You can use standard libraries such as NumPy and other math libraries, but not libraries that already contain recommender system implementations. If in doubt, ask the module leader at one of the coursework surgery sessions (not by email). Any use of libraries should also be clearly acknowledged in the report. You can use existing code as a benchmark to compare your system against as part of the evaluation of the report. Again, this needs to be clearly acknowledged.

Make sure that you provide a prediction for each of the rows in the test set. Failure to provide a complete set will result in a deduction of marks and the missing data will be set to zero by default. Also, explain in your report how you deal with possible missing values (i.e. item and/or user cold start problem).

Given the size of the dataset, we strongly recommend that you use a database. A simple database such as SQLite suffices. Example java code for using a database is provided to help, but it is not a requirement to use it.

Datasets

Two different datasets will be provided: a 'small' one with about 10M ratings and 20k items, and a large one with about 26M ratings and 54k items. As a guide, when submitting the results for the small dataset, a maximum of 70% of the total mark can be achieved for the predicted rating component of the coursework.

Teams and marking split

This coursework is submitted in teams of 2 or 3 students. You need to decide for yourself what team you will be in. Email sem03@soton.ac.uk with your team choices by timetable week 24 (12th March 2021) to register your team. Students who fail to register a team by this deadline will be assigned a random team pairing by lecturers.

Normally it is expected that individuals within a team contribute equally overall, even if the tasks may be different (e.g. implementation vs report writing). The marks for a team will therefore be split evenly.

However, in the case that there is a clear discrepancy in overall effort, teams can choose to split the mark unevenly and so have different individual marks, as long as (1) the average mark for the team as a whole remains the same, (2) a non-failing mark cannot become a failing mark, (3) individual marks for the coursework do not exceed 100%, and (4) all team members agree on the mark split. A short motivating statement needs to be included. To request a non-equal mark distribution, students should send a single email to the module leader with the proposed split and CC all students in the team. This should be done within a week of receiving the report feedback. The module leader retains the right to adjust or ignore the proposed split if it is deemed inappropriate.

Learning Outcomes

Use recommender technologies such as item-based and user-based collaborative filtering techniques

Describe the most important techniques and issues in designing, building and modelling social computing systems.

Set up social computing experiments and analyse the results using a scientific approach

Late submissions

Late submissions will be penalised according to the standard rules.

The handin submission time is based on your last submission, so if you submit after the deadline you will incur a late penalty.

Plagiarism

Your submitted code and report need to be your groups own work. In case of doubt, feel free to ask! This is important as any violations, deliberate or otherwise, will be reported to the Academic Integrity Officer.