CS6220 Unsupervised Data Mining

HW6 Social Graphs, Recommendation Systems

Make sure you check the syllabus for the due date. Please use the notations adopted in class, even if the problem is stated in the book using a different notation.

We are not looking for very long answers (if you find yourself writing more than one or two pages of typed text per problem, you are probably on the wrong track). Try to be concise; also keep in mind that good ideas and explanations matter more than exact details.

Submit all code files Dropbox (create folder HW1 or similar name). Results can be pdf or txt files, including plots/tabels if any.

"Paper" exercises: submit using Dropbox as pdf, either typed or scanned handwritten.

DATATSET MovieLens 100K Ratings https://grouplens.org/datasets/movielens/100k/

DATATSET Netflix Prize ratings Dataset https://www.kaggle.com/netflix-inc/netflix-prize-data

DATATSET Friendster Social Graph http://socialcomputing.asu.edu/datasets/Friendster

DATATSET Flicker Social Graph http://socialcomputing.asu.edu/datasets/Flickr, but use the one curated in DM resources

PROBLEM 1: Recommender System using Collaborative Filtering

Implement a Movie Recommendation System and run it on the Movie Lens Dataset (Train vs Test). Mesure performance on test set using RMSE

- First you are required to compute first a user-user similarity based on ratings and movies in common
- Second, make rating predictions on the test set followoing the KNN idea: a prediction (user, movie) is the weighted average of other users' rating for the movie, weighted by user-similarity to the given user.

PROBLEM 2: EXTRA CREDIT Netflix Recommendations

Implement a recommender system on the Netflix Prize Dataset. Use the "probe" set for testing. For a competitive systems, one need to add movie content features such as actors, genres, directors, music, etc. These features are not available from Netflix, but for some movies they have been crawled by Movie Title form other websites such as IMDB (for example "movie_details.xml" file in DM_resources, but you can get more such features on your own.)

PROBLEM 3: Social Community Detection

Implement a community detection algorithm on the Flicker Graph. Use the betweenes idea on edges and the Girvan–Newman Algorithm. The original dataset graph has more than 5M edges; in DM_resources there are 4 different sub-sampled graphs with edge counts from 2K to 600K; you can use these if the original is too big. You should use a library to support graph operations (edges, vertices, paths, degrees, etc). We used igraph in python which also have builtin community detection algorithms (not allowed); these are useful as a way to evaluate communities you obtain