Predicting What Books You Will Like

Chris Barnett - cjbarnet@ucsc.edu, class 142

I would like to use a collaborative filtering approach to learn what books someone is likely to appreciate based on what books they report liking in the past. The output of my algorithm will be a user-specific score over all books based on the that user's scores of one or more other books. In order to generate recommendations for a user U, all books U has not reported reading would be ranked by their score with respect to U.

I intend to use the Book Crossing Dataset for this project, which I found at http://www.informatik.uni-freiburg.de/~cziegler/BX/.

I plan to compare some existing techniques for collaborative filtering with an approach to user-based collaborative filtering that uses a variation of cosine similarity between users that I think will be more scalable and amenable to sparse data. Instead of taking the cosine between the preference vectors of two users to judge their similarity, I propose using just the dot-product. This technique has the advantage of only requiring calculations over the books that have been mutually rated by both users, however, it means that the resulting scores will not be normalized. The score of a book B with respect to user U would be given by the average rating of B weighted by the similarity of the scorer to U. Note that I am using the word "rating" to indicate a rating from 1 and 10 that an individual user assigns to a specific book and I am using the word "score" to identify the user-specific value the algorithm will assign to a book after aggregating many similar user's ratings. A score will not correspond directly to a rating from 1 to 10, however it should still provide a good prediction of the order of a user's preferences over books.

In order to evaluate the effectiveness of my algorithm, I propose to use a rank correlation algorithm such as Kendall's tau coefficient to measure how well my algorithm predicts how a user will rank the second half of their rated books after using the (randomly selected) first half as input. To clarify, I will partition the data by user into about 80% for training and 20% for testing. For each user in the testing block, I will randomly split their set of rated books into two groups: A and B. Using group A as input to my algorithm, I will use it to generate a score for each book in group B. I will rank the books in group B by both score (generated by my algorithm) and rating (specified by the user) and measure the correlation of the two rankings using Kendall's tau coefficient. I will then find the average rank correlation over all users in the training data to come up with an overall measure of success from -1 to 1 in terms of correlation between my algorithm's predictions and the training data.

Finally, I plan to compare how well my algorithm performs at ranking a user's rated books with other collaborative filtering algorithms.

The justification for this overall approach is based on the assumption that the most important aspect of a recommendation system is to prioritize items for a user according to their preferences. The actual number assigned to each item does not necessarily need to be meaningful on its own. Thus I propose that trying to precisely predict the rating that a user will give an item in an arbitrary range (such as from 1 to 10) is not solving the right problem. Instead I suggest that it is more worthwhile to predict the relative value a user will place on different items in a collection so that they can be appropriately prioritized for him.