Assignment 1

Predictions relating to book reviews on Goodreads

By Amogha Mohan Sekhar, A53301791

Task 1

Improvement of Baseline solution in HW3

High-Level Description:

Apart from just the popularity metric, a similarity metric was also introduced during my homework 3 solution. Two similarity metrics were introduced namely, Jaccard Similarity and Cosine Similarity. These worked well when combined with the popularity metric. Logistic regression was further built on this model to improve the accuracy.

Details of implementation:

Firstly, improving the popularity metric was essential. In my solution, a popularity metric of 63%, that is, a set of the most popular books (return1) that account for at least 63% of the interactions in the training data, was built. An addition to my solution was another set (top_books) of extremely popular books which accounted for 80% of the interactions in the training data. The idea behind building a second set of most-popular books was that my model confidently predicts a book would be read by the user if it belongs to top_books regardless of other factors. This is analogous to the real world where an individual would read an extremely popular book.

Secondly, when building a model for prediction based on item-item and user-user similarity, Jaccard Similarity and Cosine Similarity was implemented.

The Jaccard similarity worked well when combined with the popularity metric. The implementation of it was as follows: for a particular user u and book b, we would build a set user_books which consists of all the books u has read. For each of these books bi, the similarity was calculated with b. That is, |Number of users who have read bi and b|/|Number of users who have read b or bi|. Specifically, if this similarity>0.0405, my model predicts the book would be read by the user regardless of other factors. When combined with a popularity metric, specifically if a book belonged to return1, then the Jaccard similarity would have to be at least 0.01 to be predicted as would be read.

Thirdly, a logistic regression model was implemented on top of the model described above with C=1 and class weight as balanced. Logistic regression performed poorly for the read task if considered alone, however, when combined with Cosine Similarity, specifically having a value of at least 0.0045, and being predicted as true, we would predict that a user would read the book.

As such my model implementation for the read task is a set of if-elif conditions to determine if a particular user u would read a book b.

Things which did not work for my model:

I implemented Pearson Correlation but this did not improve the model above. I also implemented K-Nearest Neighbors to generate a new feature but this again did not prove fruitful.

Accuracy

The model produces an accuracy of 70.728%.

Task 2

Improvement of Baseline solution in HW3

High-Level Description:

I implemented the complete latent factor model to predict the item rating by a user. I fit the a model for:

$$rating(user, book) = \alpha + \beta user + \beta item + \gamma user * \gamma item$$

This was implemented with regularization. To reduce the Mean Squared Error on this model, lambda was initialized to be 2, which resulted in the lowest MSE of 1.11171, which beat the strong baseline.

Details of implementation:

I tried several different values of lambda and performed different kinds of random initializations using different seeds to finally end up with lambda= 2 as the best threshold. The code I implemented, considered a latent factor of 5 and trained it for over 10 epochs.

Things which did not work for my model:

I implemented the partial latent factor model, but it’s MSE was far higher compared to the complete latent factor model.

MSE

The MSE of the model was 1.13686