CSE258 Assignment1 - Read & Rating Prediction

Shihan Ran
M.S. in Computer Science
Kaggle User Name: shihanran / Display Name: Shihan Ran
Student ID: A53313589
sran@ucsd.edu

1 Read Prediction

1.1 Adjusting Jaccard Feature

Previously, our Jaccard similarity was computed as follows: Given a pair (u, b) in the validation set, consider all training items b' that user u has read. For each, compute the Jaccard similarity between b and b', i.e., users (in the training set) who have read b and users who have read b'. Predict as 'read' if the maximum of these Jaccard similarities exceeds a threshold.

However, this feature is too sparse. We adapted this to: Given a pair (u, b) in the validation set, consider all training items b' that user u has read. For each b', find all users u' who has read the book b'. Union all users u' together as U. For b, again, find all users u' who read the book b as U'. Compute the Jaccard similarity between U and U'. Predict as 'read' if the maximum of these Jaccard similarities exceeds a threshold. By doing this, we reduce the sparsity of the feature and improve the results.

1.2 Adjusting Popularity Feature

Previously, our popularity is defined as follows: Using a threshold of the Nth percentile of popularity. We could also define popularity as the number of how many people read this book.

1.3 Paying Attention to The Distribution

After trying much feature engineering, we found out that the maximum score we can achieve by designing good features is still under 70%. To get beyond that, we most likely have to think about other properties of the dataset and try out some models aimed at dealing with sparse datasets.

One thing we should pay particular attention to is that the dataset is pretty much balanced in the sense that for every user/book pair there is a negative book (a book that the user did not read) with the same user. Hence, each user shows up an even number of times in the test set and exactly 50% of the labels in the test set are positive.

Our method is given a user, and all the n books we want to predict if they are read or not. We can compute the Jaccard score of all the n pairs (u, b). After getting that, we rank the Jaccard scores and predict true for the first 50th percentage of pairs, false for the other. This guarantees that our result is balanced and much closer to the ground truth.

1.4 Adding Validation Set to Training

After training and fine-tuned all the parameters on the validation set, we re-train the model on the **whole** dataset using the best parameters we picked and outputs the prediction file directly. By doing this, we "increase" our training data and get better results.

2 Rating Prediction

2.1 Parameter Finetuning

- For Latent Factor Model, we can use the individual lambda parameters for user and item and finetune on them separately.
- Different methods of initialization.
- Add Gamma to Latent Factor Model and finetune on K.

2.2 Truncate Out-of-Range Values

For predicted values over 5 or below 0, we truncate them to 5 and 0 since no rating will go over 5 and below 0. This will definitely reduce our MSE.

2.3 Using Round

In a previous midterm exam (CSE255 fall2015), one of the problems (Problem 7) has proved that when we are trying to predict star ratings. Rounding the output to the nearest integer will decrease the MSE. Hence, we use this property and round our prediction to the nearest integer when they are close enough.

2.4 Adding Validation Set to Training

Again, after training and fine-tuned all the parameters on the validation set, we re-train the model on the **whole** dataset using the best parameters we picked and outputs the prediction file directly.