Recommendation system using Goodreads data

Yi Li(liy31) Tianshu Chu(tc2992)

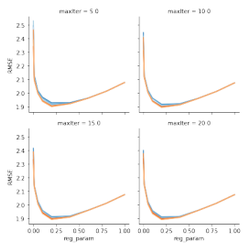
May 12, 2020

1 Introduction

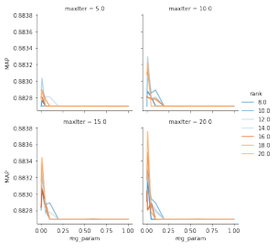
We have 228648342 ratings from 876145 users on 2360650 books. The rating is integers from 0 to 5 with mean of 1.80 and SD of 2.07. First, we split data by splitting the users into 60%,20%,20% into training, validation and test set. Then we split read books for each user in validation and test set into halves and merge half into training set. And we dropped those users not appeared in training set due to uneven split. Finally, we have 868319 users in training set, 167529 and 166776 in validation and test set respectively. The similar methods are used to generate 3 subset samples of entire data including 1%, 5%, 25% users respectively. We report the performance of baseline models as well as extended models on each sample for comparison.

2 Baseline Model

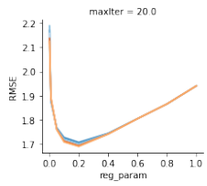
At baseline, we used all interactions to build ALS models. For the evaluation, we chose the top 500 recommended books for each user based on the predicted scores. The hyperparameters we tuned are: 1) max iteration:[5,10,15,20]; 2)rank:[8, 10, 12, 14, 16, 18, 20]; 3)regularization parameter:[0.001, 0.01, 0.05, 0.1, 0.2,0.4,0.6,0.8,1.0]. In the evaluation, we choose RMSE as our metrics, but we would report MAP in the meantime. Although the change of MAP during parameter tuning is quite unconspicuous. As we can see from the 1% data plots, higher rank renders better results both in terms of RMSE and MAP. While regularization term around 0.1, 0.2 generates better results under all situations. As ALS model typically converges to a reasonable solution in 20 iterations or less, we choose max iteration = 20 for further parameter tuning on larger data and extensions.



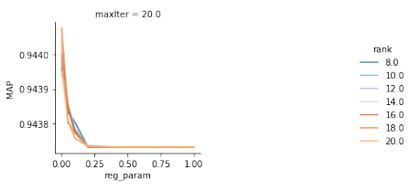
(a) RMSE on 1% data



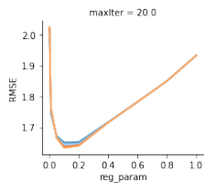
(b) MAP on 1% data



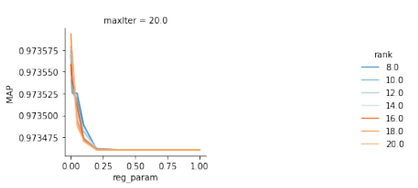
(c) RMSE on 5% data



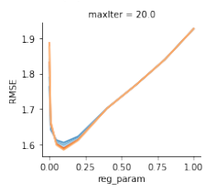
(d) MAP on 5% data



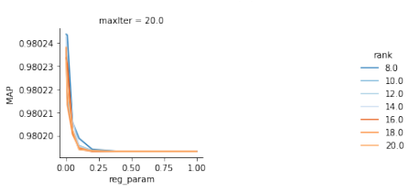
(e) RMSE on 25% data



(f) MAP on 25% data



(g) RMSE on 100% data



(h) MAP on 100% data

Figure 1: Parameter tuning on validation set for baseline model

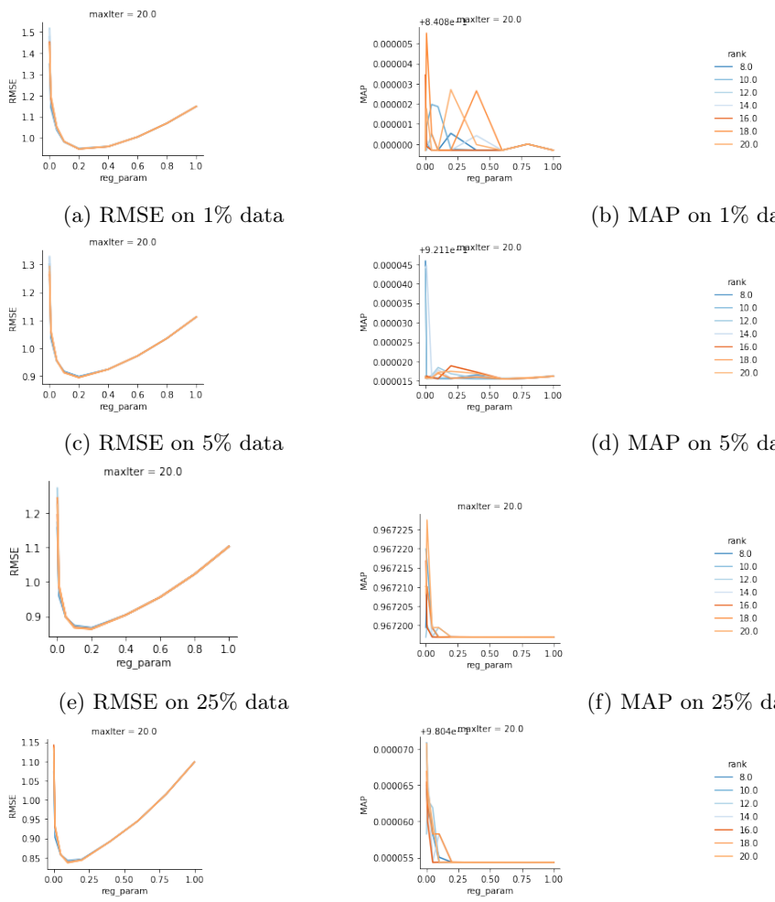
3 Extentions

3.1 Read or not

As we can see from Table 1, there are about 51% interactions indicating unread books and among those 0-rating books, 93.9% are unread. Actually, all unread books were given rating 0 in the data set. They are different from those books that were read but rated as 0. Therefore, we choose to build two different ALS models for read books and unread books. Then we merge the predictions together to obtain ranked list. The evaluation performance of read/unread model on validation set improved RMSE significantly (about 50% reduction). Although the MAP decreased a little bit, the difference is minimizing as sample size increases.

rating	**0**	**1**	**2**	**3**	**4**	**5**
is_read = 1	7579654	2050529	6189946	23307457	37497451	35506166
is_read = 0	116517139	0	0	0	0	0

Table 1: Rating by read or not



(a) RMSE on 1% data

(b) MAP on 1% data

(c) RMSE on 5% data

(d) MAP on 5% data

(e) RMSE on 25% data

(f) MAP on 25% data

(g) RMSE on 100% data

(h) MAP on 100% data

Figure 2: Parameter tuning on validation set for read/unread model

3.2 Review data

We can see from Table 2 that the rating for reviewed books could vary from 5 to 0. It is difficult to get any valid indication on rating score merely based on whether the book has been reviewed or not. Therefore we decided to build an external NLP model with more information to help predict ratings.

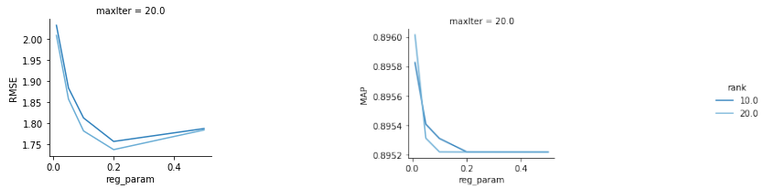
rating	**0**	**1**	**2**	**3**	**4**	**5**
is_reviewed = 1	573902	466301	1139874	3202940	5390833	5445299
is_reviewed = 0	123522891	1584228	5050072	20104517	32106618	30060867

Table 2: Rating by review or not

We split the review data into train/val/test sets by mapping each "user-book" instance to the train/val/test sets of original interaction data. After filtering out punctuation and stop words, we represented each review in the dataset as a vector of TF-IDF values. Then we fitted the train review vectors into a linear regression to get the prediction of rating scores for each instance in validation and test set. Finally, for all the instances that have been reviewed, we used the outcome from the linear regression as the final prediction score; for all

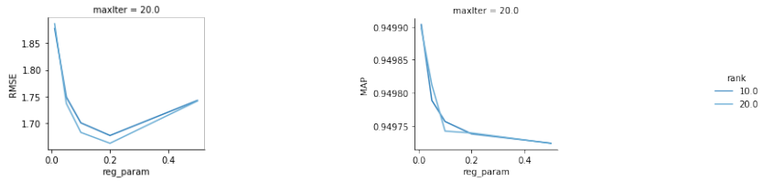
the other recommendations generated by the baseline ALS model, we kept their original outcomes. The top 500 books for each user from the combined prediction were used for evaluation.

For the linear regression, we tuned the following hyper-parameter: 1)regParam: [0.01,0.05,0.1,0.2,0.3,0.4,0.5]; 2)elasticNetParam: [0.001,0.01,0.05,0.1,0.2]. The model with lowest RMSE on the review validation data was chosen to produce the review prediction score. For ALS model, we set max iteration to 20, and tuned: 1)regParam: [0.01, 0.05, 0.1, 0.2, 0.5]; 2)rank: [10, 20]. The best model was the one that generate lowest RMSE on the combined top 500 books for each user from validation set. This approach could achieve better performance on both RMSE and MAP compared with baseline model.



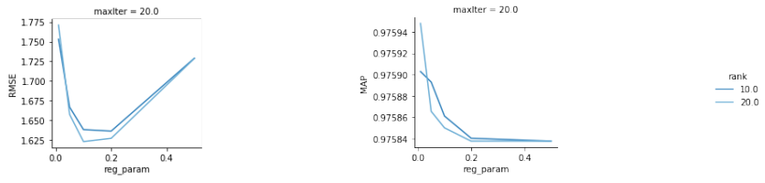
(a) RMSE on 1% data

(b) MAP on 1% data



(c) RMSE on 5% data

(d) MAP on 5% data



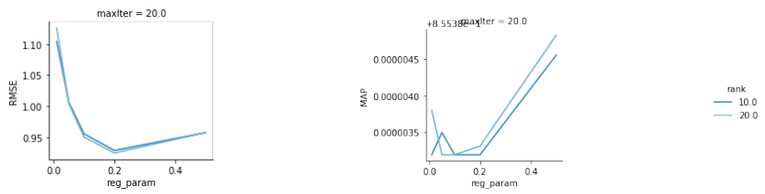
(e) RMSE on 25% data

(f) MAP on 25% data

Figure 3: Parameter tuning on validation set for review model

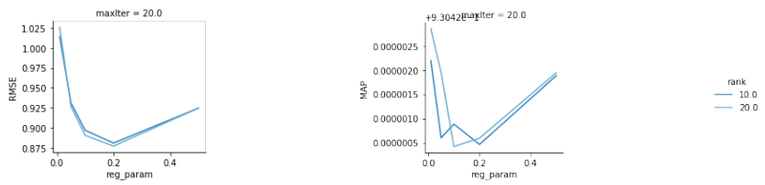
3.3 Combination of Review and Read

In this approach, we simply combined the above two extensions. We trained the ALS model for read books and unread books separately. Then, we merged the outcomes from two ALS model with the one from linear regression for reviewed books. The evaluation was made on the top 500 ranked list for each user from combined output, based on RMSE. The hyper-parameters are conducted on read books ALS model and reviewed books linear regression. The range are among the same as in section 3.2. The combination model reached the lowest RMSE among the others.



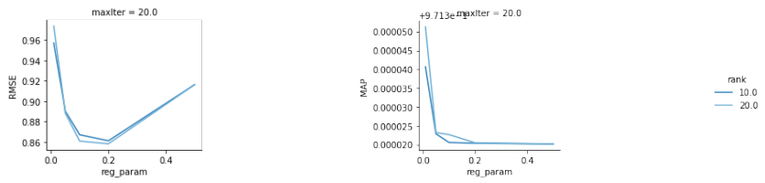
(a) RMSE on 1% data

(b) MAP on 1% data



(c) RMSE on 5% data

(d) MAP on 5% data



(e) RMSE on 25% data

(f) MAP on 25% data

Figure 4: Parameter tuning on validation set for read/unread+review model

4 Experimental Results

We chose models with best performance on validation set on each data sample and evaluate their performance on according test set. It seems to be clear that training read and unread books separately can largely lower RMSE but also decrease MAP. On the other hand, building the external NLP model for review data can help to perform better on both RMSE and MAP, but the improvements tend to be trivial as the sample size increases. In general, the approach of external NLP model solely could reach the highest MAP, while the combination of the above two methods could achieve the best RMSE. The observations are consistent across 1%, 5%, and 25% data samples. Some model results for full data are missing due to the cluster issue, but we can see a clear pattern given the existing output.

Best params	1%			5%			25%			100%		
	rank	reg	maxiter	rank	reg	maxiter	rank	reg	maxiter	rank	reg	maxiter
baseline	18	0.2	20	20	0.2	20	20	0.1	20	20	0.1	20
read/unread	20	0.2	20	18	0.2	20	20	0.2	20	20	0.1	20
review	20	0.2	20	20	0.2	20	20	0.1	20			
read/unread +review	20	0.2	20	20	0.2	20	20	0.2	20			

Table 3: Best parameters on the evaluation set selected by RMSE

RMSE(MAP)	**1%**	**5%**	**25%**	**100%**
baseline	1.8919(0.8827)	1.6900(0.9437)	1.6322(0.9735)	1.5853(0.9802)
read/unread	0.9475(0.8408)	0.8936(0.9211)	0.8623(0.9672)	0.8358(0.9805)
review	1.7379(0.8952)	1.6626(0.9497)	1.6232(0.9758)	
read/unread+review	0.9247(0.8553)	0.8772(0.9304)	0.8583(0.9713)	

Table 4: Performance of best model on the evaluation set

RMSE(MAP)	**1%**	**5%**	**25%**	**100%**
baseline	1.7254(0.8808)	1.6647(0.9437)	1.7025(0.9729)	1.6566(0.9802)
read/unread	0.9124(0.8397)	0.8844(0.9220)	0.8778(0.9670)	0.8300(0.9804)
review	1.7240(0.8942)	1.6668(0.9502)	1.6946(0.9751)	
read/unread+review	0.8852(0.8577)	0.8736(0.9321)	0.8638(0.9711)	

Table 5: Performance of best model on the test set

5 Conclusions

We can see from Table 4 and Table 5 that the combination of review and read model performed the best on RMSE across 1%, 5%, and 25% data samples. Furthermore, as sample size increases, the differences of MAP between models tend to be trivial. Thus, we chose the combination model as our best performed recommendation system.

Contributions

Baseline model, extension on read/unread books: Yi Li

Extension on reviewed books, NLP model and integrated model with read/unread books: Tianshu Chu

Report write-up: Yi Li, Tianshu Chu