Assignment1 DS4I

Gareth Edwards

09/11/2020

Introduction

The data given for this project is from a book rating database. This database contains ratings of 10000 users on 150 books. The task is to predict the scores of users, given previously read and rated books, using 4 different types of recommender systems. These systems are user-based collaborative filtering, item-based collaborative filtering, matrix factorisation and an ensamble method that combines the forementioned approaches.

User-based collaborative filtering bases predicted ratings on users that are similar to a given user. Item-based collaborative filtering is very similar to user-based but predicts ratings based on how similar other items are to the ones rated by the user. Matrix factorisation is a very different approach that decomposes users and items into latent factors. These latent factors then interact to produce a predicted rating for a user.

A common problem with item rated data (in this case book rating data) is its sparse nature. Many users only read a few books, and in the case of the given dataset, some users have read no books at all. The users that have read no books at all (it is assumed a rating of 0 means the book hasn't been read) are removed from the database. This results in a dataset with 7503 users who have read at least one book. On average each user has only read 2.44 books. This poses a problem in many recommender system algorithms as it is very difficult to tell what books users will like in the future based on ratings of only two books.

for the purposes of the matrix factorisation, the dataset has been split into a train and test set with a 70/20 split. For the purposes of computation time, the train set is used to test the accuracy on both the user-based, item based and ensamble rating prediction approaches. The split is based on users. This means that the training set consists of 70% of all the users, and their book ratings, who have read at least one book. The test set consists of the remaining users. Furthermore, each train and test set was converted to a wide format with users as rows, book ISBNs as columns and corresponding ratings in each cell. This wide approach makes calculations in the recommender systems easier.

User Based Collaborative Filtering

A K-nearest neighbours user-based approach was used. First a similarity matrix was created by finding the similarity between each user in the train set. The similarity measure used is the cosine similarity:

$$cos(\theta) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{||\boldsymbol{x}|| \; ||\boldsymbol{y}||} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}}$$

Where x and y refer to two different users rating vectors. The User-based system then measures the k closest users to a given user, averages the scores these k users have given to each book and assigns those values to the user in question. A problem arises when the k nearest users have all not read a certain book, and therefore not rated it. In this case the score of 0 is given to this book as we assume if similar users haven't read the book then the given user will also not want to read it. This is ofcourse not the best approach as all k users plus the new user may all have not read a book that they all might rate highly.

The value of k chosen for this approach is 15. Higher values include too many users that are only slightly similar to the new user and thus should have no say in the new user's ratings. Lower values leave the user with a lot of predicted ratings of 0 as it's more likely that a smaller group of users would have read less of the total books.

Root mean square error was used to calculate accuracy of the recommender system. Users from the train set were presented as input to the user based recommender system and the root mean square error between the predicted and the original rating was taken for each user. The average RMSE was then taken over all the users from the training set.

Item Based Collaborative filtering

The item-based approach is similar to the user-based approach. However, in this case the predicted ratings are based on the similarities between the books a user has read and the rest of the books in the database. In this approach a user's read books are extracted from the database. The similarity between the user's read books and the rest of the books in the database is calculated. For each book in the database, the overall similarity is the sum of the similarities between a book and the users read books. Once this is calculated we can see which books are most similar to the books read by the user.

To predict the score we assign a total of 1500 rating points (total available rating scores per user), proportionally, to each book based on the overall similarity score. This means for books most similar to the books read by the user a larger proportion of the 1500 points will be assigned to it. These values were then scaled back to a rating out of 10 for each book. The ratings predicted in this method had a maximum of whatever the maximum rating of the user was on the books they already rated. This was because we can only tell which books are most similar to the books the user already read. We cannot tell if a user will like similar books more than the ones already read. In this case we can only predict that they will like it as much as they liked what they already read or they will like it less.

Users from the train set was used again as input to the item based function and the results were predicted and compared to the original results. The root mean square error was again used to calculate the accuracy.

Matrix Factorisation

With matrix factorisation the users and books are decomposed into a certain amount of latent factors. These factors are unknown but interact with each other in such a way that it produces predicted ratings. In this approach we used the recosystem package to decompose the users and books into latent factors.

The process of matrix factorisation is to predetermine a certain amount of latent factors for both users and books. In our case 20 latent factors were chosen. The cross product between the user latent factors and book latent factors result in a predicted rating. Initially the latent factors are randomly estimated but are changed with respect to a loss function given by:

$$\min_{P,Q} \sum_{(u,v)\in R} \left[f(p_u, q_v; r_{u,v}) + \mu_P ||p_u||_1 + \mu_Q ||q_v||_1 + \frac{\lambda_P}{2} ||p_u||_2^2 + \frac{\lambda_Q}{2} ||q_v||_2^2 \right]$$

Where P and Q are latent factor vectors of users and books respectively. Lowercase p and q refer to the individual latent factors. λ_P and λ_Q are L2 regularisation values for users and items respectively. This L2 regularisation factor penalises large parameter values. The latent factors are thus altered as to minimise the above function.

Two different matrix factorisation models were created. One with no L2 regularisation term and one with the L2 regularisation term. The L2 regularisation values are 0.1 for the users and 0.01 for the books. In this case we observe the cross validation error to identify which model works best.

Matrix Factorisation with no Regularisation

```
iter      tr_rmse          obj
   0       4.8347   3.0088e+05
   1       1.6144   3.3549e+04
   2       1.2887   2.1376e+04
   3       1.1721   1.7685e+04
   4       1.0627   1.4536e+04
   5       0.9375   1.1314e+04
   6       0.7987   8.2115e+03
   7       0.6737   5.8415e+03
   8       0.5690   4.1681e+03
   9       0.4872   3.0556e+03
  10       0.4273   2.3504e+03
  11       0.3799   1.8575e+03
  12       0.3430   1.5140e+03
  13       0.3122   1.2542e+03
  14       0.2884   1.0706e+03
  15       0.2665   9.1416e+02
  16       0.2487   7.9610e+02
  17       0.2329   6.9847e+02
  18       0.2193   6.1912e+02
  19       0.2066   5.4944e+02
```

Matrix Factorisation with Regularisation

iter	tr_rmse	obj
0	4.8289	3.0699e+05
1	1.6020	3.9983e+04
2	1.2952	2.8539e+04
3	1.1907	2.5305e+04
4	1.0907	2.2526e+04
5	0.9804	1.9791e+04
6	0.8584	1.7106e+04
7	0.7385	1.4861e+04
8	0.6330	1.3152e+04
9	0.5481	1.1988e+04
10	0.4825	1.1206e+04
11	0.4321	1.0681e+04
12	0.3928	1.0315e+04
13	0.3628	1.0062e+04
14	0.3368	9.8536e+03
15	0.3168	9.7061e+03
16	0.2996	9.5955e+03
17	0.2849	9.4955e+03
18	0.2716	9.4107e+03
19	0.2609	9.3485e+03

As can be seen from the above graphs, the matrix factorisation without the L2 regularisation factor performed slightly better in terms of the cross validation error.

Ensamble method

The ensamble method was created by taking the average of all the previous predicted ratings for each book and assigning it to the new user. That is, calculating the predicted rating using user based collaborative filtering, item based collaborative filtering and matrix factorisation without the L2 regularisation factor. The average of these three values are then taken and this rating is assigned for a particular book for a user.

The initial idea was to find the k most similar users to a new user based on a precalculated similarity matrix of the users in the train set. The cross product of the latent factors of these k nearest neighbours and the book latent factors would then produce k different rating values for each book. The average of these k values would then be taken to calculate the predicted rating for the new user for each book. This approach, however, didn't work as it was often the case that there was either not enough similar users or no similar users to the new user.

The users in the training dataset is again used as input to this ensamble method and the RMSE is obtained in the same way, by comparing the predicted rating for a book with its original rating.

Results and Discussion

Table 1: RMSE for each model

	RMSE
User based	2.164
Item Based	0.721
Matrix Fac.	0.176
Ensamble	1.335

The results show that the matrix factorisation approach was the most effective as it produced the lowest RMSE score. The matrix factorisation approach is a more sophisticated and well tested user recommendation method. The user based approach and the item based approach had makeshift and clearly inaccurate ways of predcting user ratings for unread books. Between the user based and item based approach the later performed better. This could be because the calculation of the predicted rating for the item based approach relied heavily on the rank order of the similarity scores.

The similarity scores are arguably better predictors of which books a user might want to read next. This is because it shows which book is most similar to the books a user likes or has already read. Most of the time people are drawn to a specific genre or a specific author and the book similarities adhere well to this. Often, upon observation, the most similar book to another is written by the same author and, therefore a similar genre too.

The similarity scores between users (in the user based approach) is also an arguably better method of determining which books a user might like than predicting a rating for each book. This is because similar people will like similar books and therefore is a good way to predict what book a user might like. The difficulty, however, is converting from similarity scores to a predicted book rating. The k nearest neighbours approach, in theory, seemed to be a good idea but the sparsity of the data inhibits its performance. This shows clearly as it is the worst performing method.

Conclusion

The tried and tested way to develop recommender systems, such as matrix factorisation methods, are best. Similarity scores between users or similarity scores between books are good indicators of which books a user might want to read next, but converting this similarity score to a predicted book rating is not a simple task. In practice, if a user based or item based recommender system has to be used, then the best way to go about it is to recommend items based on similarity scores alone.

The sparse nature of rating data is a primary cause in large inaccuracies in predicted book rating or similarity rating. In the above approaches, if any user has only read and rated 2 or 3 books it is unlikely that the system can accurately predict ratings for every book. A few very similar users and very similar books may produce an accurate rating for a small portion of books but for the most part the predictions will be fairly inaccurate. In the end, tried and tested methods are best to use.