Incremental SVD and SVD: Optimal Alternatives for Book Recommender Systems

Abstract
Recommendation of useful books to the customers referring to online bookshops or even traditional book- stores could constitute a win-win approach. But, the key question of how such an approach should be adopted still remains unanswered. One of the possible solutions recently proposed is the 'Recommender System with Collaborative Filtering Approach'. In the case of book recommendation systems, since we have been encountering a large sparse user-item matrix representing each user rating to the set of items, utilizing model-based algorithms seems to be an appropriate option. The present paper is aimed at finding a suitable algorithm among SVD algorithms family (Incremental SVD and SVD in this case) and k-NN by evaluating these matrix factorization algorithms applied to the Book-crossing[1] and goodreads comic books[2] datasets and comparing them with k-NN[3] algorithm. The experimental results based on the mentioned datasets indicate that higher accuracy is obtained through matrix factorization algorithms means SVD and Incremental SVD (SVD++) versus k-NN leading the authors to design their own Book Recommender System accordingly.

Keywords: Recommender system, Collaborative Filtering, matrix factorization, SVD, SVD++.

1. Introduction

The ever-expanding of information diversity range across the internet has caused a lot of difficulties for the users in terms of both time and cost. In detail, wide range of items in the market renders decision making a confusing and time-consuming process. Recommender Systems help better recognition of users' expectations, leading to providing them with a more efficient selection process. Hence, the application of Recommender Systems as an advanced tool of technology is not just a viable solution for the customers but rather a seminal one for the seller in the competitive markets. Therefore, introducing an applicable algorithm among k-NN, SVD, and SVD++ to design these systems for the mentioned case could be a problem tackled in this article.

Recommender Systems are software tools and methods providing suggestions of items to users. "Item" is the general term signifying what the system recommends to a user [1]. This might be a book, music or any other entity. One of the most applicable models in Recommender Systems is Collaborative Filtering. This approach can generate personalized recommendations [2]. It works through searching large groups of people and finding smaller set of users with tastes similar to the target user.

In model-based Collaborative Filtering, users' rating is used to learn a predictive model. The main idea is to model the user-item interactions with factors representing latent characteristics of the users and items in the system, like the preference class of users and the category class of items. This model is then trained using the available dataset, and later used to predict ratings of users for new items [1]. Matrix Factorization algorithms like SVD and SVD++ are instances of model-based methods.

Each model enjoys its own advantages and the drawbacks that impress the system performance. The major problems of a model could be low accuracy in some cases and high rate of data dispersion. In our problem we face a sparse user-item matrix in which each row represents a user's rating relative to items and each column is indicator of an item. For sparse user-item matrices, reducing dimensions can improve the performance of the algorithm in terms of both space and time. To achieve this goal, we applied the mentioned matrix factorization algorithm on "Book-crossing" dataset to build a Recommender System. The experimental results showed that these algorithms worked slightly better than k-NN family.

[1] Book-crossing dataset

[2] Goodreads graphical comic books dataset

[3] k-Nearest Neighbors

The structure of this paper consists of the related works in the second part, materials and methods in the third part, and a comparison of the experimental results in the fourth part. Finally, the conclusion is presented in the last part.

2. Related works

Methods that are used typically in Collaborative Filtering are memory-based methods and model-based methods [3]. Suggestions of memory-based method rely on direct accessing to database, whereas model-based method make suggestions by creating model from transaction data [4]. User-based and Item-based approaches are common in Collaborative Filtering systems [5]. User-based technique proposes items on the basis of similarity of other users to the target user [6]. Nevertheless, Item-based technique finds items that are target user-rated and consider their similarity to the target item and recommends top-k of them [7].

M.G Vozalis et al. applied matrix factorization algorithm SVD on item-based filtering [8]. Xun Zhou et al. proposed an incremental algorithm called Incremental ApproSVD which is a suboptimal approximation with lower running time [9]. M.G Vozalis et al. used SVD and demographic data for the enhancement of generalized Collaborative Filtering [10]. Yangcheng Jia et al. recognized users' brands preference based on SVD++ [11]. Yehuda Koren et al. introduced Matrix Factorization techniques for Recommender Systems [12]. R. Kumar et al. examined social popularity based SVD++ Recommender System [13]. Ramakrishnan Kannan proposed bounded Matrix Factorization for Recommender Systems [14]. D. Kun presented a research of personalized book recommender system for university library based on collaborative filtering [15].

3. Materials and methods

Showing collaborative filtering basic function in a figure could be helpful, illustrated in the next parts of this section.

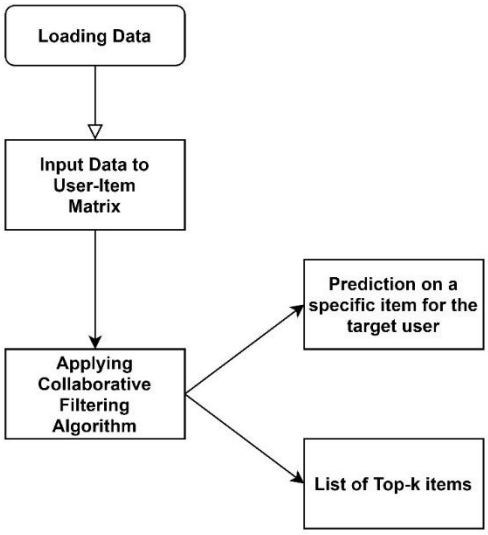

Fig1. Collaborative Filtering process of making recommendation output in a simple form

As a powerful tool, we used Surprise[4] that is a python library to implement Recommender Systems' algorithms and provide information about methods of this paper.

In this section, the rating set is designated by R and the true rating of user u for item i is denoted by r_{ui} . Nonetheless, the predicted set is signified by $\hat{R}$ and the prediction of rating user u for item i is represented by $\hat{r}_{ui}$. As a further matter, I represents the set of items and U is the set of users.

[4] Surprise Documentation

3.1 SVD

SVD algorithm had effective performance during the Netflix Prize [16] and it is considered as an equivalent to Probabilistic Matrix Factorization [17]. $\hat{r}_{ui}$ in this method is calculated as below:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

where, μ is the mean of all ratings, b_u defines as the baseline ratings of user u, b_i is the baseline ratings of item i that baselines initialized to zero and q_i, p_u are both factors of I, U respectively. We consider 20 factors initialized by a normal distribution with 0 for its Mean and 0.005 for its Standard Deviation. If the user is unknown, then the b_u and p_u are assumed to be zero. This fact is also hold about b_i and q_i for unknown items. More computational detail proposed in [12] and [1]. Therefore, we have adequate content to discuss the process of learning in SVD shortly. In order to predict the ratings, we minimize the following regularized squared error using a simple SGD[5] (regularization term $\lambda = 0.02$). In detail, we set the number of SGD iteration to 20 and learning rate γ to be 0.01.

(1) $\sum_{r_{ui} \in R_{train}} (r_{ui} - \hat{r}_{ui})^2 + \lambda(b_i^2 + b_u^2 + ||q_i||^2 + ||p_u||^2)$

by defining $e_{ui} = r_{ui} - \hat{r}_{ui}$ we have:

(2) $b_u \leftarrow b_u + \gamma(e_{ui} - \lambda b_u)$
(3) $b_i \leftarrow b_i + \gamma(e_{ui} - \lambda b_i)$
(4) $p_u \leftarrow p_u + \gamma(e_{ui} - \lambda p_u)$
(5) $q_i \leftarrow q_i + \gamma(e_{ui} - \lambda q_i)$

In each of the iterations we apply formulas (2) to (5) in order to minimize equation (1). To understand better the idea of Matrix Factorization, showing an image behind SVD is worthwhile.

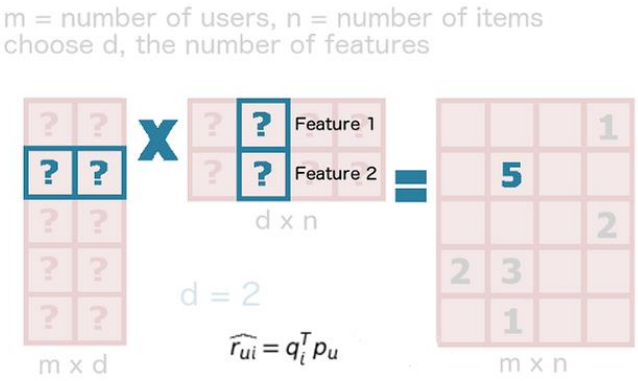

Fig2. Describing factorization of one matrix using two factors

3.2 k-NN

In Recommender Systems, this algorithm inspired by a basic k-NN approach for user-based method is represented as follows:

$$\hat{r}_{ui} = \frac{\sum_{v \in N_i^k(u)} Sim(u, v) \cdot r_{vi}}{\sum_{v \in N_i^k(u)} Sim(u, v)}$$

For item-based method:

[5] Stochastic Gradient Descent

$$\hat{r}_{ui} = \frac{\sum_{j \in N_u^k(i)} Sim(i,j) \cdot r_{uj}}{\sum_{j \in N_u^k(i)} Sim(i,j)}$$

In which, $N_i^k(u)$ is the k nearest neighbors of user u that have rated item i. In turn, $N_u^k(i)$ is the k nearest neighbors of item i which is rated by user u. In general, $Sim(u,v)$ is the similarity between u, v based on a similarity measure. In this problem, $1 \leq k \leq 40$, the approach is user-based and similarity measure is Mean Squared Difference. The important point that should be regarded in using k-NN is choosing the fitting number of neighbors.

3.3 SVD++

The difference between SVD and SVD++ lies in consideration of implicit rating by SVD++. Calculation of $\hat{r}_{ui}$ in this algorithm is as follows:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \left(p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j \right)$$

where, I_u is the set of items rated by user u and y_j is the set of item factors which consider implicit rating.

The mentioned property of unknown items or/and users in SVD is also satisfied in this algorithm and more details about that could be found in [18]. Just as for SVD, the parameters are learned using a SGD on the regularized squared error objective. In our configuration of algorithm, all initializations are the same as SVD. In addition, the process of minimization using SGD is quite similar to the SVD process mentioned above. In order to clarify the process of work the source code [6]has presented.

4. Experimental Results

Accuracy of the models is computed based on 62656 samples of Book-crossing dataset and 62489 records of goodreads graphical comic books dataset. Statistical detail about ratings are been demonstrated in the tables below:

Table1. Statistical measures for Book-crossing dataset

Count	62656
Mean	7.9537
Std Deviation	1.7178
Min	1
1st Quartile	7
2nd Quartile	8
3rd Quartile	9
Max	10

[6] Source Code

Table2. Statistical measures for goodreads graphical comic books dataset

Count	62489
Mean	4.0205
Std Deviation	0.9368
Min	1
1st Quartile	3
2nd Quartile	4
3rd Quartile	5
Max	5

Besides, the share of test set in these models is 20%; the results have been examined based on MAE[7] and RMSE[8] metrics. If we define $\hat{R}$ as the set of predicted ratings and n as the number of members in prediction set, then MAE and RMSE are computed by the following formulas:

$$MAE = \frac{1}{n} \sum_{\hat{r}_{ui} \in \hat{R}} |r_{ui} - \hat{r}_{ui}|$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{\hat{r}_{ui} \in \hat{R}} (r_{ui} - \hat{r}_{ui})^2}$$

Table3. Evaluation of algorithms based on the given metrics for Book-crossing dataset

Book-crossing	**MAE**	**RMSE**
SVD	1.13	1.5011
k-NN	1.2077	1.6674
SVD++	1.1242	1.4977

Table4. Evaluation of algorithms based on the given metrics for goodreads graphical comic books dataset

Goodreads graphical comic books	**MAE**	**RMSE**
SVD	0.6616	0.8459
k-NN	0.6928	0.9438
SVD++	0.6615	0.8457

Regarding the results contained in Tables (3) and (4), we conclude that in both datasets SVD++ algorithm produces a slight difference compared with SVD. Furthermore, the accuracy of SVD++ and SVD are higher than k-NN.

We have just compared the histogram of SVD++ with k-NN and ignored the SVD histogram in this comparison.

[7] Mean Absolute Error

[8] Root Mean Squared Error

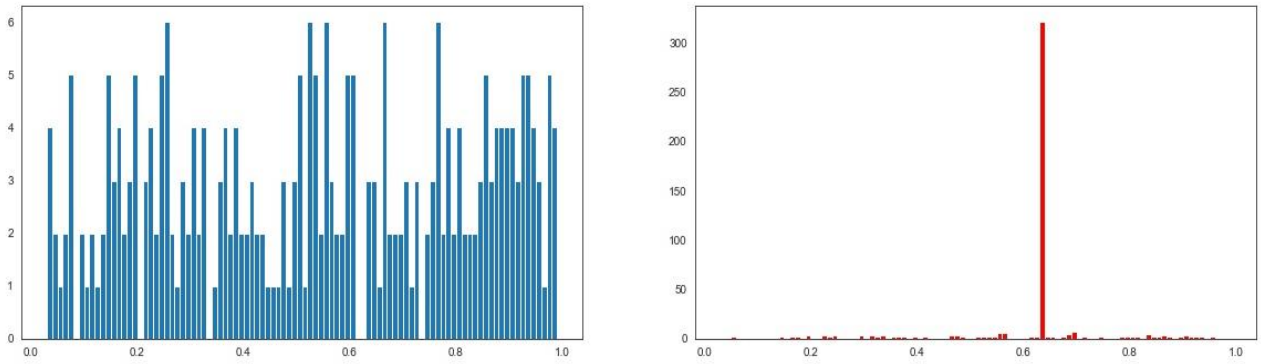

Fig3. Percentage error histograms of SVD++ and k-NN (left to right) for Book-crossing dataset

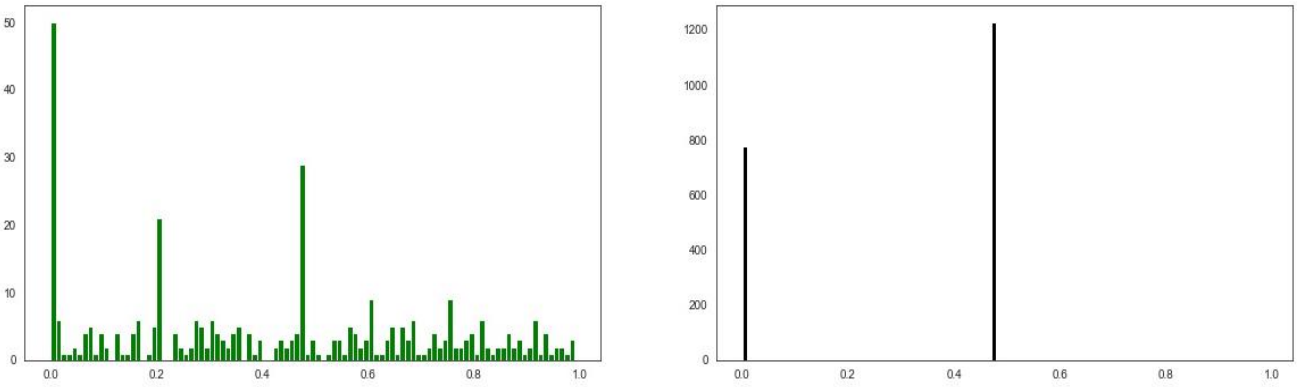

Fig4. Percentage error histograms of SVD++ and k-NN for Goodreads graphical comic books dataset

Histograms apparently represent better performance of SVD++. Here is the formula applied for the computation of this error:

$$percentage\_error = \frac{|r_{ui} - \hat{r}_{ui}|}{r_{ui}} \times 100$$

5. Conclusion

Thanks to Recommender Systems, selection between books you want to buy does not take too much time since it advises items to users according to the users' desire, such as ratings, reviews and clicks. One of the most applicable models used in these systems is Collaborative Filtering. Given that the user-item matrix in book datasets is sparse, thus the performance in the Book Recommender System could reach its peak by dimension reduction. Hence, we used two Matrix Factorization algorithms, SVD and SVD++, and compared the results with k-NN which illustrates the excellence of these algorithms in this case. Hence, the problem of finding an applicable algorithm to build recommender systems in case of books was solved.

For the future works, we recommend the examination of SVD++ and SVD for book recommender systems to analyze the strength points of the algorithm and improve them to a better method for these systems. In addition, apart from the book recommendation problem discussed above, considering other recommendation problems that SVD++ and SVD can properly handle are good cases for further studies.

References

1. Ricci F, Rokach L, Shapira B (2015) Recommender systems: introduction and challenges. In: Recommender systems handbook. Springer, pp 1-34

2. Redpath J, Glass DH, McClean S, Chen L Collaborative filtering: The aim of recommender systems and the significance of user ratings. In: European Conference on Information Retrieval, 2010. Springer, pp 394-406

3. Aggarwal CC (2016) Recommender systems, vol 1. Springer

4. Aditya P, Budi I, Munajat Q A comparative analysis of memory-based and model-based collaborative filtering on the implementation of recommender system for E-commerce in Indonesia: A case study PT X. In: 2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS), 2016. IEEE, pp 303-308

5. Wang J, De Vries AP, Reinders MJ Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, 2006. pp 501-508

6. Ge F (2011) A user-based collaborative filtering recommendation algorithm based on folksonomy smoothing. In: Advances in Computer Science and Education Applications. Springer, pp 514-518

7. Sarwar B, Karypis G, Konstan J, Riedl J Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web, 2001. pp 285-295

8. Vozalis MG, Margaritis KG Applying SVD on item-based filtering. In: 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), 2005. IEEE, pp 464-469

9. Zhou X, He J, Huang G, Zhang Y (2015) SVD-based incremental approaches for recommender systems. Journal of Computer and System Sciences 81 (4):717-733. https://doi.org/10.1016/j.jcss.2014.11.016

10. Vozalis MG, Margaritis KG (2007) Using SVD and demographic data for the enhancement of generalized collaborative filtering. Information Sciences 177 (15):3017-3037. https://doi.org/10.1016/j.ins.2007.02.036

11. Jia Y, Zhang C, Lu Q, Wang P Users' brands preference based on SVD++ in recommender systems. In: 2014 IEEE workshop on advanced research and technology in industry applications (WARTIA), 2014. IEEE, pp 1175-1178

12. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. Computer 42 (8):30-37. https://doi.org/10.1109/MC.2009.263

13. Kumar R, Verma B, Rastogi SS (2014) Social popularity based SVD++ recommender system. International Journal of Computer Applications 87 (14). https://doi.org/10.5120/15279-4033

14. Kannan R, Ishteva M, Park H (2014) Bounded matrix factorization for recommender system. Knowledge and information systems 39 (3):491-511. https://doi.org/10.1007/s10115-013-0710-2

15. Kun D (2012) Research of personalized book recommender system of university library based on collaborative filter. Data Analysis and Knowledge Discovery (11):44-47

16. Simon Funk (2006) Netflix Prize and SVD. The Evolution of Cybernetics A Journal. https://sifter.org/~simon/journal/20061211.html. Accessed 12 March 2020.

17. Mnih A, Salakhutdinov RR Probabilistic matrix factorization. In: Advances in neural information processing systems, 2008. pp 1257-1264

18. Koren Y Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 2008. pp 426-434