10th International Conference on Control Engineering &Information Technology (CEIT-2025)
Proceedings Book Series –PBS- Vol 23, pp.343-349

Matrix factorization based model for book recommendation

BENABBOU Amin[#1], HAMDADOU Djamila[*2]

[#]*Department of social sciences, University of Tiaret, Algeria*

amin.benabbou@univ-tiaret.dz

**Department of Computer Science, Laboratory LIO, University Oran 1, Algeria*

dzhamdadoud@yahoo.fr

***Abstract*— In this article, we have covered the main concepts that relate to the use of matrix factorization methods in the context of a book recommendation system that can be integrated into an online bookstore. We experimented with two implementations of this kind of models to study the impact of integrating the biases, which usually characterize readers and books, on the quality of prediction. This experiment allowed us to conclude that the matrix factorization based recommendation strategy that took into account biases in the learning stage produced fewer prediction errors compared to the strategy that did not integrate them. Another experiment allowed us to compare the performance of the best experimented matrix factorization based model with that of neighborhood-based model. The obtained results allowed us to conclude that the first strategy produced slightly fewer prediction errors than the second one.**

***Keywords*—Book recommendation system, Collaborative filtering, Dimensionality reduction, Matrix factorization**

I. Introduction

The proliferation of digital libraries, like online services, has flooded readers with an ever-increasing number of books both printed and digital. Recommender systems (RSs), whose objective is to estimate how much a user will like items that he has not yet seen and to suggest those that are likely to interest him, have been proposed to deal with the information overload to which the consumer is usually exposed each time he wants to acquire a new book, buy a product from an online store, watch a content in a streaming platform such as YouTube or Netflix and so on.

Integrating a recommendation system into a digital bookstore allows the latter to acquire a form of intelligence because, after some number of interactions between readers and the proposed books, the bookstore in question gradually learns the preferences of each reader and becomes able to suggest books that are likely to interest him.

In the context of recommender systems, matrix factorization is a recommendation strategy of type: *Collaborative Filtering* (*CF*). Collaborative filtering is the name given to approaches that rely exclusively on the past interactions of users with a system (a digital library, an online store, etc.) to predict their behavior toward products they have not yet seen. These interactions are often represented by a matrix that we call R, where r_{ui} represents the rating given by the user *u* to the item *i* (book, product, video, etc.). The goal of matrix factorization is to decompose R into two matrices, called P and Q, whose dimensions are too small compared to those of R using some number of factors. The entries of the resulting matrices will then be used to predict the unknown ratings of R.

According to [1], previous works, like [2] and [3], relied on imputation in which missing values in R were replaced with plausible values (like the mean of known ratings of a row or a column) to make it dense, then a factorization method- like *Singular Value Decomposition* (SVD)- was applied to the resulting matrix. The major drawback of these works is the introduction of inaccurate ratings which will necessarily have negative consequences on the quality of prediction.

More recent works, like [4] and [5], have suggested to see the problem of factorizing R as an optimization problem in which only the known ratings were used to obtain the entrees of the matrices: P and Q. According to [6], this choice is justified by the fact that R is made of redundant values (often between 1 and 5).

ISSN: 2961-6611

In this work we propose to use factorization of the rating matrix as a book recommendation strategy. The factorization is based on the use of a *Stochastic Gradient Descent* algorithm to find the coefficients of the matrices that result from this decomposition. In this article we have covered the following points:

- We have addressed the main concepts that relate to recommendation strategies that are based on the factorization of the rating matrix.
- We have showed the impact of integrating users and books biases on the prediction quality of the experimented matrix factorization based prediction model.
- We have compared the performance of the experimented matrix factorization based prediction model with that of a model that is neighbourhood-based.
-

The rest of the article is organized as follows: in the next section, we present a synthesis of the state of the art covering the main concepts that we have covered in this study. In the third section, we proceed to a detailed description of the recommendation systems that were tested and to the analysis of the obtained results. The article ends with a conclusion in which we recall the main results of this study.

II. Related works

In this section, we delve into the central concepts that structure the recommendation strategies that are based on factorizing the rating matrix using an optimization algorithm in addition to give a summary description of neighbourhood-based recommendation approaches.

A. Matrix factorization based recommendations:

In the context of recommender systems, factorization is used to characterize users and items using two matrices, Q and P respectively, with respect to some number of factors. Each factor represents dominant correlation pattern in the rating matrix [6]. Each item “i” is associated with a vector q_i (a row in the matrix Q), and each user “u” is associated with a vector p_u (a row in the matrix P).The vectors q_i and p_u measure the affinity of “i” and “u” respectively toward the set of factors.

1) Obtaining the Coefficients of the Matrices P and Q

In a matrix factorization based recommendation strategy, the main challenge is to calculate the coefficients of the two matrices P and Q using R which contains several missing values. Once these coefficients are computed, the ratings that are not known ($\hat{r}_{ui}$) can be estimated using the following equation:

$$\hat{r}_{ui} = p_u q_i^T \qquad (1)$$

The coefficients of the vectors: p_u and q_i can be obtained using an optimisation algorithm that minimizes the following function [7]:

$$\Psi = \sum_{u,i \in k} (rui - \hat{r}\,ui)^2 + \lambda\left(\left||qi|\right|^2 + \left||pu|\right|^2\right) \qquad (2)$$

where:

- K: is the set of pairs (u, i) for which r_{ui} is known.
- λ: controls the extent of the regularization.
- $||\, v \,||^2$: denotes the squared Frobenius norm of the vector v.

According to [7], the minimization of function (3) can be done using a Stochastic Gradient Descent algorithm. The latter looks like this:

The Stochastic Gradient Descent Algorithm:

Inputs:

- **R**: the rating matrix.
- **α**: the learning rate.

ISSN: 2961-6611

- **λ**: a parameter that controls the extent of the regularization.
- **F**: number of factors.

Outputs:

- **P**: a matrix that characterizes the users.
- **Q**: a matrix that characterizes the items.

Begin

Randomly initialize the coefficients of P and Q ;
S= {(u,i) where r_{ui} is known} ;
As long as there is no convergence do
Begin
    For each (u,i) in S do
    Begin
        $e_{ui}= r_{ui} -\sum_{j=1}^{F} q_{uj}\, p_{ij}$;
        For each k ϵ {1,..,F} do $p^{+}_{uk}= p_{uk}+ \alpha\, (e_{ui}\, q_{ik} -\lambda\, p_{uk})$;
        For each k ϵ {1,..,F} do $q^{+}_{ik}= q_{ik}+ \alpha(e_{ui}\, p_{uk} - \lambda\, q_{ik})$;
        For each k ϵ {1,..,F} do
        Begin
            $p_{uk} = p^{+}_{uk}$;
            $q_{ik} = q^{+}_{ik}$;
        End
    End
End

End

In the previous algorithm, the temporary variables p^{+}_{uk} and q^{+}_{ik} are used to store the results of the changes that must to be made to the variables: p_{uk} and q_{ik}.

2) Biases Integration:

The prediction model that was previously described explains the variation of the ratings only by the interaction of users with the articles (equation 01), however, a part of these variations can also be explained by the generosity of users in assigning the ratings or by the popularity of the items [6]. These two aspects are called biases and are independent of any interaction. According to [7], a rating prediction model based on stochastic gradient descent and integrating user and article biases learns by minimizing the following function:

$$\Psi=\sum_{(u,i)\in k}(rui - \mu - bu - bi - \hat{r}ui)^2 + \lambda(\left||qi|\right|^2 + \left||pu|\right|^2 + bu^2 + bi^2) \quad (3)$$

Where:

μ: is the overall average of all the known ratings in R.

$\mathbf{b_u}$: is the deviation from μ that is observed in the ratings that are assigned by the user u.

$\mathbf{b_i}$: is the deviation from μ that is observed in the ratings that are attributed to the article i.

B. Neighbourhood-Based Recommendations

In a neighborhood-based recommendation strategy, the prediction of unknown ratings is based on the search for the nearest neighbors of an item (when the CF is centered on items) or on the search for the nearest neighbors of a user (when the CF is centered on users). In this study we limited ourselves to experimenting with a few recommendation strategies that are centered on items because, as already pointed out by other researchers, in particular:[6], these strategies often provide more relevant recommendations than user-centric strategies. Three important steps must be made in creating an item-centric recommendation system [8]:

- Normalizing the rating matrix: using *mean-centering* or *Z-score* for example.
- Calculating the degree of similarity between each pair of items using measures like: *Pearson's correlation coefficient* or *cosine similarity*.

ISSN: 2961-6611

10th International Conference on Control Engineering &Information Technology (CEIT-2025)
Proceedings Book Series –PBS- Vol 23, pp. 343-349

- Predicting unknown ratings: the quality of the prediction depends on the normalization method, the measure of distance, and the number of neighbors.

III. METHODOLOGY AND OBTAINED RESULTS

The purpose of this section is to provide a detailed description of the methodology we followed in creating and evaluating the prediction models that were tested. This description also includes the database that was used and the results that were obtained from the comparison of the models in question.

A. *The Database*

The models that were experimented in this study were trained and tested using a reduced version of the GoodBooks-10K[1] database. The reduced version of this database contains: *8962 books*, *10,000 users* and *1,163,774 interactions* of the type: (*user_id*, *book_id*, *the rating*). The database contains on average: 116.38 ratings per user and 129.86 ratings per book.

B. *Evaluation Procedure and Tools*

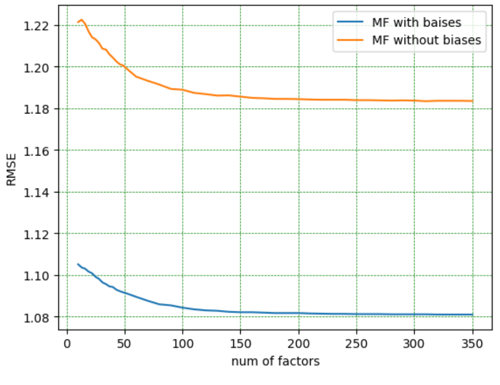
To measure the quality of the predictions made by the experimented models we divided the initial content of the database into three parts: training, validation and test using a random sampling. The experimented prediction models were trained and tested using Python and Surprise [9]. The performance that was observed in this study concerns the accuracy of the predictions which was measured using the Root Mean Squared Error (RMSE).

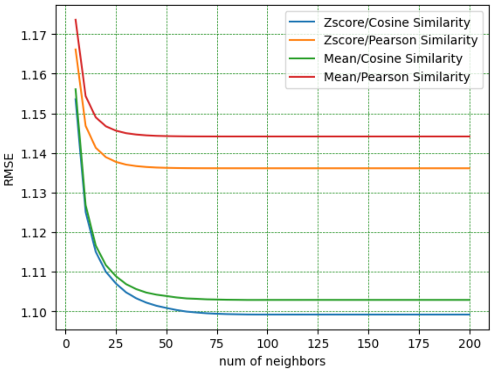
C. *The Experimented Models*

1) *Matrix Factorization Based Models:*

Two experiments were conducted as part of a recommendation strategy that relays on the factorization of the rating matrix. In the first experiment, the trained model (*MF without biases*) estimated the unknown ratings using the known ratings only. In the second experiment, the model (*MF with biases*) estimated the unknown ratings using the known ratings while taking into account the biases that usually characterize readers and books. The aim was to see the effect of integrating these biases on the prediction quality. Fig. 01 (a) shows RMSE of the two models as a function of the number of factors, F, that were considered in the training phase (**F϶ [10, 350]**). The two variants of the recommendation system (unbiased and biased) converged for **α=0.05** and **λ=0.1**. From the diagram in Fig. 01 (a) we can draw the following conclusions:

- As the number of factors, F, increases, RMSE of both models decreases. The prediction errors produced by both models become more stable when **200 factors** or more are used to characterize users and books.
- The model integrating biases produced fewer prediction errors than the model that did not integrate them, for all the values of F.





[1] https://github.com/zygmuntz/goodbooks-10k

ISSN: 2961-6611

(a) *Matrix factorization based models* (b) *Neighborhood-based models*

Fig. 01 Evolution of the RMSE as a function of the number of factors (a) and the number of neighbors (b)

2) *Neighbourhood-Based Models:*

Four experiments were conducted as part of a recommendation strategy that relays on the neighborhood of books (*item-centered*). These experiments are distinguished from each other by the method that was used in normalizing the ratings (*Mean-centering*, *Z-score*) and the measure that was used in calculating the degree of similarity between each pair of books (*Pearson correlation coefficient*, *Cosine similarity*). The RMSE of these four variants was observed while varying the number of neighbors, which were considered in predicting the unknown ratings, from **5 to 2000**. The aim was to observe the effect of these choices on the prediction quality. The diagram in Fig. 01 (b) allows us to draw the following conclusions:

- The recommendation strategy that relied on the Z-score and the Cosine similarity produced fewer prediction errors than the other models, regardless of the number of neighbors that were considered in the prediction of the unknown ratings.
- As the number of neighbors increases, the prediction errors of the four models decrease. In the case of the model with the smallest RMSE, namely: *Z-score/cosine similarity*, the prediction errors start to become stable when **80 neighbors** or more were considered in the prediction of the unknown ratings.

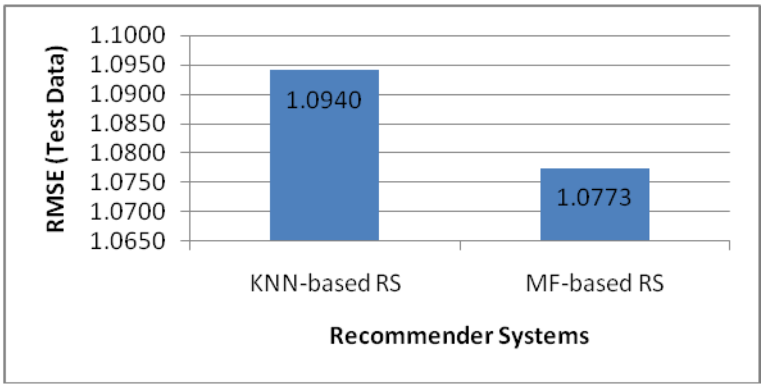


Fig. 02 The RMSE of the two recommendation strategies

3) *Comparison Between the Two Strategies:*

After identifying the best performing prediction model within each recommendation strategy, we proceeded to compare the performance of the chosen models using the test data this time. The diagram in Fig. 02 shows that the RMSE of MF-based RS was slightly lower than that of KNN-based RS.

IV. CONCLUSION

In this article, we first highlighted the importance for digital libraries to have a recommendation system if we want to offer readers an effective and personalized way to explore their content. We then covered the main concepts related to recommendation strategies that rely on the use of matrix factorization models. We also compared the performance of these recommendation strategies with that of a neighborhood-based strategy. The latter is well known in this field and is often considered as a reference against which the performance of more sophisticated models is compared. The experiments conducted in this study allowed us to draw the following conclusions:

- In a matrix factorization based model, the greater the number of factors, that are taken into account in the training step, the lower its prediction errors.
- A matrix factorization based model that integrates the biases produces fewer prediction errors than a prediction model of the same type that does not integrate them..

ISSN: 2961-6611

- Matrix factorization based recommendation strategy was slightly more accurate than a neighborhood-based recommendation strategy
- .

References

[1] Koren, Y., Rendle, S., & Bell, R. (2021). "*Advances in collaborative filtering*". Recommender systems handbook, 91-142.

[2] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. T. (2000). "*Application of dimensionality reduction in recommender system-a case study*".

[3] Kim, D., & Yum, B. J. (2005). "*Collaborative filtering based on iterative principal component analysis*". Expert systems with Applications, 28(4), 823-830.

[4] Bell, R., Koren, Y., & Volinsky, C. (2007, August). "*Modeling relationships at multiple scales to improve accuracy of large recommender systems*". In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 95-104).

[5] Paterek, A. (2007, August). "*Improving regularized singular value decomposition for collaborative filtering*". In Proceedings of KDD cup and workshop (Vol. 2007, No. 2007, pp. 5-8).

[6] Aggarwal, C. C. (2016). "*Recommender systems*" (Vol. 1). Cham: Springer International Publishing.

[7] Koren, Y., Bell, R., & Volinsky, C. (2009). *Matrix factorization techniques for recommender systems*. Computer, 42(8), 30-37.

[8] Adomavicius, G., & Tuzhilin, A. (2005). "*Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions*". IEEE transactions on knowledge and data engineering, 17(6), 734-749.

[9] Hug, N. (2020). "*Surprise: A Python library for recommender systems*". Journal of Open Source Software, 5(52), 2174.

ISSN: 2961-6611