Matrix Factorization for Recommender System on Music Data using Stochastic Gradient Descent

Mannat Soni
Advanced Financial Technology Course Assessment 1

I. Introduction

A. Recommender system

Today everything around us is data and by merging that data with the power of Machine Learning, companies like Netflix and Amazon use techniques like Recommender systems to continuously try and improve user experiences [1]. A recommender system is ideally used for generating recommendations of particular items or products that customers might like, depending on different attributes [1]. There are two kinds of recommender systems, one that uses profile attributes called content filtering and the other that analyzes historical interactions called collaborative filtering [2].

B. Matrix factorization

The main work behind the recommender system comes from matrix factorization by generating latent features [1] because patterns are observed in user-generated data. Had the data generated been random, performing matrix factorization would have yielded no significant results.[7] In matrix factorization, factorization is done by splitting the data into two matrices P and Q where P is the user vector and Q is the item vector. The dot product of the two vectors yields the user's overall interest in item I (equation 1) [1]. To find the optimal values of P and Q, Stochastic Gradient Descent is used. [1][3].

$$\hat{r}_{ui} = P_u^\top Q_i \tag{1}$$

C. Singular Value Decomposition vs Stochastic Gradient Descent

The data obtained is a sparse matrix and SVD is undefined for matrices with missing values as it assumes that the user-data matrix has no null values and interprets missing values as 0 [7]. While SGD minimizes the error generated by the difference between actual ratings vs the predicted ratings. To avoid over-fitting, a regularization term λ (Equation 2) is introduced which helps in regularizing the learnt parameters [1]. The main idea is for our model to act more complex when we have sufficient data and become more simple when we don't, which is controlled by λ [7].

$$\min_{q \cdot p \cdot} \sum_{(u,i)\in\kappa} \left(r_{ui} - p_u^T \cdot\ q_i\right)^2 + \lambda \left(\|p_u\|^2 + \|q_i\|^2\right) \tag{2}$$

II. Design and Analysis of the Experiment

A. Experiment with default hyperparameters

1) Dataset: Splitting into Training and Validation sets: The dataset of Digital Music consists of 50,000 rows and 4 columns. There are 34947 unique users, 1449 unique items and 5 unique ratings. The dataset is sparse due to a huge number of missing values. On computing, the total percentage of user-items that have ratings is found to be 0.10%, i.e. 0.10% of ratings have values. In order to tackle this huge number of missing values, these values are replaced with 0 [4]. The dataset is split into training and validation sets in such a way that if the length of ratings that are non-zero is greater or equal to 35, 15 ratings per user from the training set will be added to the validation set [4]. This gives us (train, val) = (34947, 1449).

2) Performing Matrix Factorization: To perform Matrix Factorization, P and Q vectors are initialized. This is done by fixing a value of k, thus determining the dimensions of P and Q vectors.

TABLE I
Fixed Hyperparameters used for Matrix factorization

λ	k	α	(m,n)	Epochs
0.4	3	0.01	(34947,1449)	30

The dimensions of P are (k,m) Q are (k,n) and of prediction matrix will be (34947, 1449). With default hyperparameters mentioned in Table 1, Simon Funk's popularized SGD is used to loop 30 times through all the ratings in the training set. For every item, the system predicts a rating and calculates the associated error (equation 3). It then modifies the P and Q vectors by learning rate α to go in the opposite direction of gradient descent till the model converges (equation 4).

$$e_{ui} \stackrel{\text{def}}{=} r_{ui} - p_u^T \cdot\ q_i \tag{3}$$

$$\begin{aligned} q_i &\leftarrow q_i + \gamma \cdot (e_{ui} \cdot p_u - \lambda \cdot q_i) \\ p_u &\leftarrow p_u + \gamma \cdot (e_{ui} \cdot q_i - \lambda \cdot p_u) \end{aligned} \tag{4}$$

On plotting the training and validation error, as seen in Figure 1, for k=3 the model converges somewhere between 15 to 20 epochs.

B. Sensitivity Analysis

1) Experiment 1 k = [3,8,16,64], for all other constant parameters: Ideally, we want a huge number of k to capture all the signals. But this causes the test data to rise, data becomes noisy and the model overfits, performing poorly [8]. The model was able to converge for values of k=3,8,16 displaying

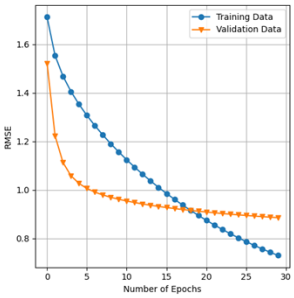

Fig. 1. Training vs Validation error plot with default hyperparameters.

TABLE II
3 EXPERIMENTS WITH DIFFERENT TWEAKED PARAMETERS

Experiment No.	λ	k	α	Epochs
1	0.4	3,8,16,64	0.01	30
2	0.1,0.2,0.3,0.4,0.5,0.6	64	0.01	30
3	0.02	3,8,16,64	0.001	100

a significant decrease in performance but it did not converge for k=64 because the difference between the training error and validation error got so huge that the model showed an error as values reached infinity.

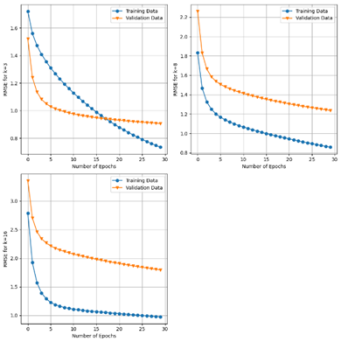

Fig. 2. Training vs Validation error plots for k=3, 16, 64 with default hyperparameters.

2) Experiment 2 k=64, λ∈(0.1,0.6): Keeping k constant, λ was varied to see if the model was able to regularize well and converge for k=64. But for no value of λ in the desired bracket the model produced successful convergence. Hence, for higher values of k, λ had no effect.

3) Experiment 3: Simon Funk's proposed parameters: α=0.001, λ=0.02 with epochs = 100 [3]: Using the above values, k=64 was able to converge but with a very high error. On a rather unique side, k=8,16 performed much better showing significantly improved convergence with minimized RMSE. At the same time, k=3 was underfitting, and k=16 was overfitting.

III. COLD START PROBLEM

Cold start problem is faced by both new users and items. Noticed specifically in Collaborative Filtering, one way of

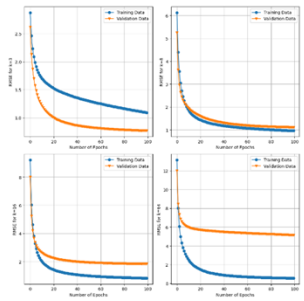

Fig. 3. Training vs Validation error plots for k=3, 8, 16, 64 with Simon Funk's proposed parameters.

solving this is by using Content-Based filtering [1][2]. If attributes of items are available, the Content-Based method can be used to produce ratings for extremely new items [2]. Another way is by generating implicit ratings. By collecting data from a user's purchase history or browser history, a model can understand the user's behaviour and thus yield recommended items [1]. Many researchers have also suggested a hybrid method, where combining Collaborative and Content-Based filtering i.e additional user information such as gender, age, geographic location etc and of items such as genres, product categories, keywords etc, and mapping them with latent features of a matrix [5] yields useful results. This way the latent factors of a matrix factorization model can be applied to any new user-item pair.

IV. CONCLUSION AND FUTURE WORK

With an increase in the dimension of the prediction matrix (k), the performance of the prediction model decreases significantly. For higher values of k, λ has no effect but significant changes in α allow convergence. Additionally, issues like the cold start problem can be catered to with multiple techniques. Future analysis can be done by tweaking α or using a new optimization technique.

REFERENCES

[1] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," Computer, vol. 42, no. 8, pp. 30–37, Aug. 2009.
[2] P. Melville and V. Sindhwani, "Recommender Systems." Accessed: Feb. 09, 2023. [Online].
[3] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," 2007. Accessed: Feb. 09, 2023. [Online].
[4] E. Rosenthal, "Intro to Recommender Systems: Collaborative Filtering — Ethan Rosenthal," www.ethanrosenthal.com.
[5] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. Schmidt-Thieme, "Learning Attribute-to-Feature Mappings for Cold-Start Recommendations," IEEE Xplore, Dec. 01, 2010.
[6] F. Kühn, M. Lichters, and N. Krey, "The touchy issue of produce: Need for touch in online grocery retailing," Journal of Business Research, vol. 117, pp. 244–255, Sep. 2020.
[7] X. N. Lam, T. Vu, T. D. Le, and A. D. Duong, "Addressing cold-start problem in recommendation systems," Proceedings of the 2nd international conference on Ubiquitous information management and communication - ICUIMC '08, 2008.
[8] "Lecture 56 - Finding the Latent Factors — Stanford University," www.youtube.com. https://www.youtube.com/watch?v=GGWBMg0i9d4 (accessed Feb. 10, 2023).