Machine Learning/ Deep Learning based Recommendation System

Dr. Gopal Pardesi[1], Nikita Pardeshi[2]

[1]*Associate Professor, Department of IT, Thadomal Shahani Engineering College, Mumbai, India*
[2]*Final-year B.E.-EXTC, Thadomal Shahani Engineering College, Mumbai, India*

---***---

Abstract - *This paper is aimed at briefly describing filtering- based recommendation system and its different sub-categories. Many popular E-commerce sites have been using recommendation systems to recommends music, movies, news, articles, books and other products. This paper attempts to describe complete construction of one such system using Weighted Alternating Least Squares method.*

Key Words*: *Collaborative Filtering, Hybrid based recommendation system, Hyperparameters, WALS method

1.INTRODUCTION

Web services such as Amazon, Netflix, YouTube and others are constantly advertising their products and helping their customers by suggesting the best items that could interest them or match their preferences. This task is performed by Recommendation systems. Recommendation systems play an important role in many companies and generate a huge amount of revenue. Unique recommendation systems also provide competitive advantage in the market. [1]

Recommendation systems can be divided into three categories: Collaborative filtering, Content-based filtering and Hybrid system. Collaborative filtering predicts items by taking into account the past behavior of a user; history of purchased items, ratings given to those items, as well as similar decisions made by other users. Content-based filtering uses pre-tagged characteristics of an item to recommend additional items with similar properties. Hybrid based systems are a combination of both the above methods. [3]

2. FILTERING METHODS

2.1 COLLABORATIVE FILTERING

Collaborative filtering is based on the assumption that the users will like similar kind of items that they liked in the past. Peer users or items with a purchase/ rating history similar to the current user or item are located and recommendations are made using this neighborhood [2]. These past interactions are stored in so-called 'user-item interaction matrix'. The more users interact with items, the more accurate recommendations will be made. [4]

The most important advantage of Collaborative filtering is that it does not require an 'understanding' of the item and hence, can recommend complex items. However random the combination of items or users be, the collaborative filtering approach works does not rely on what the content is [4]. Algorithms such as k-Nearest Neighbor (k-NN) and Pearson Correlation can be used for measuring similarity between items or users.

Problem arises while implementing this approach under three circumstances:

- When there isn't enough data for a new user/item i.e. cold start problem, under which random items are recommended to new users or vice versa
- If computation power available is less to calculate recommendations for millions of users or items
- If only a few numbers of user ratings are available for an item/items [3]

Collaborative filtering algorithm is further divided into sub-categories: Memory based and Model based approaches. Memory based approach are based on nearest neighbor search and directly work with values of recorded interactions, without assuming any model. Model based approach, as the name suggests, assumes an underlying model that explains and tries to discover user-item interactions to make predictions [4]. These concepts shall be discussed further in detail later in this paper.

2.2 CONTENT BASED FILTERING

Content-based filtering methods are based on the description of an item (set of discrete attributes and features) and a profile of the user's preferences. Keywords are used to describe the items and a user's profile is built indicating the kind/ type of items liked by the user. Hence, these algorithms try to match and recommend items that are similar to those liked by the user in the past, or is examining in the present [2]. In particular, various candidate items are compared with the items rated by the user in the past and the best-matched items are recommended.

An item presentation algorithm is applied to extract features of all items in the system. A widely used algorithms is the tf-idf (Term Frequency and Inverse Document Frequency) representation, also called vector space representation [4]. Based on weighted vector of item features, content-based user profiles are created where, weights represent the importance of each feature to the corresponding user. These weights can be computed using simple approaches such as the average values of the rated item vector. Machine learning techniques such as Bayesian Classifiers, Decision Trees, Cluster Analysis and Artificial Neural Networks (ANNs) can be used to calculate the probability that the user is going to like the item. [5]

Problems arise when the system is limited to recommending content of the same type as already used by the user. Under such cases, the value of the recommendation system is significantly less as compared to recommending other content types from different services. For example, news article recommendations based on the user's browsing of news is useful, but would be much more useful and interesting when products, music, videos etc. from different services can be recommended using the same news browsing history of the user [3]. Hence, problems of both, Collaborative filtering and Content-based filtering are overcome by Hybrid systems.

Although, Content-based methods are not that affected by the problem of cold start since, new users/ items can be described by their characteristics and relevant recommendations can be made. [7]

2.3 HYBRID RECOMMENDER SYSTEMS

Nowadays, most recommendation systems use the hybrid approach i.e. a combination of Collaborative Filtering and Content-based Filtering. Hybrid systems can be built by making content-based and collaborative-based predictions separately and then combining them or by adding collaborative-based approach to content-based model etc. Hybrid models have proved their worth time and again by providing the best recommendations since, these systems overcome the problems of collaborative and content-based approaches of cold start or sparsity. [7]

Netflix is a good example of the use of hybrid recommender systems. The website makes recommendations by comparing the watching and searching habits of similar users (i.e., collaborative filtering) as well as by offering movies that share characteristics with films that a user has rated highly (content-based filtering) [6].

3. COLLABORATIVE FILTERING

3.1 MEMORY BASED

In memory based collaborative methods, no latent model is assumed. The algorithms directly work with the user-item interactions where the users are represented by their interactions with the items. Then nearest neighbor search algorithms are implemented on these representations to provide new recommendations. A drawback of memory based collaborative methods is that these methods have a low bias, but very high variance since no model is assumed. [3]

3.2 MODEL BASED

As the name suggests, model based collaborative methods assume a latent model which is then trained to reconstruct user-item interactions values [8]. Since a model for user-item interaction is assumed, these methods theoretically, have a higher bias but a lower variance.

3.3 MATRIX FACTORIZATION

Matrix factorization algorithms for model based collaborative filtering decomposed the huge and sparse user-item interaction matrix into a product of two smaller and dense matrices called 'user-factor' matrix containing user representations that multiplies a factor-item matrix containing items representation. [4]

Matrix factorization assumes that there exists low dimensional latent space of features wherein both users and items can be represented, such that the interaction between them can be obtained by calculating the dot product of corresponding vectors in space [10].

Rather than providing these features explicitly to our model, we let the model discover these features by itself and make its own user and item representations. It is usually observed that close users in terms of preferences, as well as close items in terms of characteristics end up having close representation in the latent space. [4]

For better understanding, a classical approach is described based on gradient descent used to obtain factorizations for very large matrices.

Consider a ratings interaction matrix M (i*j) where only some items have ratings given by users. Our aim is to factorize the matrix such that

$$M \approx U * I^T$$

where U is the user matrix (i*d) whose rows represents i users

and I is the item matrix (j*d) whose rows represent j items

Here, d is the dimension of latent space in which users and items will be represented. Each row corresponds to a unique user and each column corresponds to a unique item. Each entry in the matrix is the user's rating or preference for a single item. We look for matrices U and I whose dot product best approximates the existing interactions i.e. we find U and I such that the 'rating reconstruction error' is minimized. Once the matrix is factorized, we need not manipulate a lot of information to make new recommendations, we can simply multiple a user vector with any item vector and get the corresponding rating. We can use the nearest neighbor search method too, but only for small, dense matrices for accurate results.

The matrix factorization method assumes that there is a set of attributes common to all items and each user has its own expression for each of these attributes, independent of the items. These attributes are called hidden or latent factors and can be very useful for many types of recommendation systems. [5]

3.4 TRANSFORMATION OF INTERACTION MATRIX TO REPRESENT LATENT FACTORS

of size 'm' and set of items Y of size 'n' and choose an arbitrary number 'k' representing latent factors. We perform factorization on the larger matrix M (i*j) into two smaller matrices, say A (row factor) and B (column factor). [4]

Matrix A has dimensions m*k and B has dimensions k*n. The sparse information on matrix M is compressed into much lower dimensional spaces m*k and k*n. Matrices A and B are multiplied to get M` which is an approximation of the larger matrix M. Each row in matrix A corresponds to the strength of the user's preferences for the k latent factors and each column in matrix B corresponds to the item's expression of the same k latent factors. Hence, to calculate the user's (X) rating for item Y we take the dot product of the two vectors:

$$r=A_m^T * B_n$$

This dot product is a real number and represents the prediction of user's rating for an item n. The loss function for measuring the accuracy of this prediction is given by:

$$LF= \Sigma_{m,n} (r- A_m^T * B_n)^2$$

To prevent overfitting, regularization terms are added in the loss function.

$$LF= \Sigma_{m,n} (r- A_m^T * B_n)^2 + \lambda \Sigma_m ||A_m||^2 + \lambda \Sigma_n ||B_n||^2$$

4. WALS METHOD

The weighted alternating least squares method introduces different weights for zero, unobserved or non-zero entries in the matrix.

$$L^w= W^* \Sigma_{m,n} (r- A_m^T * B_n)^2$$

Here,

$w_{mn}= w_0$; for zero (unobserved) entries in the ratings matrix and

$w_{mn} =w_0 + f(c_i)$; for observed entries where

$c_i= \Sigma_{m,n}$ 1 if r>0; the sum of the number of non-zero entries for column n

The weights are scaled by the sum of non-zero entries in a row to normalize the weights for users who have rated a different number of items. This type of weighting method yields better empirical results. [9]

5. HYPERPARAMETERS

Hyperparameters are the variables which determines how the network is trained and governs the entire training process [4]. These parameters are set before training. As the model is being trained, the training application handles three different kinds of data:

- Input data- Also called as training data, is a collection of records containing features that are important to the machine learning model.
- Model's parameters- defines the variables that the machine learning model uses to adjust to your data set. These parameters set your model apart from other models of the same type working on similar data.
- Hyperparameters- These are configuration variables and remain constant during any job as compared to other parameters which change during a training job.

Hyperparameters are tuned by running your whole training job, looking at the aggregate accuracy, and adjusting.

Hyperparameters are divided into two types: Default hyperparameters and Tuned hyperparameters.

5.1 TUNED HYPERPARAMETERS

For tuning hyperparameters, we must define the variables that we want to adjust and a target value for each hyperparameter. You also define the hyperparameter type and the range of values to try. Tuning optimizes a single target variable, also called the hyperparameter metric that we specify. This variable has to be specified before the job starts.

Finding the optimal set of hyperparameters is very critical as this dictates the performance of the machine learning model. Experimenting with these parameter values, reasonable ranges and then testing the model's performance based on these values can be very time consuming and forces you to assume these parameters. But there are various platforms available which automatically searches for an optimal set of parameters for the hyperparameters that you want to tune, as well as their expected values and ranges. [10]

6. RESULTS

WALS method is implemented on the MovieLens data set (1m and 20m) using both default and tuned hyperparameters. RMSE is calculated as an accuracy measure. Root mean square error (RMSE) measures the difference between the values predicted by the model and the values observed [10]. The lower the RMSE, the better the model. The following results were obtained:

Table 1- RMSE results

Data set	RMSE with default hyperparameters	RMSE with tuned hyperparameters
1m	1.2	0.92
20m	1.35	0.89

7. CONCLUSION

WALS method is an effective algorithm for building better recommendation systems since it includes optimizations making it easy to incorporate weights and effectively calculate row and column factor updates. This weighted algorithm can be used in data sets with both, implicit and explicit features. WALS method is capable of handling and processing large matrices with millions of rows.

The results obtained with default parameters is less accurate than those obtained using the tuned parameters, as shown in Table 1.

REFERENCES

[1] Silvana Aciar, Debbier Zhang, Simeon Simoff, John Debenham, 'Recommender System Based on Consumer Product Reviews', 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings) (WI'06), 15 January 2007

[2] Sung- Woo Byun, So-Min Lee, Seok-Pi Lee, Kwang-Yong Kim, Cho Kee-Seong, 'A Recommendation System Based on Object of the Interest', 2016 18th International Conference on Advanced Communication Technology (ICACT), 03 March 2016

[3] Kunal Shah, Akshaykumar Salunke, Saurabh Donagre, Kisandas Antala, 'Recommender systems: An overview of different approaches to recommendations', 2017 Internaltional Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), 01 February, 2018

[4] Francesco Ricci, Lio Rokach, Bracha Shapira, Paul B. Kantor, Recommendation Systems Handbook, Springer, 2010

[5] J.S. Breese, D. Heckerman, C. Kadie, 'Empirical analysis of predictive algorithms for collaborative filtering', Proc. Of the 14th Annual Conference on Uncertainty in Artificial Intelligence, pp. 4352, 1998

[6] M. Deshpande, G. Karypis, 'Item-based top-N recommendation algorithms', ACM Transaction on Information Systems, vol. 22, no. 1. Pp. 143177, 2004

[7] J. Karim, '2014 Hybrid Systems for Personalized Recommendation Research Challenges in Information Science (RCIS)', 2014 IEEE Eighth International Conference, May 2014

[8] G. Linden, B. Smith, J. York, 'Amazon.com recommendations item-to-item collaborative filtering', IEEE Internet Computing, vol. 7, pp. 76-80, January 2003

[9] Yifan Hu, Yehuda Koren, Chris Volinsky, 'Collaborative Filtering for Implicit Feedback Datasets'

[10] Bhumika Bhatt, Prof. Premal J Patel, Prof. Hetal Gaudani, 'A Review Paper on Machine Learning Based Recommendation System', 2014 IJEDR, Volume 2, Issue 4, ISSN: 2321-9939