

# PREFERENCE WEIGHTED RESTRICTED BOLTZMANN MACHINES FOR COLD START RECOMMENDATIONS

G.S. Ananth<sup>1</sup>, K. Raghuveer<sup>2</sup>, and S. Vasanth Kumar<sup>3</sup>

<sup>1</sup>Department of MCA, The National Institute of Engineering, India

<sup>2</sup>Department of Information Science and Engineering, The National Institute of Engineering, India

<sup>3</sup>Department of Mathematics, Mangalore Institute of Technology and Engineering, India

## Abstract

*Collaborative filtering and content filtering are the two techniques used in filtering data for Recommender Systems. One of the not fully resolved problems of Collaborative filtering is the cold start problem and is still an active research topic in the area of Recommender Systems. The cold-start problem of items is still a great research topic to work on. With the advent of Machine Learning there seems to be some success though; different models and algorithms with diversified approaches are emerging. In this work, we make use of Boltzmann machines and specifically a Restricted Boltzmann machine (RBM) with dynamic data to solve the cold start problem. We propose a collaborative filtered model with RBM and call it CF\_RBM. The active and passive browsing history and the time spent on a website are taken as dynamic data and passed along with the other visible units of the RBM. The user preferences of nearly 100 random users of the system was also included. We consider the MovieLens 10M dataset and apply the CF\_RBM model. The recommendation results obtained along with the recommendation score shows that the new items are generated, hence resolving the cold-start problem.*

## Keywords:

*Recommender Systems, Intelligent Systems, Decision Support Systems, Collaborative Filtering, Boltzmann Machines, User Preferences, Passive Browsing History*

## 1. INTRODUCTION

Recommender Systems (RS) are Intelligent Systems that help towards decision making. They are also sometimes called Decision Support Systems (DSS). RSs help towards obtaining recommendations and getting results of many times from unknown values and scenarios. An example is purchasing a strange and unique product from an e-commerce website that has never been purchased before. But can RSs help in obtaining such kind of recommendations is the overall problem. Or a scenario where a newly introduced product, can it get sales and move up the sales ladder is a good example. RSs involves majorly three filtering techniques of filtering data.

Note the data could be both dynamic and static in nature. Manipulating and filtering data that is static in nature is considered relatively easier than the data that is dynamic in nature. Collaborative filtering was the very first technique used, then content filtering and now in trend is the hybrid filtering technique which makes use of both content and collaborative filtering mechanisms.

Examples of RSs can be seen for Movie Recommendations, Music Recommendations, Book Recommendations, News Article Recommendations, Product Recommendations and as well Friend Recommendations. Companies like Netflix, Amazon, Facebook, Yelp, Flipkart, Spotify make use of Recommender Systems. But

this paper specifically targets the generation of cold start recommendations.

## 1.1 CHALLENGES WITH COLLABORATIVE FILTERING

Considered to be the oldest form of RS filtering mechanisms, Collaborative filtering is sub-classified into 3 types of algorithms: User-User, Item-Item and User-Item filtering under the non-parametric models.

User-user is more commonly also termed “people like you - like that, like this” situation, Item-Item is commonly termed “if a user likes this, he could also like that item” and finally the User-Item combines both the above approaches and its an embedding overall. There are a few challenges with collaborative filtering. They are: Cold-start problem, Sparsity problem and Scalability issues.

Again, there are basically two kinds of cold-start problems. First is the cold-start of users and the second is the cold-start of items. Cold-start of users is a scenario where the RS algorithm finds it difficult to provide items of user’s liking or choice due to lack of user preferences. Whereas the cold-start of items is when and how to select a new item and place it into as per the user’s preference.

The sparsity problem could be solved by prefixing and obtaining user-preferences as an input to the RS. Parametric models or models that do not rely on the neighbourhood-memory have resolved the issues of scalability to a certain extent [2].

## 2. RELATED WORK

Zhou et al. [9] revealed that Matrix factorization concept could help in recommendation of cold-start items. They use low-rank factorization method first, applying regularization and alternating optimization to create the cold-start setting. Then, a functional matrix factorization concept is applied where in if a new user enters the system, a series of questions are asked to the new user and then the profile is created - hence capturing the user preferences for the system.

Hazrati et al. [5] show the use of visual features are used for cold-recommendation of video items. The novel features from the video are extracted without the influence of human-annotation, by analyzing features of the video - clip by clip or rather frame by frame in detail. This work also encountered the new item problem in 3 scenarios - extreme cold start, moderate cold start and warm start.

Gunawardana et al. [3] makes use of a Tied Boltzmann machine for cold start recommendations. This paper generated

recommendations by predicting a user’s vote on an item given his vote on other items by other users. Using the tied Boltzmann machine model for a training-set of logistic regression classifiers with a very particular input parametrization and parameter tying scheme to predict individual votes.

Wei et al. [7] use the content features extracted from content descriptions like movie plots and by making use of Deep Learning models. Through the neural networks, the content features are used to determine the item similarity and also become the key component of the recommendation models.

Zhang et al. [8] expresses the different deep learning-based models for obtaining recommendations. The survey speaks of a RBM and about its characteristics - deriving the meaning of a visible and hidden layer.

### 3. BOLTZMANN MACHINE

Invented by Geoffrey Hinton, a Boltzmann machine is an unsupervised model of machine learning that involves learning a probability distribution from a dataset and then predicts or infers learning about not seen data in the dataset. Typically, a Boltzmann machine has one input layer and one or multiple output layers.

The input layer is also termed as the visible unit and the hidden layer is termed as a hidden unit. Though Boltzmann machines run on the concept of a neuron on neural networks, every neuron on the network across both the layers are bidirectionally connected with each other.

From the viewpoint of the Boltzmann machine, it sees no visible or hidden units, and both are one and the same, generating information from the dataset – even for an instance with no visible units [1].

The blue circles show the visible nodes and the hidden nodes are illustrated through the red circles. The Boltzmann machine is a stochastic recurrent neural network.

#### 3.1 RESTRICTED BOLTZMANN MACHINES

A Restricted Boltzmann machine is a variant of the Boltzmann machine, invented by Paul Smolensky. The term “Restricted” is used for RBM due to the fact that no two nodes on the same layer are interconnected. Restricted Boltzmann machines have a wide range of applications like Topic Modelling, Feature Learning to Regression, Classification and also Collaborative filtering.

The RBM works with 2-phases and in 3-phases. A first phase forward passes where the input given to the visible unit is combined with weights (typically one weight) and one bias whose result is passed to the hidden unit – but may or may not be activated with any activation function. The second phase backward pass, the activation from the activation function is combined again with one or more weights with an overall bias and the result is passed to the input for generation and construction. In the last phase, the construction also called the reconstructed output of the second phase is compared with the originality of the input to see if any betterment of results is obtained.

Here, a weight is a connection point between the visible and hidden units. Likewise, a bias helps an activation function to either traverse across the passes bidirectional. There are also

multiple activation functions for a RBM – sigmoid, relu, tanh amongst others.

#### 3.2 RBM IN COLLABORATIVE FILTERING

A Restricted Boltzmann machine is used in this paper along with the addition of cumulative weights from the user preferences and captured browsing history with time spent on the webpage. A trained RBM can be used to discover the different characteristics which are the most significant ones during different designs of the neural network. A completed prepared undirected network will can predict better hidden values.

A Restricted Boltzmann machine works in two stages: Gibbs sampling and contrastive divergence stage. They permit the RBM to interpret the interrelationships among the information highlights, and they likewise help the RBM choose which info highlights are the most significant when distinguishing designs are available [6]. When distinguishing new designs are available, this is when new items are generated, hence solving the cold start of items. Every weight and its bias play an important role towards generating new patterns of output prediction. The Fig.4 depicts an example with 3 hidden units and 4 visible units.

### 4. DATASET AND DESCRIPTION

For the working of the CF\_RBM model, we made use of the MovieLens 10M dataset from Grouplens [4]. The Grouplens dataset has 10000054 movie ratings. There are overall 10,681 movies. The 71,657 users selected have rated at least 20 movies. No demographic data is available in this dataset.

There are three .dat files that contain the complete data. A 5-fold cross-validation of the data is applied in this dataset. The Table.1 provides its description. All the attributes of the MovieLens 10M dataset is mentioned. This dataset was released in the year 2009, contains 18 types of genres. The dataset is a stable benchmark and actively used in research related to Collaborative Filtering and Recommender systems.

Table.1. Description of dataset

Name of the Dataset	MovieLens 10M
Owned by	GroupLens
Category	Movies
Number of ratings	10000054
Number of movies	10, 681
Number of users	71, 567
Number of tags	95, 580
Released in	2009
Number of Genre Types	18
Number of movies at least rated	Min 20
Demographic data	No
Cross-validation	Yes, 5-fold
Number of dat files	03

## 5. CF\_RBM MODEL

The CF\_RBM Model is built on a RBM with user preferences, capturing browsing history as input layers along with other data.

### 5.1 CONSTRUCTION OF USER\_PREFERENCES TABLE

For the construction of the user\_preferences table, we took inputs from 100 random users asking them questions and converting their answers to a binary value. A Yes answer would yield to a binary value as 1 and No would yield 0. The Table.2 describes few of the preference questions asked to users.

Table.2. Preference questions asked to users

Preference Questions
Do you like the movie of a specific actor/director/producer
All ShahRukhKhan movies are good
All Horror movies are excellent. Do you agree?
Would you like to watch a Romance movie when it's raining outside?
Steven Spielberg needs no introduction. Have you seen all his movies?
Are all the Batman series movies excellent?

Overall, a set of 40 such questions were asked for every user. The results were captured. The overall weighted average for every individual user was computed. For simplicity's sake, a 0.5 value was sealed.

### 5.2 CAPTURING BROWSING HISTORY AND TIME SPENT ON THE WEBSITE

We installed a python pip package called browser history [10]. This would provide the various links visited by a user. The browser history package works for Google Chrome, Mozilla Firefox and Safari browser.

For the list of links obtained from the browserhistory, we enabled the TimeYourWeb Google Chrome JS extension [11] which would output the amount of time spent by a user on a specific site. The Table.3 shows the fields of the output of TimeYourWeb. The output of browserhistory matches the TimeYourWeb output.

Table.3. TimeYourWeb

Domain or website link	Example - www.google.co.in	
Date	System date and Time	
Time	Total Time Spent	Active Time Spent
	In ms	In ms
Title of the link	Page opened in the domain	

### 5.3 TRAINING OF RBM

The two main training steps involved for the RBM is Gibbs Sampling and Contrastive Divergence step. For given an input vector  $v$  (which is visible), the prediction of the hidden values  $h$ , is computed as in Eq.(1):

$$p(h_i = 1 | v) = \frac{1}{1 + e^{-(b_i + \sum_{n=1}^j w_{ij} v_n)}} = \sigma\left(b_j + \sum_{n=1}^j w_{ij}\right) \quad (1)$$

The updation of the weights matrix happens during the Contrastive Divergence step and is computed as in Eq.(2).

$$p(v_i = 1 | h) = \frac{1}{1 + e^{-(b_i + \sum_{n=1}^j h_n w_{ij})}} = \sigma\left(a_j + \sum_{n=1}^j h_n w_{ij}\right) \quad (2)$$

Finally, the update matrix is the gradient ascent given by Eq.(3).

$$W_{new} = W_{old} + \Delta W \quad (3)$$

### 5.4 ADDING WEIGTHS TO THE RBM DURING CONTRASTIVE DIVERGENCE

The cumulative weight of the user preferences along with browsing history per page is added to the contrastive divergence training phase as given by the Eq.(4). The algorithm 1 explains the n-step contrastive divergence with these added weights.

$$W_{ij} = \sum_{n=1}^j W_n + \sum_{n=1}^j \{BH\}_{pp} \quad (4)$$

#### Algorithm 1: n-step Contrastive Divergence with weighted CF\_RBM

**Input:** CFRBM ( $V_1, \dots, V_m, H_1, \dots, H_m$ ), training batch  $S$

**Output:** Weighted approximation  $\Delta w_{ij}$ ,  $\Delta b_j$  and  $\Delta c_i$  for  $i=1..n$ ,  $j=1..m$ .

**init**  $\Delta w_{ij} = \Delta b_j = \Delta c_i = 0$  for  $i=1..n, j=1..m$ .

**for all**  $v \in S$  **do**

$v^{(0)} \leftarrow v$

**for**  $t = 0 \dots k-1$  **do**

**for**  $i=1 \dots n$  **do**

sample  $h_i^{(t)} \approx p(h_i | v^{(t)})$

**for**  $j=1 \dots m$  **do**

sample  $v_j^{(t+1)} \approx p(v_j | h^{(t)})$

**for**  $i=1 \dots n, j=1 \dots m$  **do**

$$\Delta w_{ij} \leftarrow \Delta w_{ij} + p(H_i | v^0) \cdot v_j^0 - p(H_i = 1 | v^k) \cdot v_j^k$$

$$\Delta b_j \leftarrow \Delta b_j + v_0^{(j)} - v_j^{(k)}$$

$$\Delta c_i \leftarrow \Delta c_i + P(H_i | v^0) - P(H_i = 1 | v^k)$$

$$\Delta c_i \leftarrow \Delta c_i + \Delta w_{ij}$$

**done**

### 5.5 IMPLEMENTATION OF CF\_RBM MODEL

The CF\_RBM model was implemented using Tensorflow(TF) 1.14 on a python environment. For creating the TF model, a conda environment was created since the latest version of TF is > 2.0. The user preferences captured along with the Browsing history were added to the weight as a neural tensor.

### 6. RESULTS AND DISCUSSION

The CF\_RBM model was run across for various users picked at random. The model could run successfully and demonstrate that new items as in movies were portrayed. Few of these movies, though they are popular movies, were not watched by many users. The Fig.1 depicts the error per number of epochs. We could observe that as the number of epochs increases with batch size, the error decreases. The error is minimal and is almost near to zero at the end of the run of the batch size and with epoch equal to 15. The count of the epoch if increased, then we observe that the error is very much a zero value.

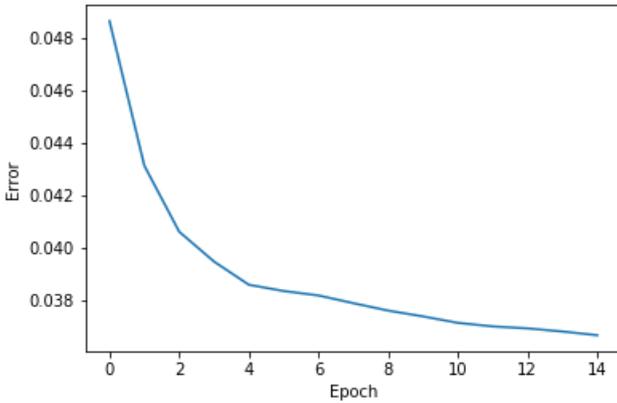


Fig.1. Error per epoch

The Table.4 shows Movie titles whose recommendation score is nearing 90%.

Table.4. Predicted movies for user with userid 85

MovieId	Title	Genres	Rec. Score
318	Shawshank Redemption	Drama	0.899200
608	Fargo	Crime Drama Thriller	0.879345
2858	American Beauty	Comedy Drama	0.809069
593	Silence of the lambs	Drama Thriller	0.783956
3160	Magnolia	Drama	0.765205
1704	Good Will Hunting	Drama	0.701307
1179	The Grifters	Crime Drama Film-Noir	0.685976
2395	Rushmore	Comedy	0.664542
296	Pulp Fiction	Crime Drama	0.660926

2997	Being John Malkovich	Comedy	0.652990
------	----------------------	--------	----------

Table.5. Predicted movies for user with userid 215

MovieId	Title	Genres	Rec. Score
260	Star Wars: Ep. 4	Action Adventure Fantasy Sci-Fi	0.862678
1196	Star Wars: Ep. 5	Action Adventure Drama Sci-Fi War	0.806127
1210	Star Wars: Ep. 6	Action Adventure Drama Sci-Fi	0.778966
2571	Matrix	Action Sci-Fi Thriller	0.773782
2628	Star Wars: Ep. 1	Action Adventure Fantasy Sci-Fi	0.766304
480	Jurassic Park	ActionAdventure Sci-Fi	0.726725
1580	Men in Black	Action Adventure Comedy Sci-Fi	0.692906
1356	Star Trek	Action Adventure Sci-Fi	0.657311
589	Terminator 2	Action Sci-Fi Thriller	0.643831
1097	E.T. The Extra Terrestrial	Children's Drama Fantasy Sci-Fi	0.640288
1270	Back to the Future	Comedy Sci-Fi	0.635573
1	Toy Story	Animation Children's Comedy	0.597626

### 7. CONCLUSION AND FUTURE WORK

In our work we introduced two extra parameters; the passive browsing history, time spent on websites along with user preferences as extra inputs to the Restricted Boltzmann machine for generating collaborative filtering.

We observed that the overall Recommendation score of the CF\_RBM model increased and in a few cases the score was near to 90%. This is a significant increase for the betterment of predictions. Even when a Top\_N recommendation is obtained, the most popular movie of this dataset is the Toy Story movie and is the most recommended. But the results show that the movie title is almost predicted after more than 10 recommended movies. The results show that the titles that are not extremely popular movies. This justifies the fact that the new movie titles are recommended to users; hence also resolving the cold-start problem, specifically the cold-start problem of items.

The Restricted Boltzmann Machines have shown various machine learning applications. Collaborative filtering features can be tweaked to obtain better predictions. If the model parameters can be changed and adjusted and along with the addition of various other parameters; the problem of cold start – both in users and items can be resolved. Also, parameters like Demographic attributes; which are not present in the dataset and more detailed user preferences can be included for the next variants of the model.

## REFERENCES

- [1] David H. Ackley, Geoffrey E. Hinton and Terrence J. Sejnowski, "A Learning Algorithm for Boltzmann Machines", *Cognitive Science*, Vol. 9, No. 1, pp. 147-169, 1985.
- [2] Ricci Francissco, "*Recommenders Systems Handbook*", 2<sup>nd</sup> Edition, 2015.
- [3] Asela Gunawardana and Christopher Meek, "Tied Boltzmann Machines for Cold Start Recommendations", *Proceedings of ACM Conference on Recommender Systems*, pp. 1-13, 2008.
- [4] F. Maxwell Harper and Joseph A. Konstan, "The MovieLens Datasets: History and Context", *ACM Transactions on Interactive Intelligent Systems*, Vol. 5, No. 4, pp. 1-12, 2015.
- [5] Naieme Hazrati and Mehdi Elahi, "Addressing the New Item Problem in Video Recommender Systems by Incorporation of Visual Features with Restricted Boltzmann Machines", *Expert Systems*, Vol. 38, pp. 1-10, 2020.
- [6] Ruslan Salakhutdinov, Andriy Mnih and Geoffrey Hinton, "Restricted Boltzmann Machines for Collaborative Filtering", *Proceedings of International Conference on Machine Learning*, pp. 79-85, 2007.
- [7] Jian Wei, Jianhua He, Kai Chen, Yi Zhou and Zuoyin Tang. "Collaborative Filtering and Deep Learning based Recommendation System for Cold Start Items", *Expert Systems and Applications*, Vol. 69, pp. 29-39, 2017.
- [8] Shuai Zhang, Lina Yao, Aixin Sun and Yi Tay, "Deep Learning based Recommender System", *ACM Computing Surveys*, Vol. 52, No. 1, pp. 1-38, 2019.
- [9] Ke Zhou, Shuang-Hong Yang and Hongyuan Zha, "Functional Matrix Factorizations for Cold-Start Recommendation", *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 31-39, 2011.
- [10] PYPI, "Browser-History", Available at <https://pypi.org/project/browser-history/>, Accessed at 2021.
- [11] Google Chrome, "Timeyourweb", Available at <https://chrome.google.com/webstore/detail/timeyourwebtime-tracker/>, Accessed at 2021.