Movie Recommendation Using Deep Learning with Hybrid Approach

Meet Gogri, Dharmil Chheda, Vinit Solani
1,2,3 Shah & Anchor Kutchhi Engineering College, Mumbai, Maharashtra
Email: meet.gogri@sakec.ac.in, dharmil.chheda@sakec.ac.in, vinit.solani@sakec.ac.in

Abstract
Proposal frameworks are a significant piece of recommending things particularly in gushing top administrations. For spilling film administrations like Netflix, proposal frameworks are fundamental for helping clients send new motion pictures to appreciate. Here, we propose a profound learning approach to deliver a community oriented separating framework which predicts film evaluations for a client dependent on an enormous database of appraisals from different clients. Utilizing the MovieLens dataset, we investigate the utilization of profound figuring out how to anticipate clients on new films, consequently empowering film Recommendations. The exploratory outcomes show that our proposal framework outflanks a client based system.

Keywords
Recommendation System, Deep Learning, Hybrid Approach, K-NN, MovieLens

Introduction
Recommendation Program is a slice of daily life in which people depend on knowledge to decide on decisions for their own purposes. Recommendation process is a subset of data sifting to forecast the reaction of consumers to the items used. Such demands present a few challenges so that various methodologies such as memory-based, model-based approaches are used. Given all, the proposed system wants progress to turn out to be great. Recommendation structure is a sharp system that offers clients think on stuff that could impress them with a few examples including amazon.com, movie photos in Movielens. In their protocols, the different methodologies are consulted in order to understand the constraints of each approach in a valid manner in order to offer acceptable potential proposals.

Review of Literature
In paper [5] the author has tried to design a scalable collaborative filtering by implementing it on MapReduce. Generally collaborative filtering can be implemented in three steps.
1. Construction of user-item matrix where number of users represents rows of matrix and the number of items represents the columns each entry in the matrix represents the corresponding rating value.
2. Calculate the similarity between the pair of items and the users and find their nearest neighbor.
3. Predict the unknown ratings for the items.
4. The above explained tasks are broken down into three map and three reduce jobs by the author. This module is common
5. For both user-based collaborative filtering and item-based collaborative filtering. Totally six modules are implemented to carry out the above mentioned task. They are group by user rating module Count ratings users frequent module, Pairwise items module, Calculate similarity module, Calculate ranking module and TopN similar items module.

Algorithms
A. Deep Learning
Profound production comes from studying artificial neural networks. Deep learning consolidates low-level highlights to slowly retrieve large-level representation classifications or highlights for discovering scattered knowledge portion portrayals [1,6] Profound learning is another field of AI Exploration This represents the human brain portion to decode information, such as pictures, sounds and messages. Unlike the AI approach, the deep learning technique is further separated between supervised learning and autonomous learning.

Fig. 1: Deep Learning Hidden Layers
B. RBM(Restricted Boltzman Machine)

There are a number of ways that the suggested structures can be installed. Many of them include techniques such as content-driven filtering, collaborative filtering driven on experience, model-based collaborative filtering, deep learning/neural networking, etc. [4] We'll concentrate on learning how to create a recommendation engine with Deep Learning. In this particular feature we can use the Restricted Boltzmann Machine (RBMs) algorithm. The main reasons for that are:

- RBMs have the ability to learn latent factors/variables (variables that are not explicitly accessible but can be derived from the variables present) from input data.
- RBMs are unsupervised learning algorithms that have the potential to recreate data input approximations. They achieve so by attempting to generate the probability distribution of input data with a reasonable estimate that helps to extract data points that did not appear in our data before.
- They achieve so by studying a lower-dimensional version of our data and then attempting to recreate the input using its representation.

The RBMs are made of an reference and a hidden layer. You are attempting to find a stochastic representation of the data. By sampling from the secret layer, you can replicate the types of samples found through testing. The RBM testing takes place by an intermediate sampling of the two layers, while back spread could also be used later to fine tune the pattern.

C. Hybrid Approach

Most of the preferred systems are currently using a half-breed methodology, incorporating mutual sifting, content-based separation, and various methodologies. There is no reason for why a few special approaches of a similar nature should not be hybridized. Half and half of the methodologies can be implemented in a variety of different ways: by rendering content-based and community-based predictions, separately and ultimately consolidated; by incorporating content-based skills to the Community-based approach (and the other way around) or by pooling the methodologies into one platform for the overall analysis of the advice frameworks [4, 5] A few studies, which contrast precisely the description of half and half with the unaltered mutual with substance-based approaches, have demonstrated that cross-breeding methods can allow more detailed recommendations than unaltered methodologies. Netflix is a true example of the application of cross-breeding rules. The platform makes recommendations by looking at the prediction and looking at the propensities of similar customers. (i.e., cooperative separating such as by providing motion pictures that share consistency with films that the company has extraordinarily measured. (content-based shifting).

Some hybridization techniques include:

- **Weighted**: Combining the score of various proposal segments numerically.
- **Switching**: Choosing between parts of ideas and adding the one chosen.
- **Mixed**: Guidelines from various guidelines are tabled together to include the advice.
- **Feature Combination**: Features gathered from various sources of knowledge are grouped together and given to a measurement of solitary suggestions calculation.
- **Feature Augmentation**: Computing an item or series of highlights that is then part of the contribution to the process below.
- **Cascade**: Recommenders are given an exact requirement, with lower needs breaking ties in higher ratings.
- **Meta-level**: A proposal process is used and a concept sort is generated, which is then the information used by the following procedure.

D. K-NN (K-Nearest Neighbor)

In pattern detection, a non-parametric approach used for classification and regression is the k-nearest neighbors algorithm (k-NN) [2] In both instances, the feedback consists of the nearest samples of training in the function space k. The performance depends on how the classification or regression uses k-NN

- The performance is a class affiliation in the k-NN group. An object is classified by a plurality vote of its neighbors, assigning the object to the most specific class of its closest k neighbors (k is a positive integer, typically small). If k = 1, then the object is strictly allocated to that neighbor's closest single class.
- In k-NN regression, the output is the value for the object. Then it is averages of the values of k nearest neighbors.

K-NN is a form of instance-based learning, or lazy learning, in which the function is only locally approximated and all computation is postponed until the function evaluation. A valuable method for both grouping and regression would be to apply weights to the contributions of the neighbors, so that the nearby neighbors add more to the average than the distant ones. Neighbors are taken from a group of objects about which the class (for classification k-NN) or the name of the entity property (for regression k-NN) are established. This can be viewed as the training set for the algorithm, although no specific training step is needed. The peculiarity of the algorithm k-NN is that it is reactive to the data's local structure.
1. We need data collection for implementing any algorithm. So during the first phase of KNN, we have to load the training as well as the test details.
2. Next, we need to pick the k value, i.e. the nearest data points. Any integer may be K.
3. For each point in the test data do the following –
   i. Using each of the methods namely: Euclidean, Manhattan or Hamming distance measure the distance between test data and each row of training data. Euclidean is the most commonly known tool for measuring space.
   ii. Now ordering them in ascending sequence, depending on the distance interest..
   iii. First, you pick the top K rows from the sort array.
   iv. Now, it will assign the class to the test point based on most frequent class of these rows.
4. End

E. Random Forest
Random forest is a type of machine-supervised learning algorithm based on learning the ensemble. Ensemble learning is a method of learning where you multiply combine different types of algorithms or the same algorithm to create a more efficient model of prediction. A random forest algorithm incorporates several algorithms of the same kind, i.e., several decision trees, resulting in a tree forest, thus the term "Random Forest." Random Forest algorithm can be used for both regression and classification tasks.
1. Pick N random records from the dataset.
2. Create a decision tree according to these N required
3. As required by user choose number of trees want in algorithm and repeat steps 1 and 2.
4. In case of a regression problem, for a new record, each tree in the forest predicts a value for Y (output).
The final value can be determined by taking the sum of all the predicted values from all forest trees. Or, each tree in the forest determines the division to which the new record belongs, in case of a classification question. Eventually, the latest record is allocated to the division where the largest vote is received.

Dataset Description
The most well-known Movie Ratings Dataset is undoubtedly the MovieLens dataset [2] So to test our Movie Recommendation system using Deep Learning with Hybrid content recommendation approach, we have used MovieLens Dataset (ml-100K). The dataset comprises individual ratings provided by users for a particular movie. This dataset contains of total 100004 ratings where, rating varies from 0 to 5, and has 9125 movies, given by 671 users. There are 20 genres available in MovieLens Dataset. The row shows users and the columns represent the movies leading to the creation of the matrix of user-item scores of 671 × 9125 user items rating matrix. The MovieLens dataset consists of the following slots userId, movieId, ratings, title, tags, and genres.

Results
Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Correlation are usually used as statistical accuracy metrics. MAE is the most popular and MAE is used commonly in many, it is a measure of deviation of recommendation from user’s actual value [4] the lower the RMSE and MAE, the more accurately and robust evaluation model the recommendation engine predicts user ratings. Such metrics are useful to use because estimating rank or number of transactions is based on the recommendations. They give us an idea of how reliable our predictive scores are and how reliable our recommendations are. RMSE and MAE are computed as follows:

$$RMSE = \sqrt{\frac{1}{N} \sum (predicted - actual)^2}$$

Fig. 2: RMSE Formula

$$MAE = \frac{1}{N} \sum |predicted - actual|$$

Fig. 3: MAE Formula
So far we have compared different Algorithms by applying MovieLens dataset (ml-100k) and split this data into two parts, i.e., 75% for Training and 25% for Testing.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>RMSE Value</th>
<th>MAE Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBM</td>
<td>1.1897</td>
<td>0.9935</td>
</tr>
<tr>
<td>K-NN</td>
<td>0.9961</td>
<td>0.7711</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.9946</td>
<td>0.7729</td>
</tr>
<tr>
<td>SVD</td>
<td>1.4366</td>
<td>1.1468</td>
</tr>
</tbody>
</table>
Conclusion and Future Scope
We have studied in detail various issues of recommendation system. We have studied many existing systems and found that they are not scalable enough. Hence, using Deep learning Algorithms like RBM, KNN, Random Forest to ensure good scalability and strength of the recommendation system. Also, this paper aims to achieve parallelization of association of different data mining with Hybrid approach to be able to handle big data. There is a need to work on the memory requirements of the proposed approach in the future. The proposed approach has been implemented here on different movie lens datasets only. It can also be implemented on the Film Affinity and Netflix datasets and the performance can be computed in the future.

References