Building an Effective Recommender System Using Machine Learning Based Framework

Ruchika[1], Ajay Vikram Singh[2], Mayank Sharma[3]

[1,2,3]Amity Institute of Information Technology, Amity University, Uttar Pradesh, Noida
[1]rbathla@amity.edu, bathla.ruchika@gmail.com, [2]avsingh1@amity.edu
[3]msharma22@amity.edu, mayanksharma28.in@gmail.com

***Abstract:* Machine learning forms the base of many information retrieval applications those effect our day to day lives directly or indirectly.One of the Commonly used application of machine learning algorithms is Recommender Systems. Recommender system are information flitering system which takes users rating for items into account and predict user preferences.Many online ecommerce and other categorical websites are able to generate recommendations either on the basis of implicit feedback or explicit feedback.In implicit feedback, preferences are actually based on analysis of browsing patterns of the user, for example, purchase history, web logs etc. Explicit feedback is generated from the ratings provided by the user.In this paper we have shown adaption of collaborative filtering in Apache Mahout platforms via Eclipse on a sample data set.**

Keywords— **Machine Learning, Recommender Systems, Apache Mahout, User-Based Collaborative Filtering, Item-Based Collaborative Filtering**

I. INTRODUCTION

The term machine learning was given by "Arthur Samuel". It is a branch of statistics and artificial intelligence which deals in programming computers to optimize the performance criterion using our previous experience [13].In other words machine learning is a process through which we can train the system and the system becomes capable of taking decisions just like humans .Learning is a process of gaining knowledge either by self study or experience or by being taught. To understand machine learning we need to focus on basic learning techniques which can be broadly categorized into three categories: supervised learning, unsupervised learning and semi-supervised learning.

In supervised learning we learn from the data which is already labelled. System analyses the current data set and whenever next time we feed our algorithm with new dataset, the system will be able to generate the right solution by matching the previously trained dataset. Applications of supervised learning are but not limited to handwriting recognition, speech recognition, spam detection etc. The algorithms associated with this type of learning are decision trees, naïve bayes, logistic regression etc.

The shortest way to describe unsupervised learning is "learning without a teacher". We deal with the unlabelled data over here and the basic aim is to find hidden structure in the data. The raw dataset is simply feed and it is up to the algorithm to decide and come up with different closet points within the dataset. Unsupervised learning is focused on exploratory analysis rather than predictive analysis. An application which is closely related to unsupervised learning is pattern recognition based on the algorithms like hidden markov models, feature extraction methods etc. The reason why supervised learning methods are fast and accurate is based on the simple fact that training data includes the input as well as the desired output.

In semi-supervised learning, learning methodology of both the above learning approaches is combined. At least one or more of these learning techniques form the base of machine learning use cases. One of the common machine learning use cases is focused on recommender system for various e commerce and other category websites. Recommender systems are discussed in brief in the next section.

II. REVIEW OF LITERATURE

In our day to day life we rely on recommendations from our friends, family, media, social networks etc. Recommender systems [2] these days have become the basic component of any e-commerce website. The question which arises over here is why any company has to provide recommendations? The simplest answer to the question is recommender systems introduce the user to a wide collection of items believing that the user will be interested in.

To understand, let us take a simple example of you tube recommendations. The basic aim of providing recommendations is to increase the time the user spends on the site and also to increase the number of videos that user can watch. In order to generate personalized recommendations, recommender system of you tube combines the users' personal behaviour on the site with the related videos association rules [5] . Any video explicitly liked by the user, added to the playlist or was given a rating act as basic seed recommendation factor.

978-1-5386-0514-1/17/$31.00 ©2017 IEEE

Two basic paradigms involved in generating recommendations for any recommender systems are collaborative Filtering and content-based filtering. Collaborative filtering [6] is a technique that uses information of a user like ratings or purchases made by the user to other users of the site who have a similar taste. This can be done either matching a similarity between users or items. The two approaches have been given the names User based collaborative filtering and Item based collaborative filtering. You tube and amazon recommends items based on item to item collaborative filtering. For larger datasets, item based technique provides better results as compared to their counterparts. Because Items usually don't change frequently, so this can be computed in the offline mode also [8].

In order to develop recommender system various approaches can be used like: develop from the scratch, an existing recommender engine can be exploited or select a platform according to the requirement. A combination of these approaches can also be used. One such platforms 'Apache Mahout' forms topic of discussion for our next section

III. MAHOUT: A RECOMMENDER PLATFORM

Apache Mahout is an apache software foundation project to produce open source implementation of machine learning techniques like clustering, classification, collaborative filtering and frequent item set mining. In short mahout can be described as "a scalable machine learning library". Mahout is not the only machine learning framework available but other popular frameworks are also present like Weka and R. The reason for the preference of a platform like Mahout apart from the apache association with mahout is the scalable framework. Because of the availability of peta bytes of data (Big Data) we need a platform like this. Mahout is a very strong competitor when dealing with massive data[11].

A. Origin of Mahout

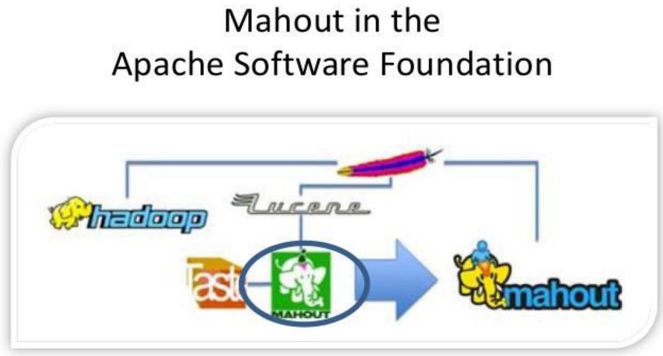

Fig. 1. Original Mahout Project

The name mahout came from a hindi word which means 'guider of elephant'. Since it runs algorithms on top of the hadoop, so it has been given the name mahout. Mahout began life in 2008 as a sub project of apache lucene's project .lucene is an API which provides advanced implementations of techniques like information retrieval & mining [1]. Soon mahout absorbed the 'taste' collaborative filtering open source project. The following figure shows the original Mahout project [12].

Some recent releases of mahout have been incorporated in the table given below.

TABLE 1: IMPORTANT BENCHMARKS IN THE RELEASE DATES OF MAHOUT

Date of release (mm-dd-yyyy)	Version
17-04-2017	0.13.0
11-04-2016	0.12.0
07-08-2015	0.11.0
31-05-2015	0.10.1
11-04-2015	0.10.0
01-02-2014	0.9
25-07-2013	0.8
16-06-2012	0.7
06-02-2012	0.6

B. Mahout Architecture

The general architecture of mahout is based on three-tier architecture including applications, algorithms and shared libraries. Applications are the one's which invoke mahout APIs. Algorithms constitutes of classification, clustering, recommenders etc. Shared libraries consist of Apache Hadoop, collections, math vectors and utilities like lucene vectorizer [10].

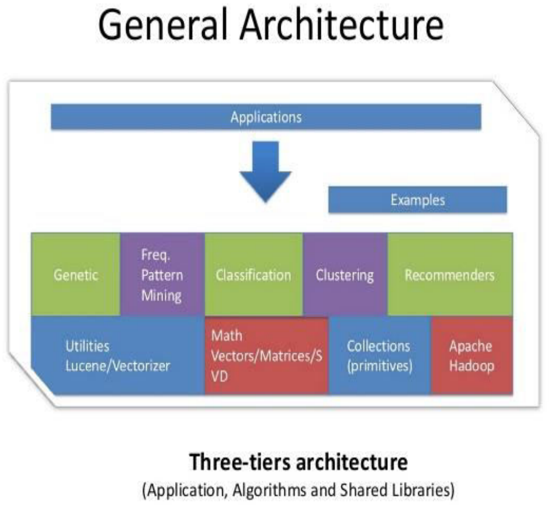

Fig. 2. General architecture

C. Mahout Recommender System Architecture

Recommender system part of mahout is known by the name ‘taste’. The basic idea involved behind recommender architecture[5] is a java based application invokes a mahout recommender which is based on data model set on a number of user preference items built on the ground of a physical storage like database, files etc [8][12].

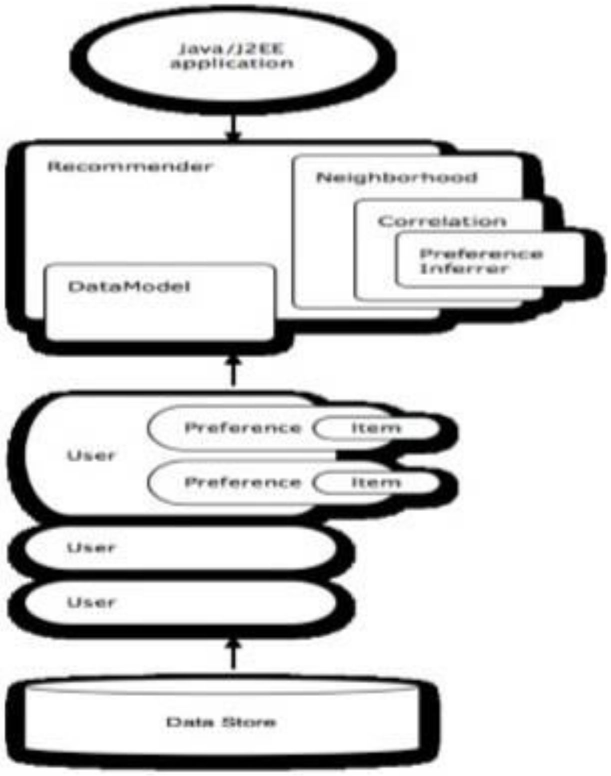

Fig. 3. Recommender System architecture

Top level packages define the mahout interfaces to these key abstractions:

1. Data Model Interface:
   Methods to map raw data in a mahout compliant form

2. User Similarity Interface:
   Methods to figure out similarity between two users

3. Item Similarity Interface:
   Methods to calculate the degree of similarity between two items

4. User Neighborhood Interface:
   Methods defining the neighborhood concept .

5. Recommender Interface:
   Methods which implement the recommendations concept

IV. BUILDING A RECOMEENDATION ENGINE USING MAHOUT VIA ECLIPSE

With the interfaces mentioned above it is possible to personalize your recommendation. The Data Model interface when implemented in Eclipse expects a file where each line contains the following data: a userid, item id, optional preference value, last but not the least optional timestamp. Fields are generally delimited via commas or tabs. For implementation purpose we have take taken a sample dataset with first three fields in the csv format: [user id, item id, ratings] [4].

TABLE 2: SAMPLE DATA SET

User id, Item id, ratings
1, 1010, 5.0
1, 1011, 3.0
1, 1012, 2.5
2, 1011, 1.0
……
5, 1016, 4.0

After importing the required packages in eclipse, the first and foremost step is to load dataset. The dataset having all known interactions is known by the class name DataModel.

DataModel model= new FileDataModel (new File (“sample.csv”));

Similarity is measured either by correlation or by distance. Let’s take a look at some common similarity measures:

Euclidean Distance:

This simplest measure is for calculating the similarity between two items. The Euclidean distance between two items A and B in a dataset is defined by the following equation:

$$\text{Euclidean Distance}(A, B) = \sqrt{\sum_{i=1}^{n} |A_i - B_i|^2}$$

Cosine Similarity:

Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. Cosine similarity between two vectors A and B is given by the equation [7]:

$$\text{Cosine Similarity} = \cos\theta = \frac{A \cdot B}{\|A\|\|B\|}$$

Pearson Correlation:

Pearson's correlation coefficient is a well known correlation coefficient calculated between two variables as the covariance of the two variables divided by the product of their standard deviations [7].

$$\rho(A,B) = \frac{\mathrm{cov}(A,B)}{\sigma_A \cdot \sigma_B}$$

```
UserSimilarity similarity = new PearsonCorrelationSimilarity
(model);
```

K- Nearest neighbour method is the most widely used approach for Collaborative Filtering. Here in this code 2 represents the number of neighbors we are interested in [14].

```
UserNeighborhood                    neighborhood=new
NearestNUserNeighborhood (2, similarity, model);
```

The Recommender Interface makes the recommendations for the user either by comparing items or by users with same taste depending upon the underlying technique i.e. either item-based or user-based. Finally it collects the highest similarity values and offers it as recommendation result.

```
Recommender recommender =new Generic User Based
Recommendation (model, neighborhood, similarity)
List<RecommendedItem> recommendations= recommender.
recommend (1, 1);
for (RecommendedItem recommendation: recommendations)
{
  System.out.println (recommendations);
}
```

Finally after the successful implementation of all the interfaces result was produced in the following format

4	1012	3.0

Mahout offers implementation both the variants item-based as well as user-based approach. Item Based approach is a viable choice when there is lots of data because it doesn't require access to the initial data as item-profile doesn't change with time and even if some changes are made to a product, it is considered as a new item and most likely will be similar to the previous item. A user based approach is considered as a laid back model as it needs to access the initial data. A user profile changes over time. Item based approach works well in terms of efficiency where as user based approach's accuracy is proven to be slightly better than item based approach.

V. EVALUATION OF A RECOMMENDER SYSTEM

The best part about Mahout's recommendation engine is the ease of implementation on an existing cluster but to ensure good performance tuning is required while working with large datasets. As MapReduce need to access data and variables from disk it is not considered a suitable choice for iterative algorithms.

A comparatively new Apache project Spark is getting wide popularity in research and industry. It works on the concept of in-memory computation due to which it has shown significant improvement in terms of speed and resource utilization.

Said etal. [9] focused on the importance of having clear instructions for effective comparison of Recommender Systems for across different platforms. But it is a difficult task as many different designs are available across platforms. But still they can be evaluated [3] on the basis of some common parameters as mentioned in the table.

TABLE 3: EVALUATION PARAMETERS FOR RECOMMENDER SYSTEMS

Evaluation	
Measures Based on IR	Measures Based on Prediction
Precision	Mean-Average Error
Recall	RMSE(root mean square error)

VI. FUTURE SCOPE & CONCLUSION

There is lot of craze for machine learning these days because of Big Data. The problem of scalability was solved by the mahout to a certain extent because of the presence of hadoop framework. But with the increasing size of data sets we need to have a look at other machine learning frameworks as well A large number of ML tool kits are available in the market but one size does not fit all. Researchers reject them sometimes because of the lack of features availability like in case of recommender system we need at least one type of filtering technique as part of the structure. Keeping the future prospects in mind we can compare following Machine Learning frameworks supporting Collaborative Filtering: Distributed-Weka, Flink-Ml and Oryx with Apache mahout in terms of Scalability, Speed and Coverage etc.

Recommender systems are turning out to be a very effective tool embedded in to the websites for increasing the user experience. The successful implementation of mahout or any other recommender architecture can prove out to be a boon for the sites trying to incorporate recommendations as part of their system.

REFERENCES

[1] Carlos E. Seminario, David C. Wilson. “Case study evaluation of mahout as a recommender platform”. ACM 2012, pp 45-50.

[2] Gediminas Adomavicius and Alexander Tuzhilin., ” 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, , 2005, pp. 734–749

[3] J.L.Herlocker, J.A. Konstan, L.G. Terveen and J. Riedl. “ Evaluating Collaborative Filtering Recommender Systems”, ACM Transactions on Information Systems, 2004, pp 5-53

[4] Katrien Verbert, Hendrik Drachsler, Nikos Manouselis, Martin Wolpers, Riina Vuorikari, Erik Duval, “Dataset driven research for Improving Recommender Systems for Learning”, Proceedings of the 1st International Conference on Learning Analytics and Knowledge, 2011, pp 44-53

[5] Luis G. Perez, Francisco Chiclana, Samad Ahmadi, “A Social network representation for Collaborative Filtering Recommender Systems”, International Conference on Intelligent Systems Design and Applications (ISDA), 2011, pp 438-443

[6] Michael D. Ekstrand, John T. Riedl and Joseph A. Konstan, ” Collaborative Filtering Recommender Systems “, Foundations and Trends in Human Computer Interaction, vol. 4, 2011, pp 81-173

[7] Ruchika, A.V. Singh, Dolly Sharma, “Evaluation Criteria for Measuring the Performance of Recommender Systems”, 4th International Conference on Reliability, Infocom Technologies and Optimization 2015, September 2-4, 2015, Noida, pp.462-467

[8] Sachin Walunj, Kishor Sadafale.” An online Recommendation System for E-commerce Based On Apache Mahout Framework”. SIGMIS-CPR. ACM 2013, pp 153-158.

[9] Said A, Bellogín “A. Comparative Recommender System Evaluation: Benchmarking Recommendation Frameworks”, Proceedings of the 8th ACM Conference on Recommender systems (RecSys’14); 2014, pp. 129–136.

[10] Sara Landset, Taghi M. Khoshgoftaar, Aaron N. Richter, Tawfiq Hasanin, “A Survey of Open Source Tools for Machine Learning with Big Data in the Hadoop Ecosystem”, Journal of Big Data, Nov 2015

[11] Sebastian Schelter, Sean Owen “Collaborative Filtering with Apache Mahout”, ACM conference on Recommender systems 2012.

[12] The Apache Mahout Project, “Apache Mahout”, available: https: //mahout.apache.org

[13] Tyson Condie, Paul Mineiro, Neoklis Polyzotis, Markus weimer “Machine Learning for Big Data”, International Conference on Management of Data SIGMOD 2013, pp939-942

[14] Zhi-Dan Zhao, Ming-Sheng Shang, “ User-based Collaborative-Filtering Recommendation Algorithms on Hadoop”, 3rd International Conference on Knowledge Discovery and Data Mining, 2010, pp 478-481.

[15] Som S., Banerjee M., (2013) “Cryptographic Technique by Square Matrix and Single Point Crossover on Binary Field”, 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA'13), IEEE Explorer, Print ISBN: 978-1-4673-2820-3, February 12 – 14, 2013, Sharjah, UAE.

[16] Shobha Tyagi, Subhranil Som, Qamar Parvez Rana (2017) “Trust based Dynamic Multicast Group Routing ensuring Reliability for Ubiquitous Environment in MANETs”, International Journal of Ambient Computing and Intelligence (IJACI), Volume 8, Issue 1, Scopus Indexed, ISSN: 1941-6237, DOI: 10.4018/IJACI, Pages 70 – 97, January – March 2017. (http://www.igi-global.com/journals/abstract-announcement/158348)