Machine Learning Algorithms for Recommender System - a comparative analysis

Satya Prakash Sahu
University of Hyderabad
Hyderabad, India
spsahu@uohyd.ac.in

Anand Nautiyal
University of Hyderabad
Hyderabad, India
anandnautiyal@uohyd.ac.in

Mahendra Prasad
University of Hyderabad
Hyderabad, India
je.mahendra@uohyd.ac.in

Abstract: Recommendation system is one of the most popular applications of Artificial Intelligence which attracts many researchers all over the globe. The advent of the Internet era has brought wide implementation of recommendation system in our everyday lives. There are many machine learning techniques which can be used to realize the recommendation system. Among all these techniques we are dealing with Content Based Filtering, Collaborative Based Filtering, Hybrid Content-Collaborative Based Filtering, *k*-mean clustering and Naive Bayes classifier. We have exploited these algorithms to their extreme in order to achieve the best possible precision and have presented a comprehensive comparative analysis. The strength of all these algorithms can be clearly realized by the significant enhancement in the accuracy, depicted by the experimental analysis taking cold start problem into consideration.

Keywords: Recommender System, Classifier, Content Based, Collaborative Based, Cluster, Correlation.

1. INTRODUCTION

Recommendation system[10] is an application which is used for prediction in various domains throughout the internet. A large amount of data flows through the internet and it gives away a lot of information regarding the user searching activity. The information extracted from the pattern of previously searched data can be molded into the prediction of relevant data for the user[1]. The implementation of the system can be performed by various techniques. In this paper, we have discussed Content Based Filtering, Collaborative Filtering[10], Hybrid Content-Collaborative Based Filtering, *k*-mean clustering Based and Naive-Bayes Classifier based techniques.

The Content Based Filtering approach takes into account a user’s profile which is constructed based on his previous ratings[2]. His ratings determine his inclination and interests, forming the basis for recommending a new item. A higher rating denotes a higher likelihood of the user to visit similar items. So, a new item is recommended according to the maximum number of ratings given by the user in a genre[3].

In the Collaborative Based Filtering, recommendation for a user is governed by other users’ profiles. An item is recommended based on the ratings of other users who have similar interests as the user under consideration[2][4]. In another approach, the content and collaborative based filtering are combined to form the Hybrid Content-Collaborative Based Filtering. It includes the advantages of both the methods and outperforms both of them.

In the *k*-mean clustering, the similarity between the objects is calculated by the means of various distance measures such as Euclidean distance[5], Pearson Correlation, etc. The value of *k* determines the number of clusters to be formed[6]. The nearest *k* objects are the most similar to one another. These clusters of similar objects drive the recommendation of new arriving objects. Naive Bayes is another popular and efficient classifier based on Bayes theorem. It is a conditional probability based classifier. The prior knowledge of the classifier assists learning. The naive assumption is that the features are conditionally independent[7].

In this paper, we have used the MovieLens dataset[8]. All the above algorithms deal with this dataset in order to recommend the movies and calculate the precision along with tackling the cold-start problem[3]. Cold-start problem is one of the most commonly encountered challenges of the recommendation system. It is also known as the new user problem as it creates problem of generating recommendations for the new user. We have divided this analysis into various sections. Section II describes the different state-of-the-art techniques for the recommendation system. Section III gives the experimental results for all these techniques. Section IV concludes the study. Section V describes the future work that we propose.

2. ALGORITHMS

2.1 Content Based Filtering

The Content Based Filtering considers the items rated by a user to formulate the future recommendations while exploring the internet services. A user tends to rate an item which he likes or dislikes. His ratings reflect his response towards that item. If he likes an item, he rates it higher and lesser ratings denote that he is not much interested. These rated items serve as the 'content' in the Content Based Filtering[2][3]. Based on this content, the user is recommended future items which he might approve of. Here, the user is recommended movies which fall in a particular genre of his liking.

Algorithm 1. Content Based Filtering

Input: users X, movies m, rating r, movie genre m_g , Number of movies to be recommended(μ).

Output: Recommended movies R

```
1. for all users do
2. Select seen movies $s$, unseen movies $s'$, association of unseen movies $as_i'$ w.r.t $X$, association of each genre $ag_j$ w.r.t $s'$ , where $i$ is 1 to $n$ and $j$ is 1 to $m$.
3. Calculate $score_j$ .
4. Select highest three $score_j$
5. Select $m' \subset s'$ according to highest three $score_j$
```

```
6.   Calculate score $m_e'$ where $e \in m'$
7.   Return top $\mu$ score recommendations.
8.   end for
```

In this algorithm, the notations used have the following meaning : association of each movie as_i represents total number of users who rated movie i Є *s'*, association of each genre ag_j represents total number of movie belonging to genre *j*.

$score_j = ag_j / m$

$score(m_e') = am_e / \text{total count of } m'$

2.2 Collaborative Filtering

There can be many users who must be having the same pattern of rating an item as the user intended. This similar pattern of their ratings with the user guides the Collaborative Filtering[2][3][10]. The notion behind the Collaborative Filtering is the recommendation of an item based on the preferences of like-minded users.

Algorithm 2. Collaborative Filtering

Input: users *X*, movies *m*, rating *r*, Number of movies to be recommended*(μ)*.

Output: Recommended movies *R*.

```
1.   for all users do
2.   Select seen movies $s$, unseen movies $s'$
3.   Find similarity ($sim_i$) w.r.t $s$, where $i$ = 1 to $n$.
4.   Select highest $sim_i$ user
5.   Select $m'$ Є $s$ of user obtained in step 4 and $s'$ of $i^{th}$ user.
6.   Calculate weight $W(m_e')$ where $e$ Є $m'$
7.   Return top $\mu$ weight recommendations.
8.   end for
```

In this algorithm, the notations used have the following meaning : sim_i represents common movies between user *i* and other users.

$weight(m_e') = \text{rating of particular movie}_e / \text{max rating}.$

2.3 Hybrid Filtering

To cater better precision, a hybrid filtering method is used which can provide the advantages of both the content and the collaborative approaches[4] and can overcome their shortcomings. Suppose, the user appreciates mostly movies in $g \subset G$ genres, and the collaborating users also give high ratings to the $g \subset G$ genres, then *g* will be taken as the metric to recommend movies to the user.

Algorithm 3 .Hybrid Filtering

Input: users *X*, movies *m*, rating *r*, movie genre m_g , Number of movies to be recommended(*μ*).

Output: Recommended movies *R*.

```
1.   for all users do
2.   Select seen movies $s$, unseen movies $s'$, association of each genre $ag_j$ w.r.t $s'$, where $i$ is 1 to n and $j$ is 1 to $m$.
3.   Calculate $score_j$ .
4.   Select highest three $score_j$
5.   Select $m''$ Є $s$ of the $i^{th}$ user according to highest three $score_j$
6.   Find similarity ($sim_j$) w.r.t m''
7.   Select highest $sim_j$ user.
8.   Select $m'$ according to its highest three $score_j$ Є $s$ of user obtained in step 7 and $s'$ of the $i^{th}$ user under consideration.
9.   Calculate weight $W(m_e')$ where $e$ Є $m'$
10.  Return top $\mu$ weight recommendations.
11.  end for
```

2.4 K-Mean Clustering

The *k*-mean is a non parametric classification technique. It distributes the items into *k* clusters according to their proximity to one another. In this paper, this proximity is being measured by using the Euclidean distance[11]. For calculating the Euclidean distance we have taken rated and unrated movies as binary. Each cluster possesses a centroid which is the mean of all the items in the cluster. All the objects in a cluster move towards the centroid and the centroid is updated in each iteration. The iteration continues until a saturation point arrives, when the centroid stops altering. By following this approach we are decreasing the search space which results in reduced computational complexity[6]. These computations are performed off-line which helps the classification to be efficient in terms of time complexity.

Algorithm 4. *k*-mean clustering

Input: users *X*, movies *m*, rating *r*, Number of movies to be recommended *μ*, value of *k*.

Output: Recommended movie *R*.

```
1.   begin
2.   Randomly select $k$ centroids.
3.   Calculate euclidean distance ($eucd$) for $X$ from $k$ centroids.
4.   Allocate $X$ to $k^{th}$ cluster according to $eucd$.
5.   Update centroid for each cluster with (summation($k_i$ ) from 1 to $p$)/$p$, where $p$ is the number of members in $k_i$ cluster
6.   Repeat step 3 to step 5 until centroid(t) $\neq$ centroid (t+1).
7.   for all users do
8.   Select seen movies $s$, unseen movies $s'$.
```

```
Find similarity ($sim_i$) w.r.t $s$, where $i$ = 1 to $p$.
Select highest $sim_i$ user.
select $m' \subset s$ of highest $sim_i$ and $s'$ of $i^{th}$ user.
Calculate weight $W(m_e')$ where $e \in$ m'
Return top $\mu$ weight recommendations.
end for
end
```

2.5 Naive Bayes

The Naive Bayes is based on the Bayes theorem. The probabilistic approach followed by Naive Bayes Classifier determines the probability of the classification and helps in finding the uncertainty about the model[9]. It is an efficient learning algorithm which uses the prior knowledge of the observed data. The Naive assumption is that the features are conditionally independent[1].

Algorithm 5. Naive Bayes

Input: users X, movies m, rating r, number of movies to be recommended(μ)

Output: Recommended movies R.

```
for all users do
Select seen movies $s$, unseen movies $s'$ .
Find similarity ($sim_i$) w.r.t s, where i = 1 to n.
Select $x' \subset X$ where $sim_i > 10$.
Calculate association of unseen movies $as_i$ ' w.r.t to $x'$
Calculate score ($s_e'$) where $e \in$ s'.
Return top $\mu$ score recommendations.
end for
```

3 EXPERIMENTAL RESULT

We now illustrate the analysis of the experiments performed and provide a comparison of all the state-of-the-art methods described above. To compare their accuracy we have used the MovieLens dataset of 10K, 50K and 100K. The dataset varies in sparsity. For example, the 100K MovieLens dataset has 100K ratings, 943 users and 1682 movies of 19 different genres. The analysis of these algorithms is demonstrated based on precision measure. For each test user, we convert 30% of the user's seen movies into unseen movies and apply the algorithms described above. Out of the total number of recommendations (T), the ones which are also present in the converted movies are the correct recommendations(tc).

$$\text{Precision} = (\Sigma tc / \Sigma T) * 100$$

For all the experiments, we are taking value of μ = 5 and value of k = 10.

Algorithm/Size	10K	50K	100K
Content Based	18.45	18.66	19.10
Collaborative	17.97	18.69	19.95
Hybrid	20.31	21.03	22.20
K-Mean	21.05	21.93	22.67
Naive Bayes	24.62	25.19	25.73

Table 1. Precision of Different Algorithms.

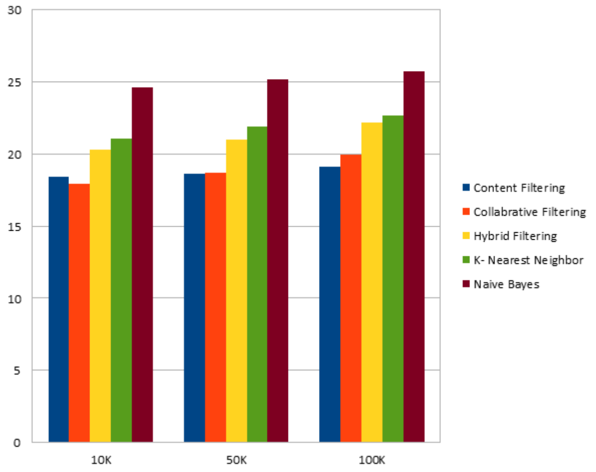

Fig. 1. Precision Comparison

4 CONCLUSION

All the algorithms described in this paper are compared with respect to their precision rates. This comprehensive analysis depicts the strength and the weakness of each one of them in different versions of the MovieLens dataset. The experiments performed are the witness of the sparsity handling by these algorithms. Our experiments have shown promising results and this paper conforms that out of all these approaches Naive Bayes gives the best precision.

5 FUTURE WORK

With this paper, we have achieved encouraging results from all these algorithms. In the real time sophisticated recommendation systems there is a need of high accuracy. Such systems still have space for improvement. There are several machine learning algorithms which can be applied to these real time systems. It is worthwhile to examine those other algorithms to improve the precision further.

6. REFERENCES

[1] Katore L.S., and Umale J.S. Comparative Study of Recommendation Algorithms and Systems using WEKA, International Journal of Computer Applications, Volume 110 – No. 3, pp14-17, 2015.

[2] Tewari A.S., Kumar A., and Barman A.G.,. Book Recommendation System Based on Combine Features of Content Based Filtering, Collaborative Filtering and Association Rule Mining, IEEE, 978-1-4799-2572-8, pp 500-503, 2014.

[3] Wanaskar U.H., Vij S.R., and Mukhopadhyay D. A Hybrid Web Recommendation System Based on the Improved Association Rule Mining Algorithm, Journal of Software Engineering and Applications, 6, pp 396-404 2013.

[4] Shinde U., and Shedge R. Comparative Analysis of Collaborative Filtering Technique, IOSR Journal of Computer Engineering, Volume 10, pp 77-82 2013.

[5] CHENG X., WANG J., Danqian LU. Research of Question Analysis Based on HNC and K Nearest Neighbor, Journal of Computational Information Systems, 6:10, pp 3449-3455, 2010.

[6] Campos P.G., Bellogín A., Díez F., and Chavarriaga J.E. Simple Time-Biased KNN-based recommendations, ACM, 978-1-4503-0258-6, 2010.

[7] Puntheeranurak S., and Pitakpaisarnsin P. Time-aware Recommender System Using Naïve Bayes Classifier Weighting Technique, International Symposium on Computer, Communication, Control and Automation, 3CA, pp 266-269 ,2013.

[8] Bellogín A. Castells P. and Cantador I. Precision-Oriented Evaluation of Recommender Systems: An Algorithmic Comparison, ACM, 978-1-4503-0683-6, 2011.

[9] Puntheeranurak S. and Sanprasert S. Hybrid Naive Bayes Classifier Weighting and Singular Value Decomposition Technique for Recommender System, IEEE, 978-1-4244-9698-3, pp 473-476, 2011.

[10] Laishram A., Sahu S.P., Padmanabhan V., and Udgata S. K. Collaborative Filtering, Matrix Factorization and Population Based Search: The Nexus Unveiled. ICONIP , Part III, LNCS 9949, pp. 352–361, 2016.

[11] Prasad M. and Singh A. A Novel Hybrid Ant Colony Optimization Approach to Terminal Assignment Problem. ACM, AICTC ’16, Article 29, August 12-13, 2016.