

A Simple Yet Effective and Efficient Collaborative Filtering Based Recommendation System

MASUM BILLAL*

In order to improve both prediction accuracy and the time required to recommend items to a user based on collaborative filtering, a simple algorithm is proposed. The algorithm is different than traditional algorithms only in some simple steps.

Additional Key Words and Phrases: Recommender system, collaborative filtering, similarity measure

ACM Reference Format:

Masum Billal. 2021. A Simple Yet Effective and Efficient Collaborative Filtering Based Recommendation System. 1, 1 (April 2021), 11 pages. <https://doi.org/10.1145/1122445.1122456>

1 INTRODUCTION

A common problem with typical filtering based recommendation systems they usually require a lot of users similar to a user for prediction (such users are called **neighbors**). This heavily affects performance since the complexity gets incredibly high as well. [?] uses a memory based collaborative filtering technique that uses similarity between users to predict recommendations using KNN algorithm. This algorithm suffers from some problems such as when too few items are common between two users, predicting accurately becomes harder. [?] uses a model based learning which is inherently very expensive. [?] introduces a way to improve calculating similarity more accurately but they do not offer all that much improvement for the cost of learning similarity, so in this paper such improvement has been ignored and more traditional approaches have been used. [?]. [?] uses hybrid collaborative and content filtering or memory and model based algorithm for recommendation. While these works have their merits, most of them are complicated and offer little value to be used in practice. In this paper, we have showed that it may be possible to gain significantly better results by simply modifying traditional approaches a little bit using some principles which are mathematically sound. We will show that using our proposed modifications, we get a flat line in accuracy metric graphs much earlier.

2 ALGORITHM

Let us first describe the assumption problem that we want to solve. Let \mathcal{M} be a set of movies and \mathcal{U} be a set of users. We are given a set of triplets $(u, m, r) \in \mathcal{S}$ such that $u \in \mathcal{U}, m \in \mathcal{M}$ and $1 \leq r \leq 5$ which denotes the rating r of movie m given by user u . We are also given a set of pairs $(u, m) \in \mathcal{T}$ such that $u \in \mathcal{U}, m \in \mathcal{M}$ and we want to predict the rating r given by user u to the movie m . Let r_{um} be the rating of movie m given by user u and C_{uv} be the set of items

*This research is done independently by the author, no grant or fund was accepted by the author for this work.

Author's address: Masum Billal, billalmasum93@gmail.com.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Association for Computing Machinery.

Manuscript submitted to ACM

that both u and v rated. We use the *Pearson Correlation Coefficient* (see [?]) as the measure of correlation between user u and v . However, we will use a normalized version of it for practical consideration as follows:

$$S_{uv} = \frac{\sum_{m \in C_{uv}} (r_{um} - \bar{r}_u)(r_{vm} - \bar{r}_v)}{\sqrt{\sum_{m \in C_{uv}} (r_{um} - \bar{r}_u)^2} \sqrt{\sum_{m \in C_{uv}} (r_{vm} - \bar{r}_v)^2}}$$

$$S_{uv} \leftarrow \frac{S_{uv} + 1}{2}$$

This normalization is done using the fact that Pearson correlation coefficient p_{uv} is always in the range $[-1, 1]$. We call two users u and v similar if $S_{uv} \geq s$ for some positive real number s such that $0 \leq s \leq 1$. Typically, we want s in the range $[.5, 1]$. For this paper, we will consider $s \in \{.7, .8, .9\}$. For a movie m , let \mathcal{U}_m be the set of users who rated m and M_u be the set of movies rated by user u . For a set or tuple A , let \bar{A} denote the average of the numbers in A . For a tuple of weights $w = (w_1, \dots, w_n)$ such that $0 \leq w_i \leq 1$ and $\sum_{i=1}^n w_i = 1$ and a tuple of positive real numbers $a = (a_1, \dots, a_n)$, the *weighted harmonic mean* of a is defined as

$$\mathfrak{H}(a, w) = \frac{\sum_{i=1}^n w_i}{\sum_{i=1}^n \frac{w_i}{a_i}}$$

Usually, *weighted arithmetic mean* is used to predict the ratings in a recommendation system. But in this paper, we have investigated the results using harmonic mean. Next, we describe the rating prediction algorithm for a pair (u, m) .

Algorithm 1: Algorithm to predict rating

Input: Test data in the format (u, m) , Threshold t , similarity s , T to take first T neighbors for a user u

Output: A single integer in the range $[1, 5]$ denoting the predicted rating

Data: Train data in the format (u, m, r)

```

1  $W \leftarrow []$ 
2  $X \leftarrow []$ 
3  $tot \leftarrow 0$ 
4  $R \leftarrow U_m$ 
5  $res = 0$ 
6 for  $v \in R$  do
7   if  $|C_{uv}| < t$  then
8     continue
9   end
10  if  $S_{uv} < s$  then
11    continue
12  end
13  if  $S_{uv} \geq s$  then
14     $W \leftarrow [W, s]$ 
15     $X \leftarrow [X, r_{vm}]$ 
16     $tot \leftarrow tot + 1$ 
17    if  $tot > T$  then
18      break
19    end
20  end
21 end
22 if  $tot > 0$  then
23    $res = \mathfrak{H}(X, W)$ 
24 end
25 else
26   if  $|M_u| > 0$  then
27      $res = \bar{M}_u$ 
28   end
29   else
30      $res = \bar{R}_m$ 
31   end
32 end
33  $res = res + .5$ 
34  $res = \text{floor}(res)$ 
35 return  $res$ 

```

2.1 Algorithm Principles

Our algorithm is based on the following principles.

First principle Two users u and v are more relevant for each other if it is ensured that $|C_{uv}| \geq t$ for some large enough positive integer t . It is obvious how this principle helps us establish better correlation between users since two users u and v can have very high correlation with very low number of common items between them.

Second principle If first principle is established, then it is possible to get an accurate estimation of predicted rating with a lower number of neighbors instead of using a very large number of neighbors. This helps us predict a rating a lot more efficiently with better accuracy. It is also possible to get a mathematical sense of why this works in practice. Let m and n be positive integers such that $m > n$. Let u be a user and $\mathcal{N}_m = \{v : |C_{uv}| \geq t\}$ and $\mathcal{N}_n = \{u : |C_{uv}| \geq t\}$ be two sets of neighbors of a user u such that $|C_m| = m$ and $|\mathcal{N}_n| = n$. Using the condition $|C_{uv}| \geq t$, it can be assumed that if $s < t$ for a positive integer s , then

$$P(\sigma^2(A) \leq \sigma^2(B)) \quad (1)$$

should be very high where $A = \{r_{vm} : |C_{uv}| \geq t\}$, $B = \{r_{vn} : |C_{uv}| \geq s\}$ and $P(x)$ denotes the probability of the random variable x . Since higher value of the threshold t ensures better similarity between two users, we can say in a non-rigorous way that the *Pigeonhole principle* ensures (1) is high enough in practice more often than not.

Third principle If two sets of ratings A and B have similar variance and consist ratings given by similar users of u only, then $P(|\bar{A} - \bar{B}| < \epsilon)$ is very high for some positive real number ϵ which is considerably smaller than 1. We can show that a special case holds very often in practice. Since A and B has ratings from similar users only, assume that both A and B consist of 4 and 5 only (our data set rating range is $[1, 5]$). If $|A| = m$ and $|B| = n$ and a is the number of 4 in A whereas b is the number of 4 in B , then

$$\begin{aligned} \bar{A} &= \frac{4a + 5(m - a)}{m} \\ &= \frac{5m - a}{m} \\ &= 5 - \frac{a}{m} \\ \bar{B} &= \frac{4b + 5(n - b)}{n} \\ &= \frac{5n - b}{n} \\ &= 5 - \frac{b}{n} \end{aligned}$$

As we can see, it does not matter if $m \gg n$ or $n \gg m$ since the difference $|\bar{A} - \bar{B}|$ or the ratio $\frac{\bar{A}}{\bar{B}}$ only depends on the ratio $\frac{a}{m}$ and $\frac{b}{n}$. So as long as these ratios are similar, \bar{A} and \bar{B} are similar as well.

3 RESULTS

We show the results of our algorithm on the Movielens 1 million data set and backup our claims with empirical evidence below. The quality of performance is measured using the $F1$ score. The $F1$ score is the harmonic mean of precision and recall.

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

$$F1 = \frac{2PR}{R + P}$$

We have chosen $F1$ as an accuracy metric for couple of reasons.

- (1) It is well known that the harmonic mean of n positive real numbers a_1, \dots, a_n is less than $\min\{a_1, \dots, a_n\}$ for $n > 1$. So an $F1$ value of x indicates that the model has at least x precision and x recall.
- (2) $F1$ score can penalize bad precision or bad recall scenarios properly where arithmetic or geometric means may fail. For example, if we consider a recommendation where we do not predict anything, technically, we do not have any errors. So the precision would be 100 percent whereas recall would be 0 percent. An arithmetic average would give us an accuracy of 50 percent which is clearly wrong. $F1$ score penalizes all such scenarios accordingly. For this particular example, we would still have an $F1$ score of 0 percent.

For this paper, we call a test data point *true positive* if a certain movie m is to be recommended to user u based on the rating. If the rating is over k for some fixed k , then the movie is to be recommended, otherwise it is to be discarded. We have showed results for both $k \geq 3$ and $k \geq 4$ in this paper.

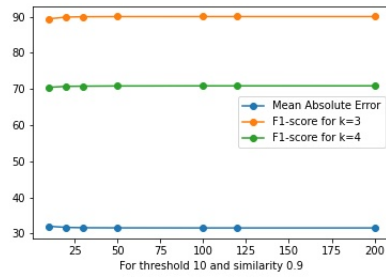
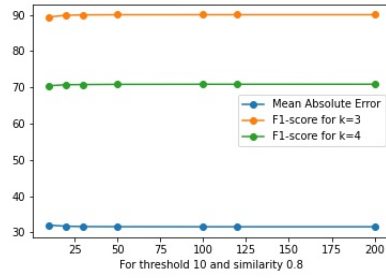
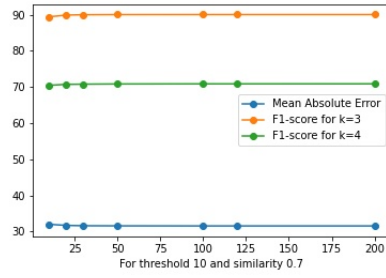
We have also used *mean absolute percentage error* as a metric in place of the more traditional *mean absolute error*. Mean absolute error is calculated using the formula:

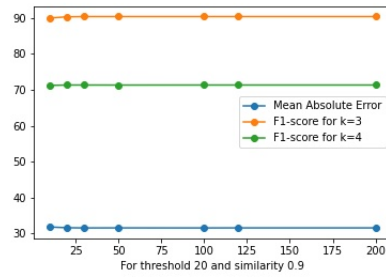
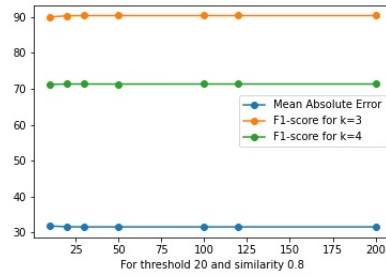
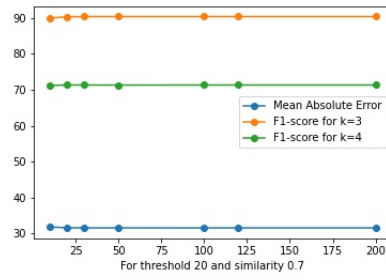
$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - y'_i|$$

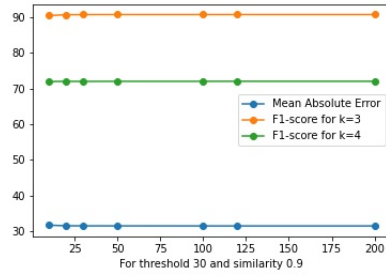
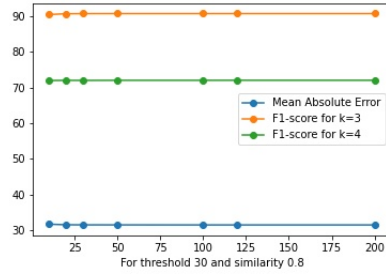
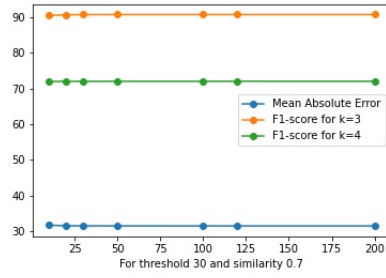
whereas mean absolute percentage error is calculated using the formula

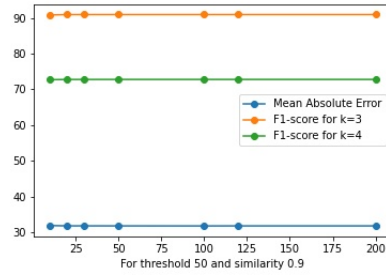
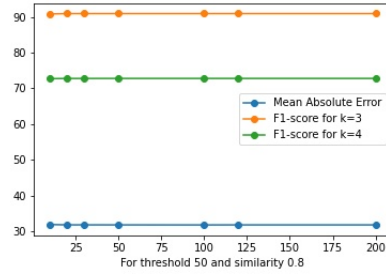
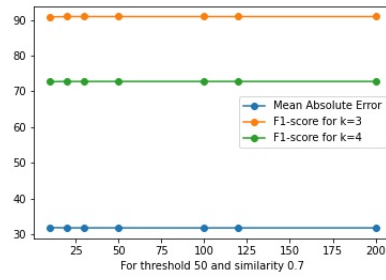
$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - y'_i|}{y_i}$$

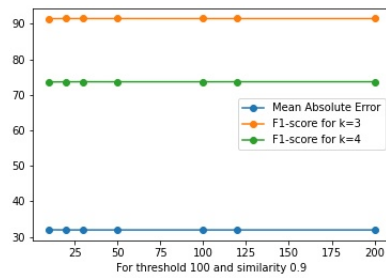
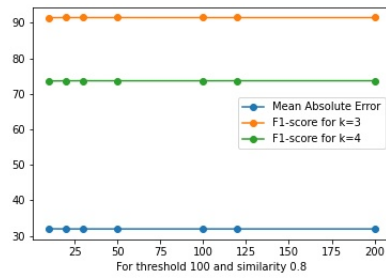
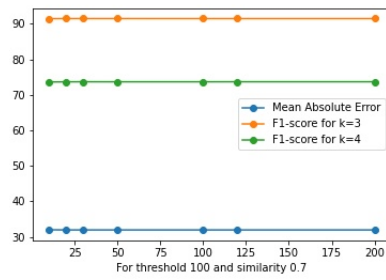
where y_i is the actual value for the data point at i and y'_i is the predicted value of this data point. It is easy to see that mean absolute error puts equal weights on all ratings whereas mean absolute percentage errors mitigates that problem to some extent. So we felt it important to also show the results in terms of mean absolute percentage errors.











4 CONCLUSION

As we can see in the result section, all the graphs tend to give us a flat line very early on. usually, we do not get much improvement in accuracy after 30 neighbors and we do not have to go beyond .7 similarity threshold for considering neighbors.