A Simple Yet Effective and Efficient Collaborative Filtering Based Recommendation System

Masum Billal Mahmudul Hasan Munna

April 15, 2021

Abstract

In this paper, we describe a simple yet effective way to recommend items to a user and predict the rating of an item given by a user with high accuracy.

1 Introduction

A common problem (insert relevant papers) with typical filtering based recommendation systems they usually consider all users similar to a user. To mitigate performance problems, (insert relevant papers) show that we can simply consider only a handful of neighbors for a particular user.

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 $u \in \mathcal{U}, m \in \mathcal{U}$ 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 m 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 that both u and v rated. We use the *Pearson Correlation Coefficient* (see Freedman, Pisani, and Purves [1]) 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 $\mathbf{w} = (w_1, \dots, w_n)$ such that $0 \leq w_i \leq 1$ and $\sum_{i=1}^{n} w_i = 1$ and a tuple of positive real numbers $\mathbf{a} = (a_1, \dots, a_n)$, the *weighted harmonic mean* of $\mathbf{a}$ is defined as

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

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).

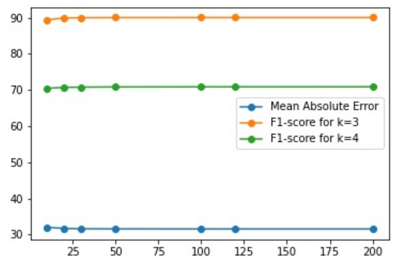
Algorithm 1: Algorithm to predict rating

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

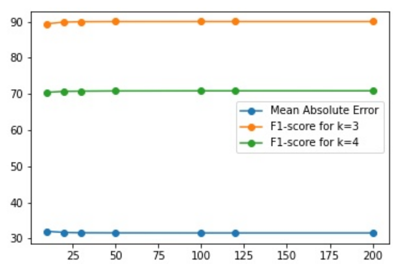
Output: A single integer in the range $[1, 5]$ denoting the predicted rating

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

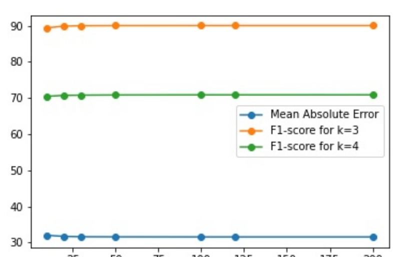
$W = []$
$X = []$
$tot \leftarrow 0$
$R \leftarrow U_m$
res $= 0$
for $v \in R$ **do**
    if $|C_{uv}| < t$ **then**
        continue
    end
    if $S_{uv} < s$ **then**
        continue
    end
    if $S_{uv} \geq s$ **then**
        $W \leftarrow [W, s]$
        $X \leftarrow [X, r_{vm}]$
        $tot \leftarrow tot + 1$
        if $tot > T$ **then**
            break
        end
    end
end
if $tot > 0$ **then**
    res $= \mathfrak{H}(X, W)$
end
else
    if $|M_u| > 0$ **then**
        res $= \bar{M_u}$
    end
    else
        res $= \bar{R_m}$
    end
end
res = res+.5
res = floor(res)
return res



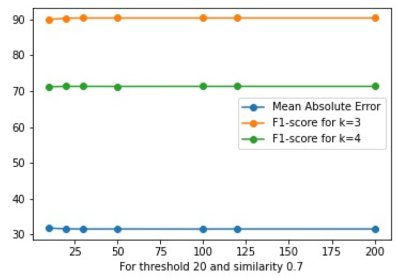
For threshold 10 and similarity 0.7

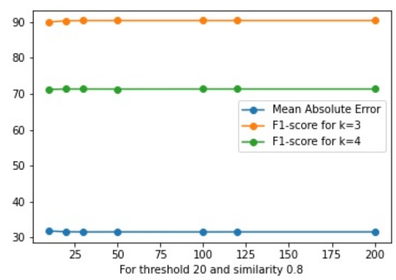


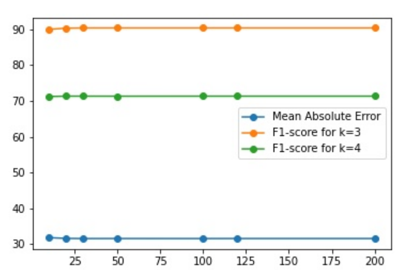
For threshold 10 and similarity 0.8

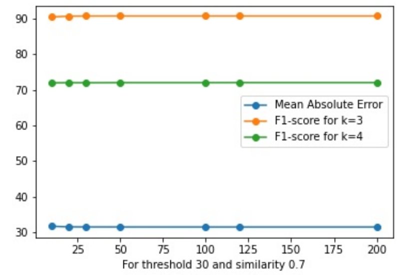


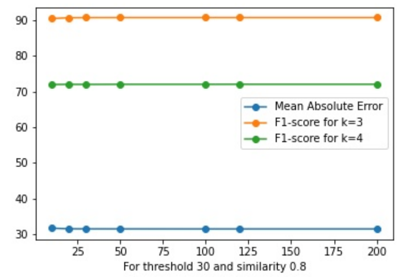
For threshold 10 and similarity 0.9

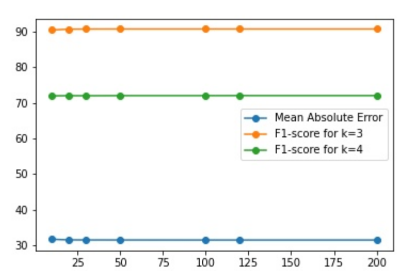

Mean Absolute Error
F1-score for k=3
F1-score for k=4
For threshold 20 and similarity 0.7

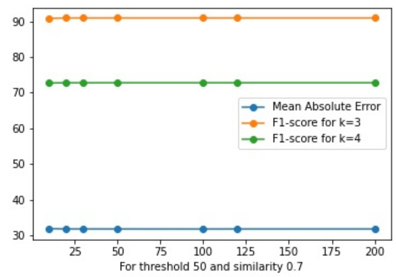

Mean Absolute Error
F1-score for k=3
F1-score for k=4
For threshold 20 and similarity 0.8

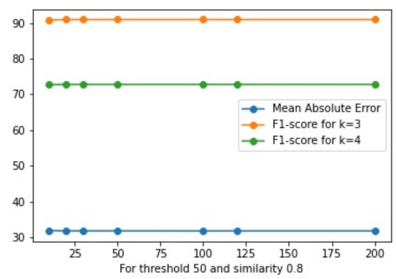

Mean Absolute Error
F1-score for k=3
F1-score for k=4
For threshold 20 and similarity 0.9

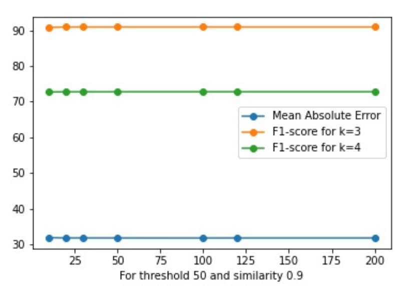

90
80
70
60
50
40
30
25 50 75 100 125 150 175 200
Mean Absolute Error
F1-score for k=3
F1-score for k=4
For threshold 30 and similarity 0.7

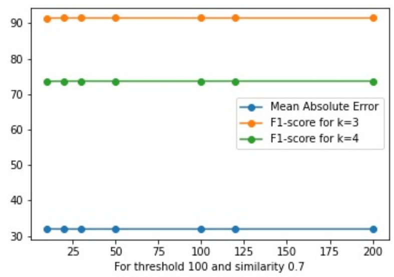

90
80
70
60
50
40
30
25 50 75 100 125 150 175 200
Mean Absolute Error
F1-score for k=3
F1-score for k=4
For threshold 30 and similarity 0.8

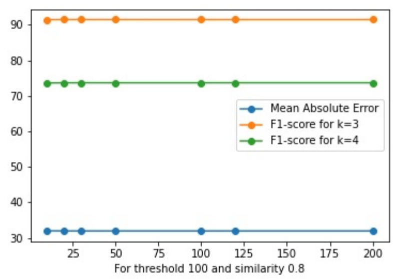

90
80
70
60
50
40
30
25 50 75 100 125 150 175 200
Mean Absolute Error
F1-score for k=3
F1-score for k=4
For threshold 30 and similarity 0.9

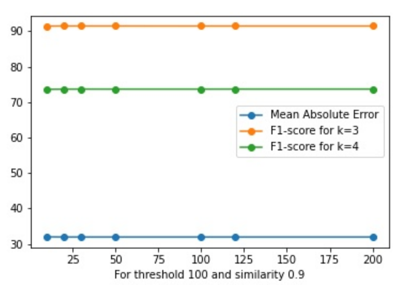

Mean Absolute Error
F1-score for k=3
F1-score for k=4
For threshold 50 and similarity 0.7


Mean Absolute Error
F1-score for k=3
F1-score for k=4
For threshold 50 and similarity 0.8


Mean Absolute Error
F1-score for k=3
F1-score for k=4
For threshold 50 and similarity 0.9


Mean Absolute Error
F1-score for k=3
F1-score for k=4
For threshold 100 and similarity 0.7


Mean Absolute Error
F1-score for k=3
F1-score for k=4
For threshold 100 and similarity 0.8


Mean Absolute Error
F1-score for k=3
F1-score for k=4
For threshold 100 and similarity 0.9

References

[1] David Freedman, Robert Pisani, and Roger Purves. "Statistics (international student edition)". In: *Pisani, R. Purves, 4th edn. WW Norton & Company, New York* (2007).