Movie Recommender System

Submitted By

Krunal M. Varma

14MCECS2

DEPARTMENT OF COMPUTER ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2017

Movie Recommender System

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering (Split)

Submitted By

Krunal M. Varma

(14MCECS2)

Guided By

Dr. Priyank Thakkar

DEPARTMENT OF COMPUTER ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2017

Certificate

This is to certify that the major project entitled **"Movie Recommender System"** submitted by **Krunal M. Varma (Roll No: 14MCECS2)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering (Split) of Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project part-II, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. Priyank Thakkar
Guide & Associate Professor,
CE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Priyanka Sharma
Professor,
Coordinator M.Tech - CSE
Institute of Technology,
Nirma University, Ahmedabad

Dr. Sanjay Garg
Professor and Head,
CE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr Alka Mahajan
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Krunal M. Varma**, Roll. No. **14MCECS2**, give undertaking that the Major Project entitled **"Movie Recommender System"** submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering (Split)** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made.It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Signature of Student

Date:

Place:

Endorsed by

Dr. Priyank Thakkar

(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Dr. Priyank Thakkar**, Associate Professor, Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Priyanka Sharma**, Coordinator M.Tech. CSE, Institute of Technology, Nirma University, Ahmedabad for her kind support and always motivating for good project work.

It gives me an immense pleasure to thank **Dr. Sanjay Garg**, Hon'ble Head of Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr. Alka Mahajan**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation she has extended throughout course of this work.

I would also thank the Institution, all faculty members and staff members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- Krunal M. Varma

14MCECS2

Publications Related to Thesis

International Conference Publications

- Priyank Thakkar, Krunal Varma, Vijay Ukani, *Outcome fusion-based Approaches for User-based and Item-based Collaborative Filtering*, 2nd International Conference on Information and Communication Technology for Intelligent Systems (ICTIS 2017), Springer SIST SERIES, forthcoming. http://www.springer.com/series/8767

Abstract

Online shopping is a trend and a way to go these days for buying many different kinds of products. Typically, before buying any product, customer sees historical ratings received by the product and then makes a conscious decision. This scenario is also applicable to a movie. User relies on various ratings and reviews given by other users before deciding about watching a movie. This form of decision making is useful but relies on general senses of mass. It does not consider individual's taste and preferences. This dissertation aims to fill in this gap.

User-based and Item-based collaborative filtering is exercised in this dissertation for personalized movie recommendation. Similarity between users and items is computed through different possible combinations and their impacts on prediction error is studied. A novel contribution of this dissertation is the fusion of user-based and item-based collaborative filtering to predict the rating. Precisely, approaches based on Genetic Algorithm (GA), Classification and Regression Tree (CART), Random Forest (RF), Linear Regression (LR) and Support Vector Regression (SVR) have been employed to address the fusion challenge. Results are encouraging and demonstrates the usefulness and superiority of fusion approaches.

Abbreviations

CART	Classification And Regression Tree.
GA	Genetic Algorithm.
IBCF	Item Based Collaborative Filtering.
IB	Item Based.
MAPE	Mean Absolute Percentage Error.
MSE	Mean Squared Error.
MAE	Mean Absolute Error.
MB	Memory Based.
MOD	Model Based.
NRMSE	Normalized Root Mean Squared Error.
NN	Nearest Neighbour.
RMSE	Root Mean Squared Error.
SIM	Similarity.
SVM	Support Vector Machine.
SVR	Support Vector Regression.
UBCF	User Based Collaborative Filtering.
UB	User Based.

Contents

List of Figures

Chapter 1

Introduction

In this chapter, first brief introduction to recommender system and what is movie recommender system is given. After that, description regarding types of movie recommender system that are available is briefed. Finally, the problem statement and research gap that is identified in this area of movie recommender system is explained.

1.1 Recommender System

In today's world of technology, everyone rely on online shops which offer good products at very good discounts. People got habitual to the websites like Amazon, Flipkart, Book My Show etc for regular life purchases and entertainments. Products with good rating are most preferably bought by users. A recommender system can help users to choose the products which they can use without any issue in future.

A recommender system predicts the "rating" or "preference" that a user would give to an item. For example, Mr. John and his friend wants to go for a dinner. Mr. John looks for a nearby restaurant and found 5 restaurants. But now the question is which one to choose among the 5 as Mr. John have not visited any of them yet. This is where a recommender system could come handy in real life scenario. The recommender system will select the restaurant with good ratings and also which matches a person's preference.

A lot of research has been done in past on collaborative filtering method. Though there are many advances, the recommender systems still needs further improvements so that it can be more effective and applicable to many real-life applications. Applications like recommending vacation, certain types of financial services to investors, smart shopping cart etc [7].

E-commerce websites has transformed alot because of recommender system. Few years back, the product websites were static i.e. they gave information about the products and their purchase procedure. But nowadays, websites have became dynamic as they give recommendation of products to users based on their previous purchases and their search history. This has changed the looks of E-commerce business alot.

1.2 Movie Recommender System

A movie recommender system predicts the "rating" or "preference" that a user would give to a movie. A movie recommender system can tell whether a person will like a particular movie or not based on his likes. It is a Regression problem where Movie Ratings are predicted which will be given by a user. The fundamental assumption in movie recommender system is that if user A and B have N items whose ratings are similar than there are chances that they will act similarly for other items. [7]

1.3 Types of Movie Recommender System

1.3.1 Collaborative Movie Recommender System

Collaborative recommender systems predicts the ratings of items for a particular user based on the items previously rated by other users. In this type of recommender system, targeted user's nearest neighbour are found who has already rated the targeted item. On the basis of similarity, the rating is calculated for the target item for the target user.

1.3.2 Content-Based Movie Recommender System

Content-based filtering methods are based on a description of the item and profile of the users' preference like actors, directors, genres etc. In this type of content-based filtering, the user's preferences are kept in mind while calculating the rating for the target item. Similar movies are found on the basis of actors, directors, genres etc to find the similarity between the targeted movies and their ratings are calculated on the basis of this similarity.

1.3.3 Hybrid Movie Recommender System

It is combination of Collaborative and Content-Based, in which both techniques can be used in any order. In this, we may first predict the ratings of items for a particular user

based on the items previously rated by other users along with keeping in mind the user's taste like actors, directors, genres etc. We may also filter the movies according to the user's preferences first and then try to predict the ratings of items for a particular user based on the items previously rated by other users or users with similar taste.

1.4 Problem Statement

In recommender system, the prediction of the rating must be accurate with minimal error ratio. The thesis focuses on predicting rating of movies with minimum error. It aims to study impact of different features and at the same time explores approaches for systematic fusion of UbCF and IbCF.

In next chapter, first the introduction of collaborative filtering is given. Next section contains the types of collaborative filtering explanation in detail.

Chapter 2

Collaborative Filtering

In this chapter, first the introduction of collaborative filtering is given. Next section contains the types of collaborative filtering explanation in detail.

2.1 Introduction

Collaborative filtering is one of the techniques used in Recommender systems. In this, we predict the ratings of the target item by finding its neighbour user/item. For e.g. if a user wants to watch a movie, he can trust his friends. The friend having similar taste in movies is preferred over the friend having a less similar taste. So based on the similarity between the target user/item and the neighbour user/item the rating is calculated. Similarity measure popularly used are Pearson Correlation and Cosine-based similarity.

2.2 Types of Collaborative Filtering

2.2.1 User Based Collaborative Filtering (UBCF)

In user-based collaborative filtering, we find target user neighbours who have rated this item and predict rating for the target item. There are two types of UBCF, one is Memory Based and other is Model-Based. Here, user-user similarity is found for calculating the ratings. In Memory Based, the entire collection of the previously rated items is used and the rating of unknown is found using the aggregate of the ratings of some other users. In Model-Based, the collection of ratings is used to create one Model first, and then that model is used to predict the unknown ratings. Commonly used techniques are Bayesian

classifiers, Clustering, Decision Trees, Artificial neural networks.

In this, the rating of items for a particular user based on the items previously rated by other users are predicted. Here, the unknown rating r for user c and item s is normally calculated using some aggregate of ratings of some other users for the same item s. In User-Based Collaborative Filtering implementation, first we need to identify the Neighbour users of Target user who have rated the target movie. Here, the neighbour is the user who has rated the similar item. Later, the similarity between the neighbour is found using either Pearson Correlation or Cosine Based Similarity. The target item ratings are computed using the Aggregation Function.

- **Weighted Average Function that are used are[7] :**

$$r_{c,s} = \frac{1}{N} \sum_{c' \in C} r_{c',s} \tag{2.1}$$

$$r_{c,s} = k \sum_{c' \in C} sim(c, c') \times r_{c',s} \tag{2.2}$$

$$r_{c,s} = \bar{r_c} + k \sum_{c' \in C} sim(c, c') \times (r_{c',s} - \bar{r_{c'}}) \tag{2.3}$$

The value of k and similarity (Pearson Correlation) in above equation is calculated using the following equation.

Calculation of k[7] :

$$k = 1/\sum_{c' \in C} |sim(c, c')| \tag{2.4}$$

Pearson Correlation[7] :

$$sim(x, y) = \frac{\sum_{s \in S_{xy}} (r_{x,s} - \bar{r_x})(r_{y,s} - \bar{r_y})}{\sqrt{\sum_{s \in S_{xy}} (r_{x,s} - \bar{r_x})^2 \sum_{s \in S_{xy}} (r_{y,s} - \bar{r_y})^2}} \tag{2.5}$$

In the following Table 2.1, we show the matrix of User Item Matrix. Here, rows are the Users and columns are the Items. Here, the value shows the rating given by User to a particular Item. Here, Question Mark (?) show the target item for which we need to find the rating which will be given by user U2. So, to predict the rating of item I2 that will be given by user U1, we need to find the nearest neighbour of user U1. So, in this case,

we have three users U2,U4,U5. We find the similarity between them using the formula of Pearson Correlation. After that weighted average function as described in Figure **??** will be used to find the rating.

- **User-Item Matrix :**

	I1	I2	I3	I4
U1	4	?	5	5
U2	4	2	1	
U3	3		2	4
U4	4	4		
U5	2	1	3	5

Table 2.1: User-Item Matrix[8]

- **Calculation of Rating[8] :**

$$P_{1,2} = \bar{r_1} + \frac{\sum_u (r_{u,2} - \bar{r_u}) * w_{1,u}}{\sum_u |w_{1,u}|} \tag{2.6}$$

$$P_{1,2} = \bar{r_1} + \frac{(r_{2,2} - \bar{r_2})w_{1,2} + (r_{4,2} - \bar{r_4})w_{1,4} + (r_{5,2} - \bar{r_5})w_{1,5}}{|w_{1,2}| + |w_{1,4}| + |w_{1,5}|} \tag{2.7}$$

$$P_{1,2} = 4.67 + \frac{(2 - 2.5)(-1) + (4 - 4)(0) + (1 - 3.33)(0.756)}{1 + 0 + 0.756} \tag{2.8}$$

$$P_{1,2} = 3.95 \tag{2.9}$$

2.2.2 Item Based Collaborative Filtering (IBCF)

In item-based collaborative filtering, we find set of items target user has rated and find similarity between them to predict the rating. There are two types of IBCF, one is Memory-Based and second is Model-Based. Here, item-item similarity is found for calculating the ratings. In Memory-Based we use the entire collection of the previously rated items and the rating of unknown is found using the aggregate of the ratings of some other users. In Model-Based, the collection of ratings is used to create one Model first, and then that model is used to predict the unknown ratings. Commonly used techniques are Bayesian classifiers, Clustering, Decision Trees, Artificial neural networks.

In item-based implementation, it looks into the set of items the user has rated and computes how similar they are to the target item i and then selects k most similar items. To find similar items Similarity between the items is computed using Pearson Correlation or Cosine Based Similarity. Once the most similar items are found, the prediction is done using a weighted average of the target users ratings on the similar items.

Item-based collaborative filtering estimates rating of the test user for the test item based on ratings given by test user to other items with similar profile. Items with similar profile are popularly known as nearest neighbours/neighbouring items. Nearest neighbouring items can be found in several ways. To find similarity/correlation between items i_1 and i_2, Pearson correlation is used and it is computed using Equation 2.10 [29].

$$sim(i_1, i_2) = \frac{\sum_{u \in U} (x_{u,i_1} - \overline{x}_{i_1})(x_{u,i_2} - \overline{x}_{i_2})}{\sqrt{\sum_{u \in U} (x_{u,i_1} - \overline{x}_{i_1})^2 \sum_{u \in U} (x_{u,i_2} - \overline{x}_{i_2})^2}} \tag{2.10}$$

Here, U designates a set of users. These users are the ones who have rated both i_1 and i_2. Average rating of item i_1 is represented by $\overline{x}_{i_1}$. Once, this similarity values are computed, user i's rating for the item j can be computed in different ways, and in this paper, Equation 2.11 is used for this task.

$$x_{i,j} = \overline{x}_j + \frac{\sum_{i' \in \hat{I}} sim(j, i') \times (x_{i,i'} - \overline{x}_{i'})}{\sum_{i' \in \hat{I}} |sim(j, i')|} \tag{2.11}$$

Here, $\hat{I}$ denotes set of N items. These items are the ones which have been rated by user i and are the most similar to item j.

In next chapter, Literature survey is given in detail.

Chapter 3

Literature Survey

Academia and the industry both have witnessed development of many collaborative filtering systems. Perhaps, the first recommender system was Grundy system [28] which used streotypes for building user models. These models used very less information about each user. Collaborative filtering algorithms were first used by GroupLens [26, 21], Video Recommender [19] and Ringo [30] to automate predictions. Book recommendation system used by amazon.com was also based on collaborative filtering. An interesting example was a tag recommender discussed in Yagnik et al. which was based on concepts from model-based collaborative filtering [32]. Number of different recommender systems were presented in a special issue of Communications of the ACM [27]. Examples of user-based collaborative filtering includes [14, 18, 20, 26] while [17, 29] are significant examples of item-based collaborative filtering. Patel et al. explored both user-based and item-based collaborative filtering [24]. They explored different possibilities through which user and item profiles can be formulated and predictions can be made.

Advances in collaborative filtering were further discussed in [22]. They discussed about utilization of implicit feedback and temporal models to improve model accuracy. Rao et al. approached collaborative filtering with graph information for low rank matrix completion [25]. They formulated and derived a highly efficient conjugate gradient based minimizing scheme. Recently, Liu et al. proposed a kernalized matrix factorization for collaborative filtering which had distinct advantages over conventional matrix factorization method [23].

Content-based and collaborative filtering were combined in a hybrid approach in [16]. Their hybrid approach was based on Bayesian network. A hybrid content-based and item-

based collaborative filtering approach for recommending TV programs was proposed in [13]. They used singular value decomposition for enhancement. Wang et al. attempted to unify user-based and item-based collaborative filtering approaches through similarity fusion [31]. Their fusion framework was probabilistic in nature. This paper also attempts to combine estimations from user-based and item-based collaborative filtering, but in a much natural way. The idea is to investigate fruitfulness of such fusion. Positive results can be encouraging and can also motivate research community to address the problem with more sophisticated methods.

In "Comparison of Various Metrics Used in Collaborative Filtering for Recommendation System" [1] the author has worked on both IBCF and UBCF. The model used was Memory Based solving the regression problem of predicting the ratings. The evaluation measure used was confusion matrix. The author has used Pearson, Cosine, Euclidian formula to find the similarity. Dataset used is MovieLens.

In "ItemBased Collaborative Filtering Recommendation Algorithms" [2] the author has worked on IBCF. The model used was Model Based solving the regression problem of predicting the ratings. The evaluation measure used was mean absolute error(MAE). The author has used Pearson, Cosine formula to find the similarity. Dataset used is MovieLens.

In "A Collaborative Filtering Recommendation Algorithm Based on Dynamic and Reliable Neighbors" [3] the author has worked on IBCF. The model used was Memory Based solving the regression problem of predicting the ratings. The evaluation measure used was mean absolute error(MAE). The author has used Pearson formula to find the similarity. Dataset used is MovieLens.

In "Improved Collaborative Filtering Recommendations via Non Commonly Rated Items" [4] the author has worked on UBCF. The model used was Memory Based solving the regression problem of predicting the ratings. The evaluation measure used was mean absolute error(MAE) and Root Mean square error(RMSE). The author has used Pearson formula to find the similarity. Dataset used is MovieLens.

In "The Research of Modified Collaborative Filtering Recommendation Algorithm" [5] the author has worked on IBCF. The model used was Memory Based solving the regression problem of predicting the ratings. The evaluation measure used was mean absolute error(MAE). The author has used Improved similarity formula to find the similarity.

Dataset used is MovieLens.

In "An Recommendation Algorithm Based on Weighted Slope One Algorithm and User-Based Collaborative Filtering" [6] the author has worked on UBCF. The model used was Memory Based solving the regression problem of predicting the ratings. The evaluation measure used was mean absolute error(MAE). The author has used Pearson,Cosine formula to find the similarity. Dataset used is MovieLens.

So, from the above papers we come to know that they have worked on simple rating prediction using aggregate function. They have focused to change similarity formula and rating formula for prediction. So, we got the direction that none of the research have done experiment on fusion of IBCF and UBCF. From this literature survey we got motivation that fusion of IBCF and UBCF can be used to study prediction accuracy.

In next chapter, first brief details regarding the proposed Fusion approach of UBCF and IBCF is given. Also, the block diagram for the Fusion approach is shown for better understanding. After that, explanation of different Fusion Techniques is given which is carried out in this research.

Title	UB/IB	MB/MOD	REG/CLASS	EVAL. MEA	SIM.	Rating Formula	Dataset
Comparison of Various Metrics Used in Collaborative Filtering for Recommendation System[1]	Both	MB	REG	Confustion Matrix	Pearson,Cosine,Euclidean	Weighted Score	Movielens
ItemBased Collaborative Filtering Recommendation Algorithms[2]	IB	MOD	REG	MAE	Pearson,Cosine	Weighted sum of IB	Movielens
A Collaborative Filtering Recommendation Algorithm Based on Dynamic and Reliable Neighbors[3]	IB	MB	REG	MAE	Pearson	Weighted sum of UB	Movielens
Improved Collaborative Filtering Recommendations via Non Commonly Rated Items[4]	UB	MB	REG	MAE,RMSE	Pearson	Weighted sum of UB	Movielens
The Research of Modified Collaborative Filtering Recommendation Algorithm[5]	IB	MB	REG	MAE	Improved Similarity Formula	-	Movielens
An Recommendation Algorithm Based on Weighted Slope One Algorithm and User-Based Collaborative Filtering[6]	UB	MB	REG	MAE	Pearson Cosine	Weighted Slope	Movielens

Table 3.1: Literature Survey

Chapter 4

Fusion of UBCF and IBCF

4.1 Introduction

In this chapter, first brief details regarding the proposed Fusion approach of UBCF and IBCF is given. Also, the block diagram for the Fusion approach is shown for better understanding. After that, explanation of different Fusion Techniques is given which is carried out in this research.

4.2 Fusion Approach

UBCF and IBCF are two popular variants of collaborative filtering. UBCF focuses on finding nearest neighbours to the target user and then uses ratings assigned to the target item by these nearest users to predict target user's rating for the target item. Similarly, the focus of IBCF is on finding nearest neighbours to the target item and then use ratings given by the target user to these nearest items to estimate the target user's rating for the target item. In Fusion of UBCF and IBCF, the ratings of both UBCF and IBCF are used that were calculated before and apply different Fusion Technique to finally predict the ratings. The block diagram 4.1 shows the evidence of Fusion Approach.

4.3 Fusion Technique

Following are the Fusion Technique that we have used to predict the final ratings :

4.3.1 Simple Average

In this Fusion technique, we take the calculated values of both UBCF and IBCF rating value and compute average of the ratings. The average value is the final rating in this

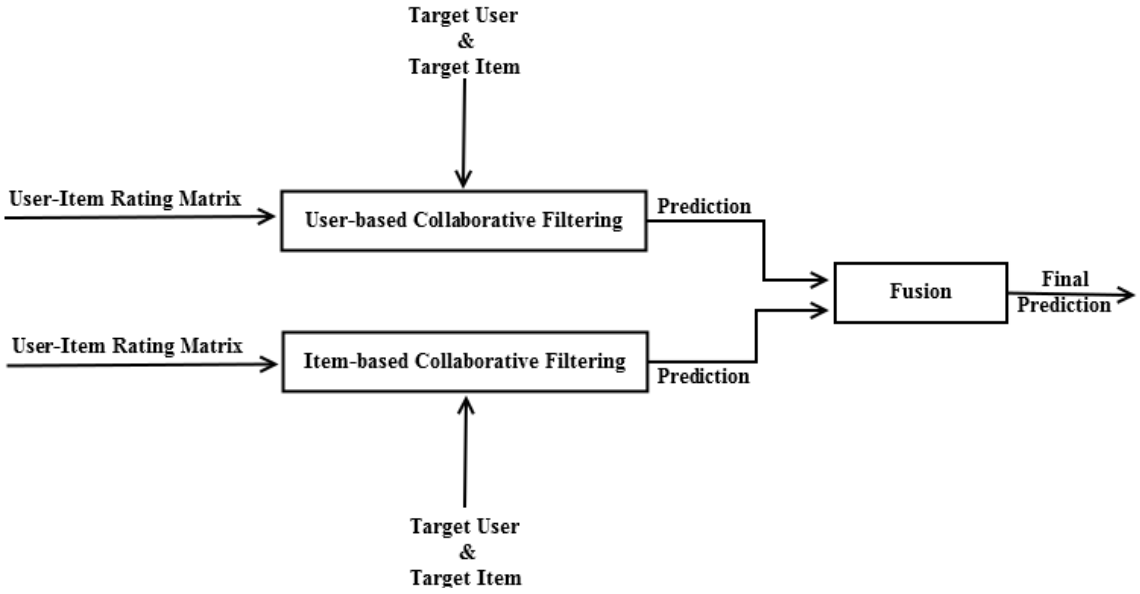


Figure 4.1: Fusion Approach

technique. The final prediction is calculated as defined in Equation 4.1 where $w_1 = w_2 = 0.5$.

$$x_{i,j} = w_1 \times x_{(i,j)(UbCF)} + w_2 \times x_{(i,j)(IbCF)} \tag{4.1}$$

4.3.2 Weighted Average

In this method, the ratings are calculated by taking a weighted average of the ratings which were calculated using IBCF and UBCF. For this, 5-Cross fold is used to get the values of w_1 and w_2 to be used in weighted average formula 4.1. It uses weighted averaging and optimal weight values are decided through performance of user-based and item-based collaborative filtering during five-fold cross validation of training set. Mean Absolute Percentage Error (MAPE) is employed as the performance measure during cross-validation. MAPE is first mapped on the scale of 0 to 1 and then w_1 and w_2 are decided using Equations 4.2 and 4.3 respectively.

$$w_1 = \frac{1 - MAPE_{UbCF}}{(1 - MAPE_{UbCF}) + (1 - MAPE_{IbCF})} \tag{4.2}$$

$$w_2 = \frac{1 - MAPE_{IbCF}}{(1 - MAPE_{UbCF}) + (1 - MAPE_{IbCF})} \tag{4.3}$$

where, $MAPE_{UbCF}$ and $MAPE_{IbCF}$ are MAPE of user-based and item-based collaborative filtering respectively during cross-validation.

4.3.3 Genetic Algorithm

A genetic algorithm (GA), is a method for solving optimization problems based on a natural selection process that mimics biological evolution. The algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm selects individuals from the current population and uses them as parents to produce the children for the next generation. Over successive generations, the population "evolves" toward an optimal solution.

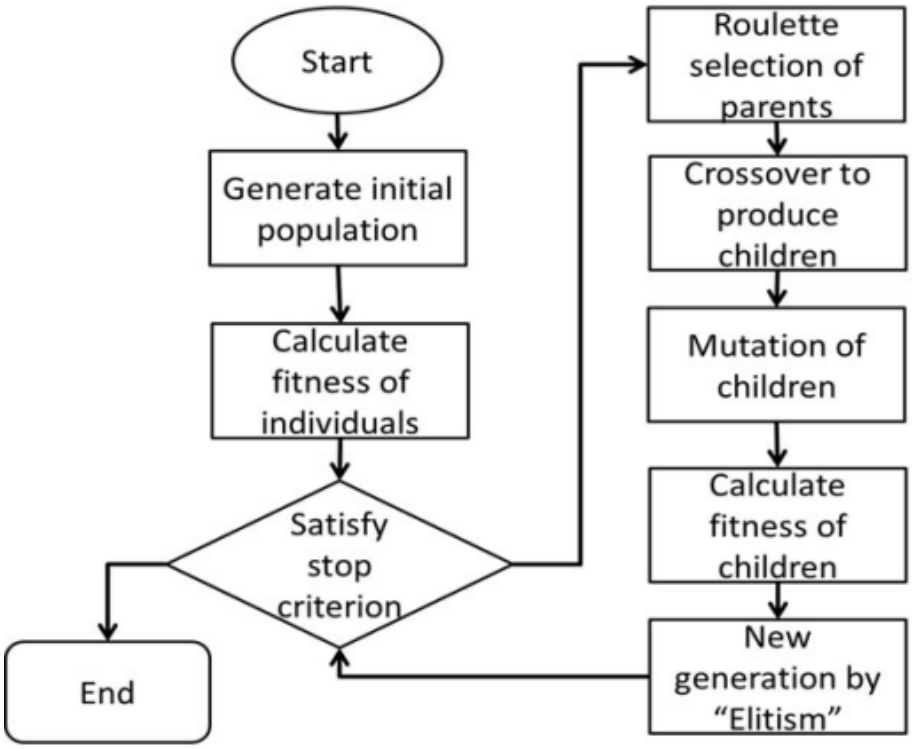


Figure 4.2: Genetic Algorithm Experimental Flow Chart

- **Genetic Algorithm Proposed Approach :**

  1. **Population :** Here, we have generated 50 random numbers as initial population in pair form [w1,w2] whose sum is equal to 1.
  2. **Chromosome :** Chromosome is the single population value in GA. To decide how to represent our problem values in to chromosome is a big task to make GA successful. Our main goal was to get optimized value of w1 and w2. So, w1 and w2 values are taken as chromosome in pair ie "w1w2"
  3. **Gene :** It is the position index in chromosome.

4. **Allele :** It is the value at particular gene in chromosome
5. **Fitness Function :** We have calculated ratings on basis of population and calculated RMSE as fitness value.
6. **Genetic Operators :**
   - Selection (Roulette Wheel Selection)
   - Crossover (BLX-Alpha Crossover)
   - Mutation (Uniform Mutation)

We have used Roulette Wheel Selection as the selection algorithm which is described here.

Algorithm 1: Roulette Wheel Selection[9]

Let i=1, where i denotes the chromosome index ;
Calculate $\varphi_s(x_i(t)) = f_\Upsilon(x_i(t))/\sum_{l=1}^{n_s} f_\Upsilon(x_l(t))$ where $f_\Upsilon(x_i(t)) = f_{max} - f(x_i(t))$;
sum=$\varphi_s(x_i)$;
Choose r=U(0,1) ;
while *sum ¡ r* **do**
    i=i+1,i.e advance to the next chromosome;
    sum=sum+$\varphi_s(x_i)$;
end
Return x_i as the selected individual;

We have used BLX-Alpha Crossover as the Crossover algorithm which is described

here. Here, the α value taken is 0.5 for the BLX-Alpha Crossover.

Algorithm 2: BLX-Alpha Crossover[10][11]

Select two parents $X^{(t)}$ and $Y^{(t)}$ from a parent pool;
α - positive real number;
Create two offspring $X^{(t+1)}$ and $Y^{(t+1)}$ as follows :
for *i=1 to n* **do**
    $d_i = |x_i^{(t)} - y_i^{(t)}|$;
    choose a uniform random real number u from interval
    ¡ $\min(x_i^{(t)}, y_i^{(t)}) - \alpha * d_i, max(x_i^{(t)}, y_i^{(t)}) + \alpha * d_i >$
    $x_i^{(t+1)} = u$
    choose a uniform random real number u from interval
    ¡ $\min(x_i^{(t)}, y_i^{(t)}) - \alpha * d_i, max(x_i^{(t)}, y_i^{(t)}) + \alpha * d_i >$
    $y_i^{(t+1)} = u$
end

For mutation in Genetic Algorithm we have used Uniform Mutation. The algorithm is described here.

Algorithm 3: Uniform Mutation[9]

p_m=0.1;
for $i = 1$ *to* n_x **do**
    if $U(0,1) < p_m$ **then**
        $x_i = x_i + 0.5$
    end
end

Genetic algorithm is run for 50 generations and best value of w1 and w2 are than used to calculate the ratings using weighted average formula.

4.3.4 Generational Genetic Algorithm

In this, we will use the concept of Genetic Algorithm that we have described in Section 4.3.3, but the new offsprings will replace all the parents for the new generation. This is done for 50 generations and best value w1 and w2 are then used to calculate the ratings using weighted average formula.

4.3.5 Genetic Algorithm with Tournament Selection

In this, we will use the concept of Genetic Algorithm that we have described in Section 4.3.3, but we have used here Tournament Selection for selecting the new Parent for the upcoming generation. In this, we have given equal chance to all the new offspring to be parent for the upcoming generation. This is done for 50 generations and best value w1 and w2 are then used to calculate the ratings using weighted average formula.

4.3.6 Genetic Algorithm with 0.25 Mutation Rate

In this, we will use the concept of Genetic Algorithm that we have described in Section 4.3.3, but we have changed the mutation rate to 0.25 which was earlier 0.1. This is done for 50 generations and best value w1 and w2 are then used to calculate the ratings using weighted average formula.

4.3.7 Genetic Algorithm with Population size 100

In this, we will use the concept of Genetic Algorithm that we have described in Section 4.3.3, but we have changed the initial population value to 100 which was 50 earlier.This is done for 50 generations and best value w1 and w2 are then used to calculate the ratings using weighted average formula.

4.3.8 Decision Tree

Decision Tree is a flow chart like structure where each node is a decision making condition and each branch is the outcome if the condition is satisfied. In data mining decision tree is used for Classification and Regression purpose. We have used CART here to predict the ratings. We have used the Matlab function classregtree to build the model using the training set data created using the 5-crossfold. Then the model is used to predict the final rating for the testing data. The prediction value is calculated for 1NN to 100NN one by one.

4.3.9 Random Forest

Decision Tree as discussed in Section 4.3.8 has the problem of over-fitting [33]. So Random Forest is the method which uses ensemble learning in which multiple decision tree are created using the training data. After that the prediction is done for the testing data by each decision tree in the forest and average value is taken as the final rating in our case.

We have used the Matlab function TreeBagger with value 50 to create the model with 50 decision tree. Then the model is used to predict the final rating for the testing data. The prediction value is calculated for 1NN to 100NN one by one.

4.3.10 Linear Support Vector Regression

In machine learning, Support Vector Machine is a supervised learning where the model is built by first using the training set. After creating the model it is used for classification or regression. Here we have used Linear Support Vector Regression as we need to predict the ratings of the movie. We have used the Matlab function svmtrain and svmpredict to do regression[34]. The kernel used is linear kernel. The prediction value is calculated for 1NN to 100NN one by one.

4.3.11 Linear Regression

Regression is the process in statistics where we try to find the relationship between the two variables mainly dependent variable and independent variable. Once the relationship is found we can predict the new value from the created model of regression[35]. In Linear regression, here we try to create model using the training data which is around 7000 of IBCF and UBCF. Once the model is created we predict the ratings and calculate the MAPE for the same.

In this chapter, first details regarding the dataset used in the experiment is explained. Also, the statistics regarding the dataset is given. After that, description of all the methods used in the experiment is given. Finally, the MAPE results are been shown with there graphs.

Chapter 5

Experimental Evaluation and Results

In this chapter, first details regarding the dataset used in the experiment is explained. Also, the statistics regarding the dataset is given. After that, description of all the methods used in the experiment is given. Finally, the MAPE results are been shown with there graphs.

5.1 Dataset

5.1.1 Details

- The dataset that we have used here for implementation is **hetrec2011-movielens-2k**
- This dataset is an extension of MovieLens10M dataset, published by GroupLeans research group. http://www.grouplens.org
- The dataset is released in the framework of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011) http://ir.ii.uam.es/hetrec2011 at the 5th ACM Conference on Recommender Systems (RecSys 2011).

5.1.2 Statistics

The following is the brief statistics regarding the Dataset used :

- 2113 users
- 10197 movies
- 20 movie genres
- 20809 movie genre assignments, avg. 2.040 genres per movie
- 4060 directors
- 95321 actors, avg. 22.778 actors per movie
- 72 countries
- 10197 country assignments, avg. 1.000 countries per movie
- 47899 location assignments, avg. 5.350 locations per movie
- 855598 ratings, avg. 404.921 ratings per user, avg. 84.637 ratings per movie

5.2 Experimental Evaluation

5.2.1 Description

We have found from the dataset the users who have rated number of movies between 100 and 120. In this, we found 87 such users from the dataset. Using that users, the Training set was created by masking the ratings of this 87 users by taking 25 random movies and Rating was calculated for the masked movies using the following methods :

- **UBCF :**

  In this method, first the nearest neighbour of the target user is found who have rated the target movie. After tha, the ratings of that nearest neighbour is stored in an array. The similarity between the target user and nearest neighbour is calculated. Average ratings of nearest neighbour are calculated for its use in the UBCF formula. After that sorting on basis of similarity is done for final calculation of rating. Now using the Weighted average function the ratings is calculated on the basis of the value we give for nearest neighbour. Here, we have calculated for 12 different NN values. The results are shown in Table 5.1. Here MAPE value has minimum 0.255 value, when result is taken for 30 and 50 nearest neighbour. Here, MAPE value has increased gradually after 30 NN. The evidence of it can be seen in Figure 5.1

- **IBCF and Variants of IBCF :**

  1. **IBCF**

     In this method, first the nearest neighbour of the target item is found which has been rated by the target user. After that, the ratings of that nearest neighbour item is stored in array. The similarity between the target movie and nearest neighbour movie is calculated. Average ratings of nearest neighbour is calculated for its use in the formula. After that sorting on basis of similarity is done for final calculation of rating. Now using the Weighted average function the ratings is calculated on the basis of the value we give for nearest neighbour.Here, we have calculated for 12 different NN values. The results are shown in Table 5.2. Here MAPE value has minimum 0.277 value, when result is taken for 10 nearest neighbours. Here, MAPE value has increased gradually after 10 NN.The evidence of it can be seen in Figure 5.1

  2. **IBCF Variant 1**

     In this method, it is similar to the above IBCF method described. The only change is that here on the basis of the genre of the movie the nearest neighbour is found for the target movie. And using this nearest neighbour the rest thing is calculated as in IBCF. The results are shown in Table 5.3. Here, MAPE value has minimum 0.271 value, when result is taken for 10 nearest neighbours. Here, MAPE value has increased gradually after 10 NN. The result is lower than the normal IBCF but higher than UBCF. The evidence of it can be seen in Figure 5.1.

  3. **IBCF Variant 2**

     In this method the similarity between the target movie and nearest neighbour is calculated using the genre of the movie. Rest all the calculation is done same as done in IBCF method. The results are shown in Table 5.4. Here, MAPE value has minimum 0.283 value, when result is taken for 10 nearest neighbours. Here, MAPE value has increased gradually after 10 NN. The result, is higher than the normal IBCF results, IBCF Variant 1 and UBCF results. The evidence of it can be seen in Figure 5.1

4. **IBCF Variant 3**

   In this method, the similarity between the target movie and nearest neighbour is calculated using the genre and ratings of training matrix.Rest all the calculation is done same as done in IBCF method. The results are shown in Table 5.5. Here, MAPE value has minimum 0.284 value, when result is taken for 20 nearest neighbours. Here, MAPE value has increased gradually after 20 NN. The result, is higher than the normal IBCF results, IBCF Variant 1, IBCF Variant 2 and UBCF results. The evidence of it can be seen in Figure 5.1

5. **IBCF Variant 4**

   In this method, the similarity between the target movie and nearest neighbour is calculated using the genre,ratings of training matrix and actors of movie. Rest all the calculation is done same as done in IBCF method. The results are shown in Table 5.6. Here, MAPE value has minimum 0.285 value, when result is taken for 10 nearest neighbours. Here, MAPE value has increased gradually after 10 NN. The result, is higher than the normal IBCF results, IBCF Variant 1, IBCF Variant 2, IBCF Variant 3 and UBCF results. The evidence of it can be seen in Figure 5.1

- **Fusion Approaches**

1. **Fusion through Simple Average**

   In this method we have calculated the ratings by taking simple average of the ratings which we have calculated using IBCF and UBCF. The results are shown in Table 5.7. Here, MAPE value has minimum 0.246 value, when result is taken for 10 nearest neighbours. Here, MAPE value has increased gradually after 10 NN. The result, is lower than the normal IBCF results, IBCF Variant 1, IBCF Variant 2, IBCF Variant 3, IBCF Variant 4 and UBCF results. The evidence of it can be seen in Figure 5.2

2. **Fusion through Weighted Average**

   In this method we have calculated the ratings by taking weighted average of the ratings which we have calculated using IBCF and UBCF. For this we have done 5-Cross fold to get the values of W1 and W2 to be used in weighted average formula. The results are shown in Table 5.8. Here, MAPE value has

minimum 0.246 value, when result is taken for 10 nearest neighbours. Here, MAPE value has increased gradually after 10 NN. The result, is lower than the normal IBCF results, IBCF Variant 1, IBCF Variant 2, IBCF Variant 3, IBCF Variant 4 and UBCF results. The evidence of it can be seen in Figure 5.2

3. **Genetic Algorithm**

   In this method, we have calculated the ratings by using genetic algorithm. In this we have chosen w1 and w2 pair as chromosome values. 50 Initial population is generated randomly whose sum is equal to 1. The algorithm is run for 50 generation and best value of each generation is calculated. The 50th Best value of w1 and w2 is than used in Fusion through Weighted Average and ratings are calculated. The results are shown in Table 5.9. Here, MAPE value has minimum 0.255 value, when result is taken for 50 nearest neighbours. Here, MAPE value has increased gradually after 50 NN. The result is almost similar to UBCF,less than IBCF and more than weighted average . The evidence of it can be seen in Figure 5.2

4. **Generational Genetic Algorithm**

   In this, we will use the concept of Genetic Algorithm that we have described above, but the new offsprings will replace all the parents for the new generation. The 50th Best value of w1 and w2 is than used in Fusion through Weighted Average and ratings are calculated. The results are shown in Table 5.10. Here, MAPE value has minimum 0.245 value, when result is taken for 10 nearest neighbours. Here, MAPE value has increased gradually after 10 NN. The result is almost similar to weighted average. The evidence of it can be seen in Figure 5.2

5. **Genetic Algorithm with Tournament Selection**

   In this, we will use the concept of Genetic Algorithm that we have described in above, but we have used here Tournament Selection for selecting the new Parent for the upcoming generation. The 50th Best value of w1 and w2 is than used in Fusion through Weighted Average and ratings are calculated. The results are shown in Table 5.11. Here, MAPE value has minimum 0.254 value, when result is taken for 50 nearest neighbours. Here, MAPE value has

increased gradually after 50 NN. The result is almost similar to UBCF,less than IBCF and more than weighted average . The evidence of it can be seen in Figure 5.2

6. **Genetic Algorithm with 0.25 Mutation Rate**

   In this, we will use the concept of Genetic Algorithm that we have described above, but we have changed the mutation rate to 0.25 which was earlier 0.1.The 50th Best value of w1 and w2 is than used in Fusion through Weighted Average and ratings are calculated. The results are shown in Table 5.12. Here, MAPE value has minimum 0.255 value, when result is taken for 50 nearest neighbours. Here, MAPE value has increased gradually after 50 NN. The result is almost similar to UBCF,less than IBCF and more than weighted average . The evidence of it can be seen in Figure 5.2

7. **Genetic Algorithm with initial Population 100**

   In this, we will use the concept of Genetic Algorithm that we have described above, but we have changed the initial population value to 100 which was 50 earlier. The 50th Best value of w1 and w2 is than used in Fusion through Weighted Average and ratings are calculated. The results are shown in Table 5.13. Here, MAPE value has minimum 0.254 value, when result is taken for 50 nearest neighbours. Here, MAPE value has increased gradually after 50 NN. The result is almost similar to UBCF,less than IBCF and more than weighted average . The evidence of it can be seen in Figure 5.3

8. **Decision Tree**

   We have used the matlab function classregtree to build the model using the training set data created using the 5-crossfold. Then the model is used to predict the final rating for the testing data. The prediction value is calculated for 1NN to 100NN one by one. The results are shown in Table 5.14. Here, MAPE value has minimum 0.285 value, when result is taken for 20 nearest neighbours. Here, MAPE value has increased gradually after 20 NN. The result is higher than UBCF, IBCF and weighted average . The evidence of it can be seen in Figure 5.3

9. **Random Forest**

   We have used the matlab function TreeBagger with value 50 to create the

model with 50 decision tree. Then the model is used to predict the final rating for the testing data. The prediction value is calculated for 1NN to 100NN one by one. The results are shown in Table 5.15. Here, MAPE value has minimum 0.251 value, when result is taken for 20 nearest neighbours. Here, MAPE value has increased gradually after 20 NN. The result is lower than UBCF, IBCF and higher than weighted average . The evidence of it can be seen in Figure 5.3

10. **Linear Support Vector Regression**

    We have used the matlab function svmtrain and svmpredict to do regression. The kernel used is linear kernel. The prediction value is calculated for 1NN to 100NN one by one. The results are shown in Table 5.16. Here, MAPE value has minimum 0.161 value, when result is taken for 20 nearest neighbours. Here, MAPE value has increased gradually after 20 NN. The result is lower than UBCF, IBCF and weighted average. The evidence of it can be seen in Figure 5.3

11. **Linear Regression**

    We have used the matlab function regress to do linear regression. The prediction value is calculated for 1NN to 100NN one by one. The results are shown in Table 5.17. Here, MAPE value has minimum 0.166 value, when result is taken for 20 nearest neighbours. Here, MAPE value has increased gradually after 20 NN. The result is lower than UBCF, IBCF and weighted average. The evidence of it can be seen in Figure 5.3

5.3 Results

5.3.1 UBCF Results

	NN											
	1	2	5	10	20	30	50	60	70	80	90	100
MAPE	0.337	0.299	0.27	0.26	0.256	**0.255**	**0.255**	0.256	0.257	0.258	0.258	0.258

Table 5.1: UBCF Results

5.3.2 IBCF Results

	NN											
	1	2	5	10	20	30	50	60	70	80	90	100
MAPE	0.32	0.305	0.285	**0.277**	0.278	0.278	0.281	0.282	0.284	0.285	0.285	0.285

Table 5.2: IBCF Results

5.3.3 IBCF Variant 1 Results

	NN											
	1	2	5	10	20	30	50	60	70	80	90	100
MAPE	0.307	0.294	0.274	**0.271**	0.273	0.273	0.277	0.278	0.278	0.278	0.278	0.278

Table 5.3: IBCF Variant 1 Results

5.3.4 IBCF Variant 2 Results

	NN											
	1	2	5	10	20	30	50	60	70	80	90	100
MAPE	0.329	0.296	0.284	**0.283**	0.284	0.284	0.284	0.286	0.288	0.29	0.291	0.291

Table 5.4: IBCF Variant 2 Results

5.3.5 IBCF Variant 3 Results

	NN											
	1	2	5	10	20	30	50	60	70	80	90	100
MAPE	0.311	0.297	0.297	0.288	**0.284**	0.287	0.286	0.287	0.288	0.289	0.29	0.29

Table 5.5: IBCF Variant 3 Results

5.3.6 IBCF Variant 4 Results

	NN											
	1	2	5	10	20	30	50	60	70	80	90	100
MAPE	0.311	0.289	0.288	**0.285**	0.286	0.289	0.289	0.29	0.291	0.291	0.291	0.291

Table 5.6: IBCF Variant 4 Results

5.3.7 Fusion through Simple Average Results

	NN											
	1	2	5	10	20	30	50	60	70	80	90	100
MAPE	0.278	0.263	0.25	**0.246**	0.247	0.248	0.249	0.25	0.251	0.252	0.252	0.253

Table 5.7: Fusion through Simple Average Results

5.3.8 Fusion through Weighted Average Results

	NN											
	1	2	5	10	20	30	50	60	70	80	90	100
MAPE	0.278	0.263	0.25	**0.246**	0.247	0.248	0.249	0.25	0.251	0.252	0.252	0.253

Table 5.8: Fusion through Weighted Average Results

5.3.9 Fusion through Genetic Algorithm Results

	NN											
	1	2	5	10	20	30	50	60	70	80	90	100
MAPE	0.336	0.299	0.269	0.259	0.256	**0.255**	**0.255**	0.256	0.256	0.257	0.257	0.258

Table 5.9: Fusion through Genetic Algorithm Results

5.3.10 Fusion through Generational Genetic Algorithm Results

	NN											
	1	2	5	10	20	30	50	60	70	80	90	100
MAPE	0.277	0.262	0.250	**0.245**	0.246	0.247	0.249	0.250	0.251	0.252	0.252	0.252

Table 5.10: Fusion through Generational Genetic Algorithm Results

5.3.11 Fusion through Genetic Algorithm with Tournament Selection Results

	NN											
	1	2	5	10	20	30	50	60	70	80	90	100
MAPE	0.336	0.298	0.269	0.259	0.256	0.255	**0.254**	0.256	0.256	0.257	0.257	0.257

Table 5.11: Fusion through Genetic Algorithm with Tournament Selection Results

5.3.12 Fusion through Genetic Algorithm with 0.25 Mutation Rate Results

	NN											
	1	2	5	10	20	30	50	60	70	80	90	100
MAPE	0.336	0.299	0.269	0.259	0.256	**0.255**	**0.255**	0.256	0.256	0.257	0.257	0.257

Table 5.12: Fusion through Genetic Algorithm with 0.25 Mutation Rate Results

5.3.13 Fusion through Genetic Algorithm with Population size 100 Results

	NN											
	1	2	5	10	20	30	50	60	70	80	90	100
MAPE	0.333	0.299	0.269	0.259	0.256	0.255	**0.254**	0.256	0.256	0.257	0.257	0.257

Table 5.13: Fusion through Genetic Algorithm with Population size 100 Results

5.3.14 Fusion through Decision Tree Results

	NN											
	1	2	5	10	20	30	50	60	70	80	90	100
MAPE	0.322	0.326	0.303	0.292	**0.285**	0.299	0.289	0.298	0.294	0.292	0.290	0.298

Table 5.14: Fusion through Decision Tree Results

5.3.15 Fusion through Random Forest Results

	NN											
	1	2	5	10	20	30	50	60	70	80	90	100
MAPE	0.285	0.279	0.257	0.252	**0.251**	0.254	0.255	0.254	0.257	0.253	0.254	0.254

Table 5.15: Fusion through Random Forest Results

5.3.16 Fusion through Linear Support Vector Regression Results

	NN											
	1	2	5	10	20	30	50	60	70	80	90	100
MAPE	0.173	0.169	0.165	0.162	**0.161**	**0.161**	**0.161**	**0.161**	0.162	0.163	0.163	0.163

Table 5.16: Fusion through Linear Support Vector Regression Results

5.3.17 Fusion through Linear Regression Results

	NN											
	1	2	5	10	20	30	50	60	70	80	90	100
MAPE	0.181	0.175	0.170	0.167	**0.166**	**0.166**	**0.166**	0.167	0.167	0.168	0.168	0.168

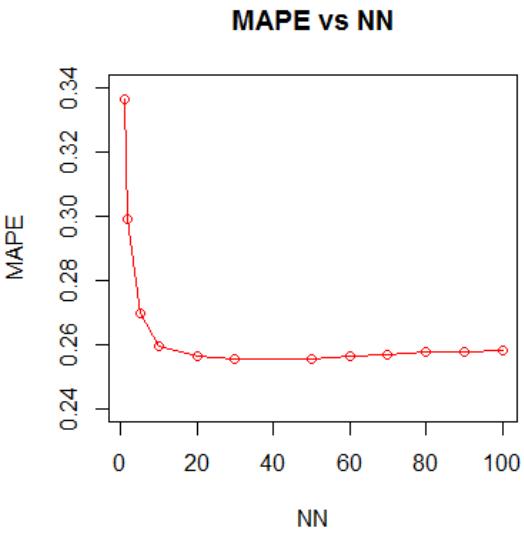
Table 5.17: Fusion through Linear Regression Results

5.3.18 Summary of MAPE Results

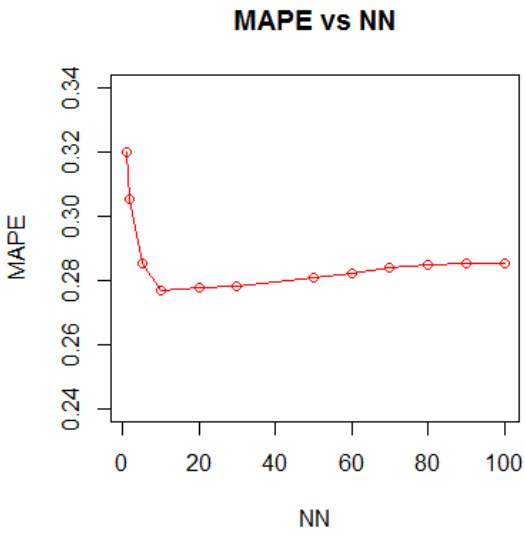
The summary of MAPE results for all the methods are shown in Table 5.19.

NN												
Methods	1	2	5	10	20	30	50	60	70	80	90	100
UBCF	0.337	0.299	0.27	0.26	0.256	**0.255**	**0.255**	0.256	0.257	0.258	0.258	0.258
IBCF	0.32	0.305	0.285	**0.277**	0.278	0.278	0.281	0.282	0.284	0.285	0.285	0.285
IBCF Variant 1	0.307	0.294	0.274	**0.271**	0.273	0.273	0.277	0.278	0.278	0.278	0.278	0.278
IBCF Variant 2	0.329	0.296	0.284	**0.283**	0.284	0.284	0.284	0.286	0.288	0.29	0.291	0.291
IBCF Variant 3	0.311	0.297	0.297	0.288	**0.284**	0.287	0.286	0.287	0.288	0.289	0.29	0.29
IBCF Variant 4	0.311	0.289	0.288	**0.285**	0.286	0.289	0.289	0.29	0.291	0.291	0.291	0.291
Fusion through Simple Average	0.278	0.263	0.25	**0.246**	0.247	0.248	0.249	0.25	0.251	0.252	0.252	0.253
Fusion through Weighted Average	0.278	0.263	0.25	**0.246**	0.247	0.248	0.249	0.25	0.251	0.252	0.252	0.253
Fusion through Genetic Algorithm	0.336	0.299	0.269	0.259	0.256	**0.255**	**0.255**	0.256	0.256	0.257	0.257	0.258
Fusion through Generational Genetic Algorithm	0.277	0.262	0.250	**0.245**	0.246	0.247	0.249	0.250	0.251	0.252	0.252	0.252
Fusion through Genetic Algorithm with Tournament Selection	0.336	0.298	0.269	0.259	0.256	0.255	**0.254**	0.256	0.256	0.257	0.257	0.257
Fusion through Genetic Algorithm with 0.25 Mutation Rate	0.336	0.299	0.269	0.259	0.256	**0.255**	**0.255**	0.256	0.256	0.257	0.257	0.257
Fusion through Genetic Algorithm with Population size 100	0.333	0.299	0.269	0.259	0.256	0.255	**0.254**	0.256	0.256	0.257	0.257	0.257
Fusion through Decision Tree	0.322	0.326	0.303	0.292	**0.285**	0.299	0.289	0.298	0.294	0.292	0.290	0.298
Fusion through Random Forest	0.285	0.279	0.257	0.252	**0.251**	0.254	0.255	0.254	0.257	0.253	0.254	0.254
Fusion through Linear Support Vector Regression	0.173	0.169	0.165	0.162	**0.161**	**0.161**	**0.161**	**0.161**	0.162	0.163	0.163	0.163
Fusion through Linear Regression	0.181	0.175	0.170	0.167	**0.166**	**0.166**	**0.166**	0.167	0.167	0.168	0.168	0.168

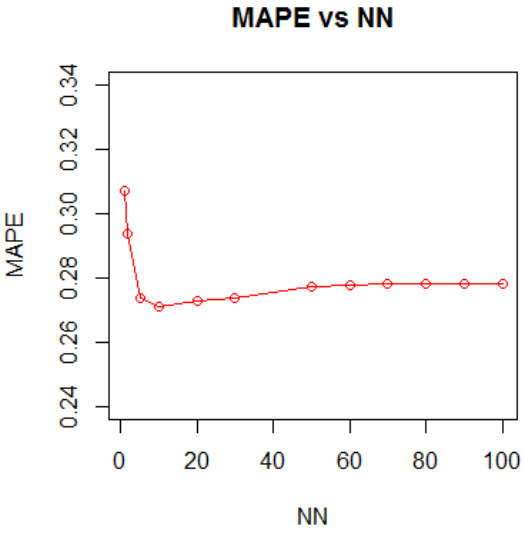
Table 5.18: Summary of MAPE Results



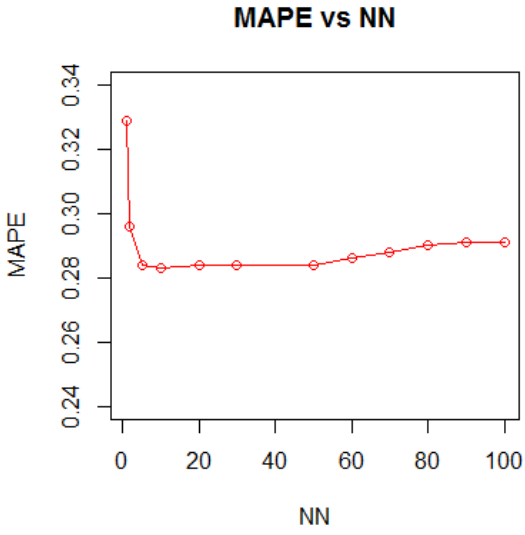
(a) UBCF



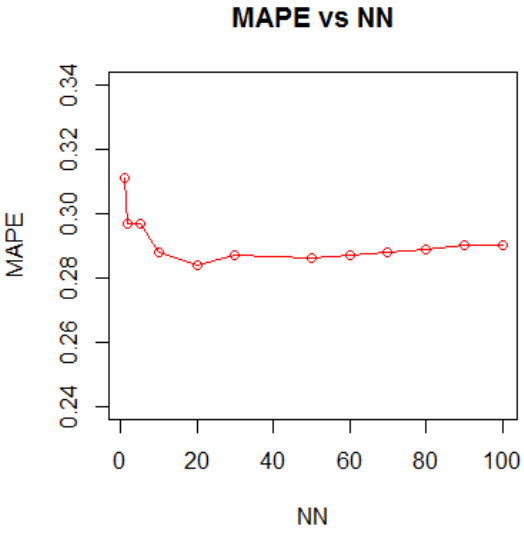
(b) IBCF



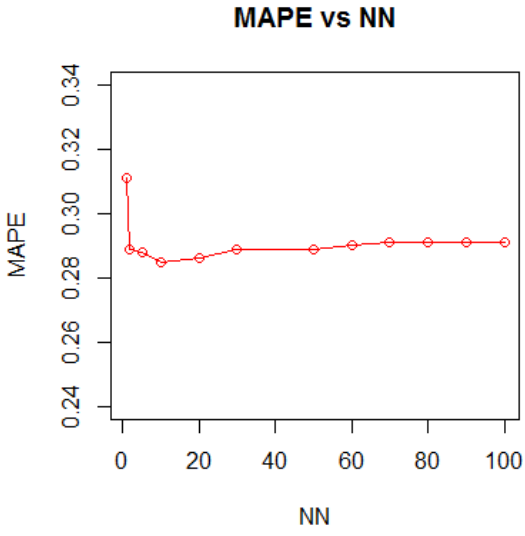
(c) IBCF Variant 1



(d) IBCF Variant 2

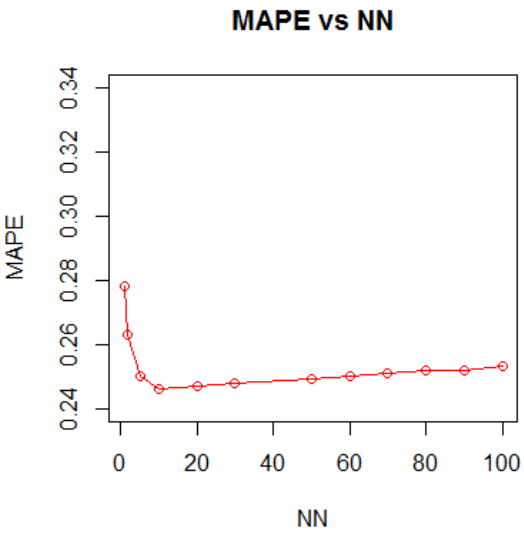


(e) IBCF Variant 3

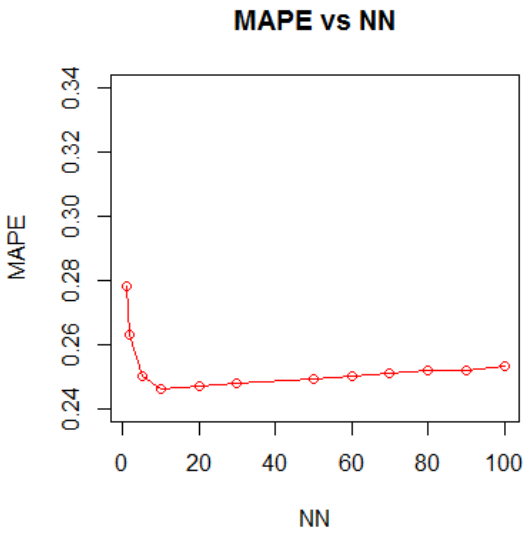


(f) IBCF Variant 4

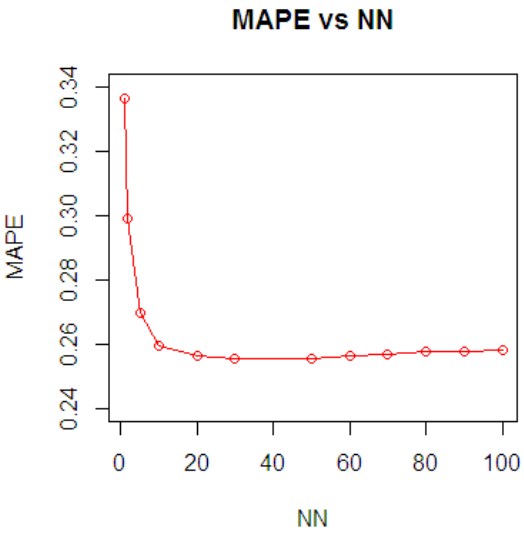
Figure 5.1: MAPE Results for different Methods



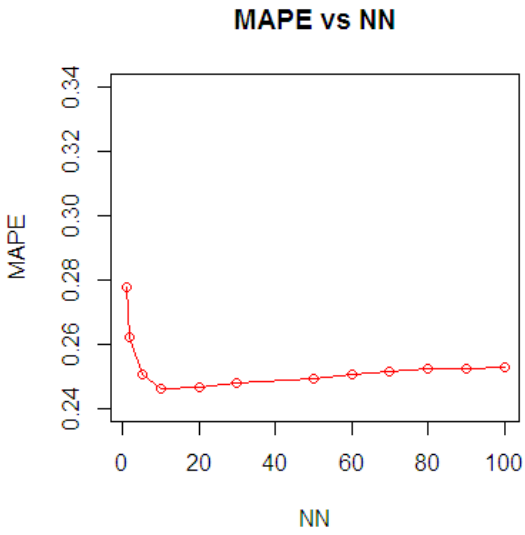
(a) Fusion through Simple Average



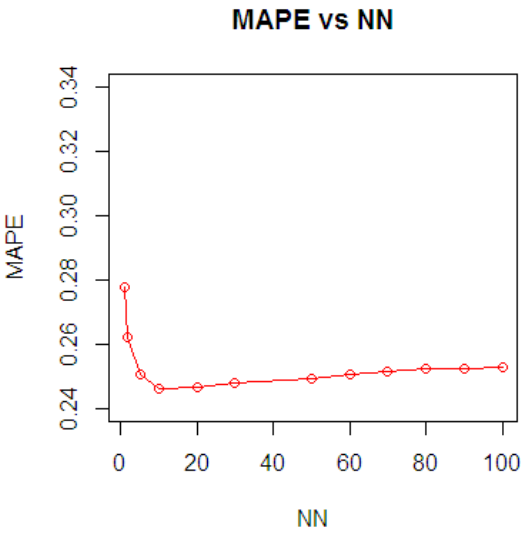
(b) Fusion through Weighted Average



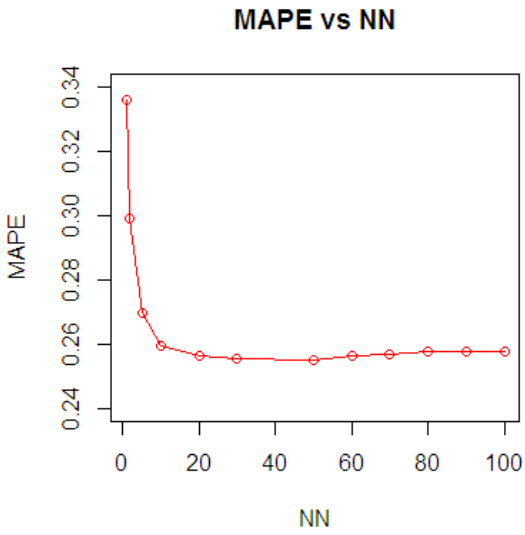
(c) Genetic Algorithm(GA)



(d) Generational Genetic Algorithm

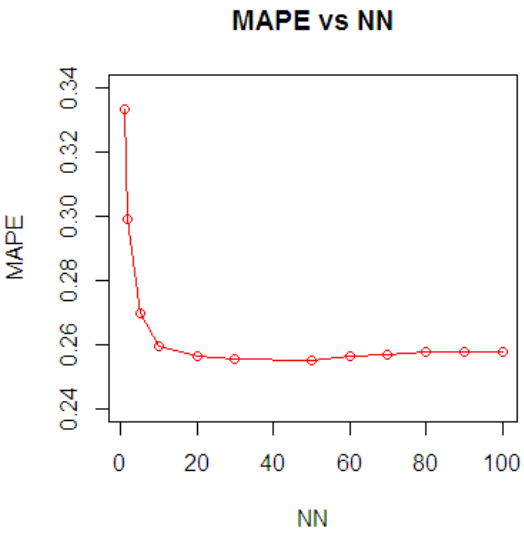


(e) GA with Tournament Selection

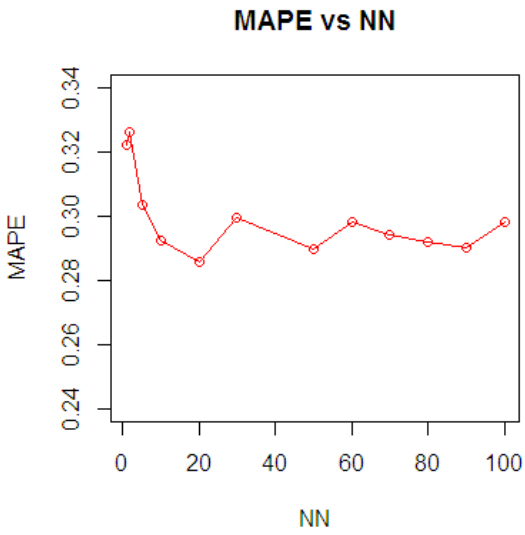


(f) GA with 0.25 Mutation Rate

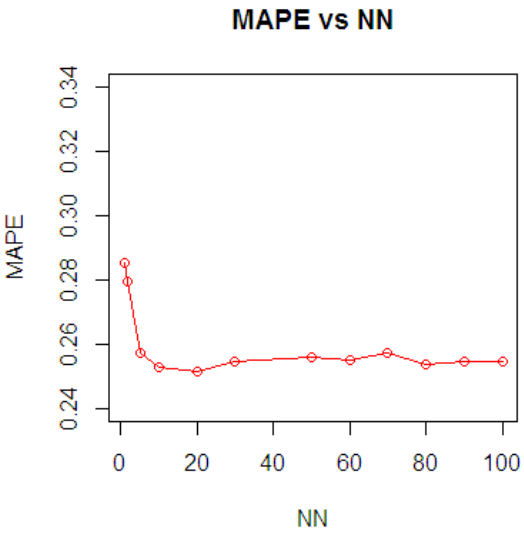
Figure 5.2: MAPE Results for different Methods



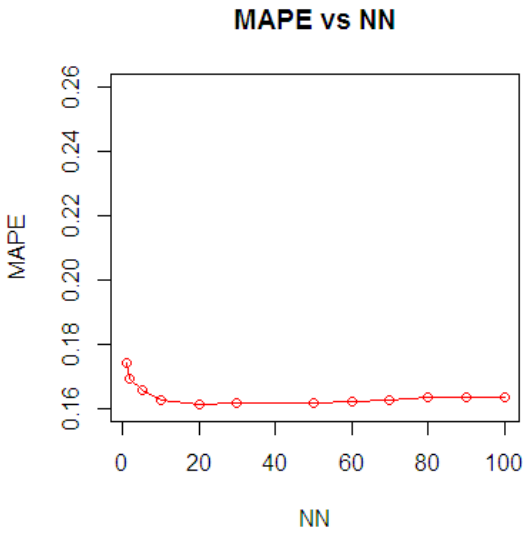
(a) GA with Population 100



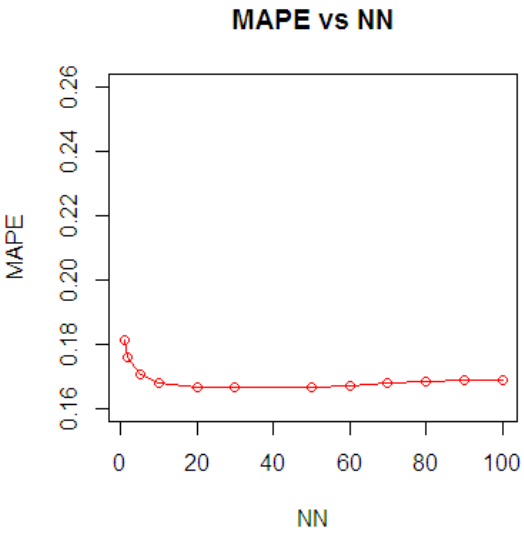
(b) Decision Tree



(c) Random Forest



(d) Linear Support Vector Regression



(e) Linear Regression

Figure 5.3: MAPE Results for different Methods

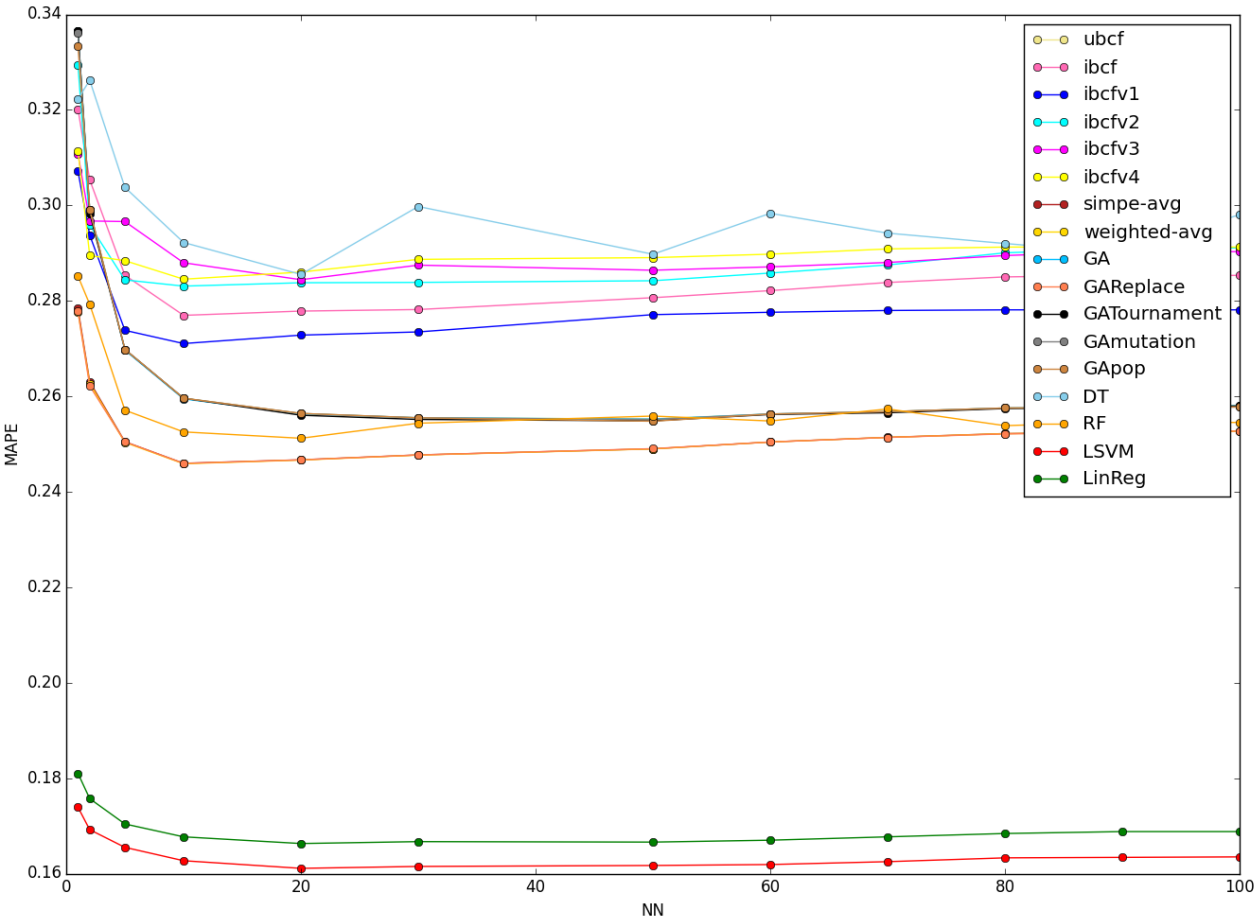


Figure 5.4: Summary of MAPE Result

5.3.19 Performance Comparison among individual and significant Fusion Approaches

The summary of MAPE results for UBCF, IBCF, Fusion through Linear Support Vector Regression and Fusion through Linear Regression are shown in Table 5.19.

	NN											
Methods	1	2	5	10	20	30	50	60	70	80	90	100
UBCF	0.337	0.299	0.27	0.26	0.256	**0.255**	**0.255**	0.256	0.257	0.258	0.258	0.258
IBCF	0.32	0.305	0.285	**0.277**	0.278	0.278	0.281	0.282	0.284	0.285	0.285	0.285
Fusion through Linear Support Vector Regression	0.173	0.169	0.165	0.162	**0.161**	**0.161**	**0.161**	**0.161**	0.162	0.163	0.163	0.163
Fusion through Linear Regression	0.181	0.175	0.170	0.167	**0.166**	**0.166**	**0.166**	0.167	0.167	0.168	0.168	0.168

Table 5.19: Performance Comparison among individual and significant Fusion Approaches

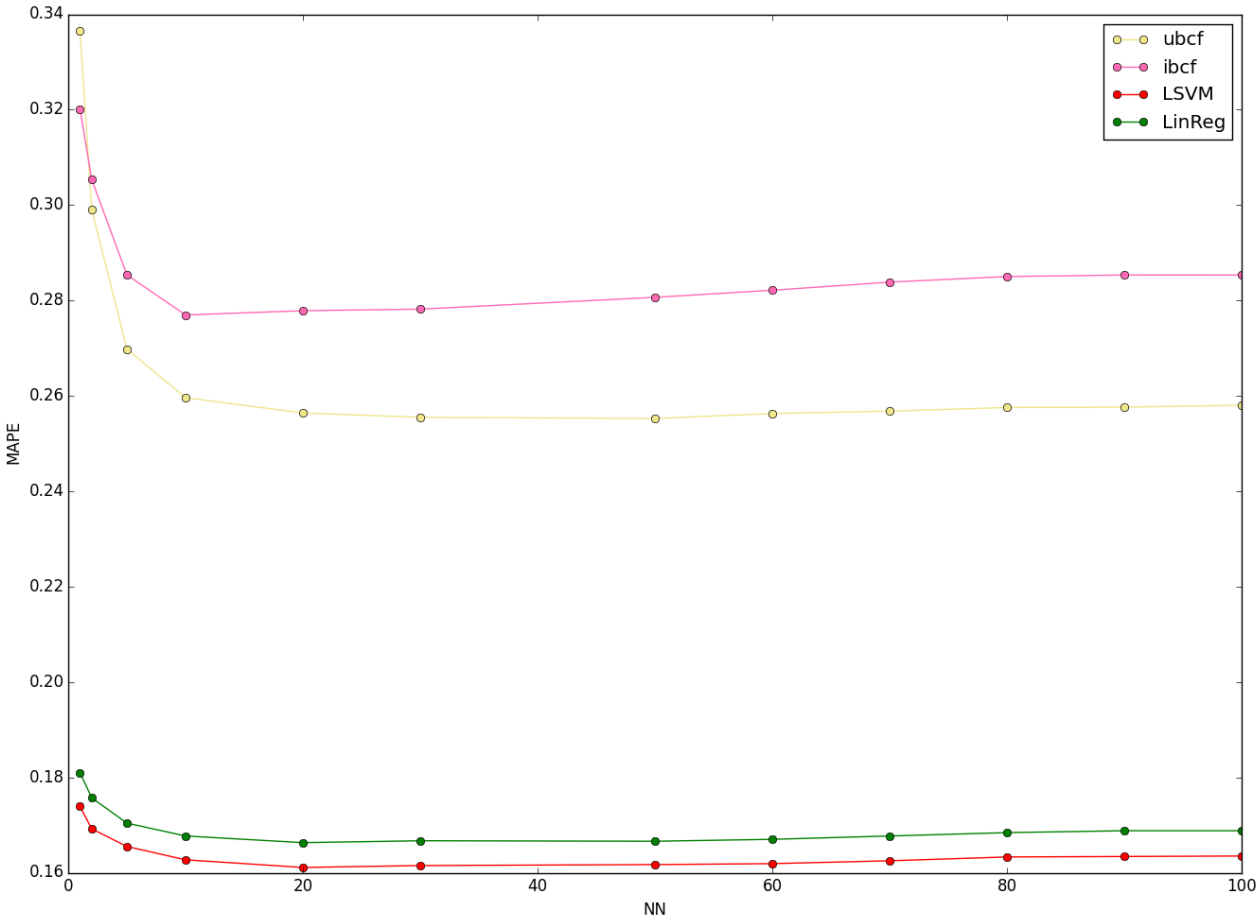


Figure 5.5: Performance Comparison among individual and significant Fusion Approaches

Chapter 6

Conclusion and Future Work

In this dissertation, user-based collaborative filtering and item-based collaborative filtering are used for predicting ratings of unseen movies. Apart from ratings, movie specific data such as genre and star-cast are also used and their impact on prediction accuracy is studied. Fusion of user-based and item-based collaborative filtering techniques is also exercised for predicting movie ratings. Fusion of UBCF and IBCF has been carried out through simple as well as sophisticated approaches. Simple approaches include simple and weighted averaging while sophisticated approaches involve algorithms such as GA, CART, RF, LR and SVR.

It is observed that fusion approaches performs better than individual approaches and in particular fusion through SVR performs the best. This is highly encouraging and motivating. A very useful direction for future is to use these fusion approaches on other datasets of different domain. This can also help in checking robustness of the proposed fusion approaches.

Bibliography

[1] Comparison of Various Metrics Used in Collaborative Filtering for Recommendation System,Anuranjan Kumar,Sahil Gupta,S. K Singh,K. K. Shukla,2015 IEEE

[2] ItemBased Collaborative Filtering Recommendation Algorithms,Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl,WWW10, May 15,2001, Hong Kong.ACM 1581133480/01/0005

[3] A Collaborative Filtering Recommendation Algorithm Based on Dynamic and Reliable Neighbors,Shang Zheng, YongJun Shen, GuiDong Zhang, YiYu Gao,2015 IEEE

[4] Improved Collaborative Filtering Recommendations via Non Commonly Rated Items,Weijie Cheng,Guisheng Yin,Yuxin Dong,Hongbin Dong,Wansong Zhang,2015 Eighth International Conference on Internet Computing for Science and Engineering

[5] The Research of Modified Collaborative Filtering Recommendation Algorithm,YU Zhenhai,FANG Yonghao,ZHANG Yikun, LIU Shufen,2015 7th International Conference on Information Technology in Medicine and Education

[6] An Recommendation Algorithm Based on Weighted Slope One Algorithm and User-Based Collaborative Filtering,WANG Panpan, QIAN Qian, SHANG Zhenhong, LI Jingsong,2016 IEEE

[7] Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions,Gediminas Adomavicius and Alexander Tuzhilin,IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 17, NO. 6, JUNE 2005

[8] A Survey of Collaborative Filtering Techniques,Xiaoyuan Su and Taghi M. Khoshgoftaar,Hindawi Publishing Corporation,Advances in Artificial Intelligence,Volume 2009, Article ID 421425

[9] Computational Intelligence-An Introduction by Andries P. Engelbrecht

[10] A Recommender System based on genetic algorithm for music Data,Hyun-Tae Kim, Eungyeong Kim, Jong-Hyun Lee, Chang Wook Ahn,2010,IEEE

[11] `www.tomaszgwiazda.com/blendX.htm`

[12] Adomavicius, G., Tuzhilin, A.: 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 17(6), 734–749 (2005)

[13] Barragáns-Martínez, A.B., Costa-Montenegro, E., Burguillo, J.C., Rey-López, M., Mikic-Fonte, F.A., Peleteiro, A.: A hybrid content-based and item-based collaborative filtering approach to recommend tv programs enhanced with singular value decomposition. Information Sciences 180(22), 4290–4311 (2010)

[14] Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence. pp. 43–52. Morgan Kaufmann Publishers Inc. (1998)

[15] Cantador, I., Brusilovsky, P., Kuflik, T.: Second workshop on information heterogeneity and fusion in recommender systems (hetrec2011). In: RecSys. pp. 387–388 (2011)

[16] De Campos, L.M., Fernández-Luna, J.M., Huete, J.F., Rueda-Morales, M.A.: Combining content-based and collaborative recommendations: A hybrid approach based on bayesian networks. International Journal of Approximate Reasoning 51(7), 785–799 (2010)

[17] Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. ACM Transactions on Information Systems (TOIS) 22(1), 143–177 (2004)

[18] Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd annual international

ACM SIGIR conference on Research and development in information retrieval. pp. 230–237. ACM (1999)

[19] Hill, W., Stead, L., Rosenstein, M., Furnas, G.: Recommending and evaluating choices in a virtual community of use. In: Proceedings of the SIGCHI conference on Human factors in computing systems. pp. 194–201. ACM Press/Addison-Wesley Publishing Co. (1995)

[20] Jin, R., Chai, J.Y., Si, L.: An automatic weighting scheme for collaborative filtering. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 337–344. ACM (2004)

[21] Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., Riedl, J.: Grouplens: applying collaborative filtering to usenet news. Communications of the ACM 40(3), 77–87 (1997)

[22] Koren, Y., Bell, R.: Advances in collaborative filtering. In: Recommender Systems Handbook, pp. 77–118. Springer (2015)

[23] Liu, X., Aggarwal, C., Li, Y.F., Kong, X., Sun, X., Sathe, S.: Kernelized matrix factorization for collaborative filtering. In: SIAM Conference on Data Mining. pp. 399–416 (2016)

[24] Patel, R., Thakkar, P., Kotecha, K.: Enhancing movie recommender system. International Journal of Advanced Research in Engineering and Technology (IJARET), ISSN pp. 0976–6499 (2014)

[25] Rao, N., Yu, H.F., Ravikumar, P.K., Dhillon, I.S.: Collaborative filtering with graph information: Consistency and scalable methods. In: Advances in Neural Information Processing Systems. pp. 2107–2115 (2015)

[26] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM conference on Computer supported cooperative work. pp. 175–186. ACM (1994)

[27] Resnick, P., Varian, H.R.: Recommender systems. Communications of the ACM 40(3), 56–58 (1997)

[28] Rich, E.: User modeling via stereotypes. Cognitive science 3(4), 329–354 (1979)

[29] Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web. pp. 285–295. ACM (2001)

[30] Shardanand, U., Maes, P.: Social information filtering: algorithms for automating word of mouth. In: Proceedings of the SIGCHI conference on Human factors in computing systems. pp. 210–217. ACM Press/Addison-Wesley Publishing Co. (1995)

[31] Wang, J., De Vries, A.P., Reinders, M.J.: Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 501–508. ACM (2006)

[32] Yagnik, S., Thakkar, P., Kotecha, K.: Recommending tags for new resources in social bookmarking system. International Journal of Data Mining & Knowledge Management Process 4(1), 19 (2014)

[33] https://en.wikipedia.org/wiki/Randomforest

[34] https://www.csie.ntu.edu.tw/ cjlin/libsvm/

[35] https://in.mathworks.com/help/stats/regress.html