Using Machine Learning in a Hybrid Recommendation System for Diet Improvement Based on Health and Taste

Yiren Qu

Abstract

Recommendation systems are used everywhere today, such as for online shopping or Netflix videos. The use of these systems gives us the ability to predict the taste of food effectively. In addition, other food concerns, such as food safety and nutritional value have become more important than before. With these thoughts, a novel diet improvement system is proposed in this research, which is based on a recommendation system using hybrid matrix decomposition and K-nearest neighborhood algorithms to help people meet their health goals. There are many existing programs to recommend a healthy diet. Unfortunately, most systems recommend foods that are radically different from a person's typical diet, making it difficult to adjust to and likely that the person will reject the diet and not meet their health goals. The results of an additional psychological study of induced motivation are incorporated into the system to generate a model enabling each user to gradually accept healthier foods into his/her diet. The need to minimize error in predicting taste requires the evaluation of a combination of different algorithms. In addition, it also requires optimization of different parameters of the algorithms to determine the most efficient and accurate way to recommend diet changes. The system contributes a new application of the hybrid optimized recommendation system on gradually diet improvement in order to satisfy user's health goals.

1 Background & Introduction

1.1 Background

Food, similar to online advertisement services, the prediction of the prices of products, and the weather forecast, is one of the most important major components in people's lives and has become the focus of development of a number of computer applications by that would enable the analysis of information. Under modern pressure, people seem to lose concern about their own eating. There are many existing software to recommend a healthy diet. Unfortunately, most systems recommend foods that are radically different from a person's typical diet, making it difficult to adjust to and likely that the person will reject the diet and not meet their health goals. In the near future, personal data of preferred food habits will connect to restaurants, home computers, and mobile devices including cars. People eating at restaurants will receive immediate information containing food recommendations based on their personal health condition and an optimized taste model. In a long term, the recommendation system would enable people to specify their own settings to make their diets healthier and easy to accept by using the induced motivation method, recommending tasty food to give user a rest time to get used to it.

The recommendation system, a hybrid technique of machine learning and data mining, is at the core of this system. It attempts to predict information items which, such as movies and web search, are likely to be of interest to the user. It has been successfully used in various fields of e-commerce and information science. For example, online shopping companies and research engines are using this technique to recommend related products or websites to users. Technically, based on known data, such as the scores awarded to items and the types of items, a personal profile and model is constructed. The recommendation system may compare a user profile to other profiles and scores with some similar factors to predict the ratings users would assign to items they have not yet evaluated. In general, the recommendation system continuously performs large-scale data mining and applies the machine-learning model.

1.2 Brief Introduction

In this paper, we propose a new type of system based on a recommendation system to help users to complete their goal according to an induced motivation model generated by a personal goal in health. Using multiple techniques in modern recommendation systems and data mining, the system is based on partial ratings of items by users and focuses on a personalized taste model. In terms of taste, the system predicts ratings: (I) by using a modified *K-nearest neighborhood* (KNN) algorithm to analyze similar user scores and similar food scores, (II) by *singular value decomposition* (SVD) in one of the matrix decomposition methods to compare the users' preferences on latent factors and similarity between foods and latent factors to predict users' rating on foods. In terms of health, the model develops a personal goal based on the user's weight and nutrition need, and then analyzes the nutritional value of food to obtain a number representing how suitable for a particular user to reach the goal in health. In this research, we evaluated this system's accuracy with virtual test and human participants' test to prove the validity of the system.

After briefly introducing the background against which this whole idea was conceived and further developed, the ideal prototype of this system with its main technique, the recommendation system, the remaining part of this research paper is organized as follows: Section 2 presents the general problems and factors affecting the system and concludes with the goal of the system; Section 3 describes the design of this system in detail and the introduction of data collection prerequisites, as well as the problems that were encountered during design; the virtual test result and real test result are demonstrated in Section 4. Finally, the conclusion summarizes the current results and presents future work.

2 Problem Formulation

2.1 General Problem for Diet Improvement System and Goal

This system aims to find a solution to recommend food that is both healthy and tasty for users with the hope that, in the long term, the diet may become healthier because of the average summation of taste and health by the personal model. Specifically, assume there are N_u users and N_f foods, given a set of training data, which is a set of triples containing columns: user id, food id, and ratings, define the $A_{u,f}$ $"user - food\ matrix"$ containing only known food ratings and $B_{u,f}$ matrix as:

$$\mathrm{A}_{u,f} = \begin{bmatrix} ID_u & ID_f & R_{u,f} \\ \vdots & \vdots & \vdots \end{bmatrix} \quad \text{size of A is } (3, N_a), in\ which\ N_a \leq N_u \times N_f$$

$$\mathrm{B}_{u,f} = \begin{pmatrix} R_{u1,f1} & \cdots & R_{u1,fi} \\ \vdots & \ddots & \vdots \\ R_{ui,f1} & \cdots & R_{ui,f1} \end{pmatrix} \quad \text{size of B is } \left(N_u, N_f\right)$$

The first task is to predict all the users' ratings on given foods based on the training set, in this case the known food ratings set. In this problem, each rating is an integer between 1 and 10, representing from unlike to like, and the goal is to minimize the *root mean square error* (RMSE) of the results of the testing sets containing unknown ratings of food. The training set, error and RMSE, specifically, are defines as:

$$S_{Test} = ([ID_u \quad ID_f], \cdots)\ , in\ which\ ID_u, ID_f \in A_{u,f}\ and\ R_{u,f} \notin A_{u,f}$$

$$\mathrm{E}_{u,f} = R_{u,f} - P_{u,f},$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{|S_{Test}|} \sum_{(u,f) \in S_{Test}} \mathrm{E}_{u,f}{}^{2}},$$

where $|S_{Test}|$ is its cardinality, $R_{u,f}$is the true rating, and $P_{u,f}$ is the predicted rating based on the recommendation system. Gunawardana and Shani, in their contribution on evaluating recommendation systems [1], argued that the RMSE is popular and highly accurate in expressing the performance of a recommendation system.

The second task is to generate a health score based on the comparison between users' needs in terms of basic nutrition and food nutrition. In this problem, each health score is an integer between 1 and 10 and the goal is to sort the predicted food-rating list after obtaining the average summation of the health score and predicted taste score. Specifically, define the health score, result score, and recommended food list as:

$$L_{u,P_f} = \begin{bmatrix} ID_f & P_{u,f} \\ \vdots & \vdots \end{bmatrix} \ (u,f) \in \ S_{Test} \ and \ A_{u,P_f} \ is \ a \ descending \ sorted \ list \ by \ P_{u,f}$$

$$H_{u,f} = \sum_{(u,f)\in} \frac{h_{f,i}}{h_{u,i}} \times a_i \ \left(u, f \in \ L_{u,P_f}\right),$$

$$L_{u,Result} = \left[P_{u,f} \times a_{u,t} + H_{u,f} \times a_{u,h}, \cdots\right] (u,f) \in \ S_{Test},$$

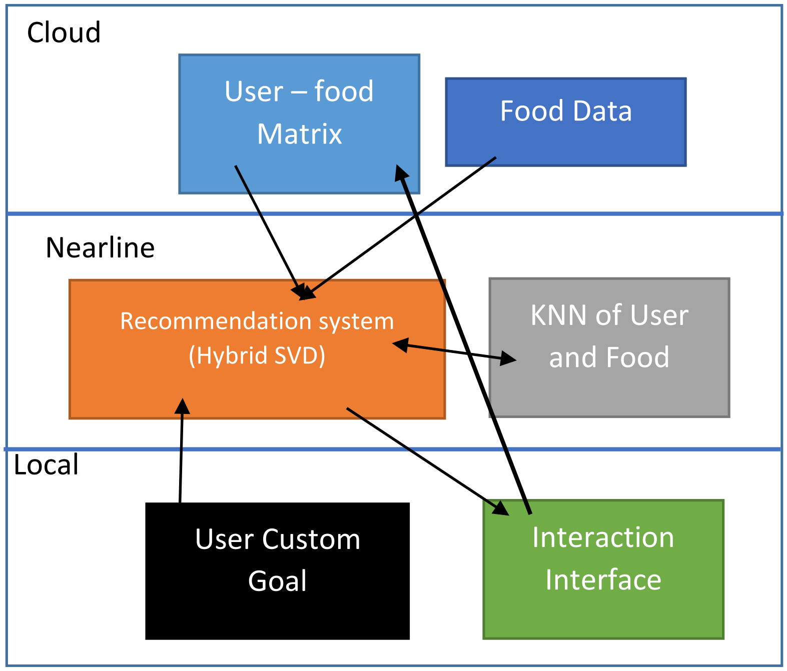
where L_{u,P_f} is a descending sorted list of tuples of predicted food scores, $h_{f,i}$ is the nutritional value of food items, $h_{u,i}$ is the nutritional needs of users, $a_{u,i}$ is the weight of each nutritional component in the weighted summation, and $(a_{u,t}, a_{u,h})$ are the weights generated by the personal goal model consisting of a taste part and health part for each user.

3 System design

3.1 System construction

The diet improvement system is divided into three parts: Cloud data storage and methods, nearline storage and methods, and local storage and methods as depicted in Figure 1.

Figure 1 System Construction



In the system, all databases containing basic information are saved in the Cloud, and specifically, there is a $"user - food \ matrix"$ and Food Dataset containing basic nutritional values, such as number of calories, amount of fat, total amount of carbon dioxides, sodium, and proteins. Nearline, according to [2], is a way of transiting data between Offline and Online. We set up the recommendation system based on a hybrid SVD / KNN method and a database of similar user and similar food generated by the KNN method. The local data consisting of the users' own goal and the induced motivation model generated by the goal are saved offline and are recalled when generating the final recommended food list. The interaction interface displays all the results received from the recommendation system.

3.2 System algorithms

3.2.1 *K-nearest neighbor* (KNN) algorithm

The first approach is the item-based K-nearest neighbor (KNN) algorithm. This algorithm finds users and foods with the highest similarity related to $User_i$ and $Food_j$ to reduce the scale of the dataset used for *singular value decomposition* (SVD) and tries to minimize the error cost of the recommendation system. On the other hand, the KNN algorithm is also used as an improved naïve method compared with the result of the hybrid SVD method. Specifically, the method determines the rating of $User_i$ on $Food_j$ by finding other similar items of $Food_j$ that $User_i$ scored before. Many ways of defining similarity between two users and two food items have their own merits and demerits and their differences are discussed in [3]. In this research, we will use three popular ways to test the error cost and use the best one to compare with the modified SVD method. The similarity functions used in the test are based on Euclidean similarity (1), which calculate the line distance between two points in Euclidean Space, cosine similarity (2), which is the measure of calculating the difference of angle between two vectors, and Pearson similarity (3), which calculates similarity based on the deviation on average ratings. Refer to [3] for more details.

$$\mathrm{sim(x,y)} = \frac{1}{1+\sqrt{\sum(x-y)^2}} \qquad (1)$$

$$\mathrm{sim}(x,y) = \frac{(x \times y)}{\sqrt{x^2 \times y^2}} \qquad (2)$$

$$\mathrm{sim(x,y)} = \frac{(x-\bar{x})(y-\bar{y})}{\sqrt{(x-\bar{x})^2 \times (y-\bar{y})^2}} \qquad (3),$$

where x and y represent the rating $R_{u1,f}$ and $R_{u2,f}$ when KNN is applied to find similar users and the rating $R_{u,f1}$ and $R_{u,f2}$ when KNN is applied to find similar food, and $\bar{x}$ and $\bar{y}$ are the average ratings of x and y.

3.2.2 **Original & Modified *Latent factor method* (LFM) Algorithm**

The core algorithm used in the recommendation system in the diet improvement system is the hybrid single value decomposition (SVD) algorithm. Because matrix B is a sparse matrix of which more than half of the values are unknown, the normal *latent factor model* (LFM) tries to use two matrices to express relationship between users and k latent factors and the relationship between k latent factors and the food items, and the inner product is the predicted value. It factors the matrix B as two smaller matrices X and Y, which represent the relationship between users, foods, and latent factors. The factorization can only be approximate because k, the number of columns in X representing the number of latent factors, is small as shown below quoted from [6]

$$\text{Matrix B} = \begin{pmatrix} R_{u1,f1} & \cdots & R_{u1,f} \\ \vdots & \ddots & \vdots \\ R_{u,f1} & \cdots & R_{u,f} \end{pmatrix} \approx \begin{pmatrix} R_{u1,k1} & R_{u1,k2} & \cdots \\ R_{u2,k1} & R_{u2,k2} & \cdots \\ \vdots & \vdots & R_{u,k} \end{pmatrix} \times \begin{pmatrix} R_{k1,f1} & R_{k1,f2} & \cdots \\ R_{k2,f1} & R_{k2,f2} & \cdots \\ \vdots & \vdots & R_{k,f} \end{pmatrix}^{Transpose}$$

The goal of the LFM method is to find the set of user and item feature vectors that minimizes the squared error to known ratings. Through decomposition on known values, matrices X and Y become sufficiently dense to help the system find the prediction rating. However, generally, there is no solution of $\mathrm{B} = \mathrm{X}Y^T$and the task is to generate an error cost function to minimize the error, see [5] for more detail. The implementation of our method in this system is a modification of that presented in [4] and [7], but we will evaluate the best values produced by overfitting to prevent some deviated value from affecting the whole model. The process consists of two components, namely steps α and β. These steps represent the time to complete the iterations required to fit the model; α is a constant whose value determines the rates at which the minimum is approached; β is used to control the magnitudes of the feature vectors.

Due to the fact that, the smaller scale of dataset using in the LFM method, the better result we may get. Thus, in the approach, we changed the original LFM method's input matrix to a similarity matrix applied KNN method. The new method successfully and radically reduces the scale of matrix B. Therefore, the time efficiency of analyzing and factoring is exponential decay and the scale of the matrix is linear decay. The system defines the matrix by applying the KNN method, which is called the Matrix C $explicit\ similar\ user\ and\ food\ Matrix$, as:

$$Matrix\ C = \begin{pmatrix} R_{u,f1} & \cdots & R_{u,fk} \\ \vdots & \ddots & \vdots \\ R^{k}_{uk,f1} & \cdots & R^{k}_{uk,fk} \end{pmatrix},$$

in which $R^{k}_{uk,fi}$ is the rating of similar user k on similar food k, and this is also a sparse matrix but denser than Matrix B. Further, a user's prediction rating $P_{u,f}$ on a food is given by the result of applying the LFM algorithm to the matrix based on the taste model.

3.2.3 Health score calculation

The final part of the calculation involves generating the health score based on personal information and food nutrition. The idea behind this is to use the weighted summation of the extent to which each basic nutritional component satisfies the user's personal needs. The nutritional components of food are calories, fat, total carbon dioxides, sodium, and proteins. According to [9], the personal need for each basic nutritional component can be calculated according to the basic BMR need. The relevant health equation is defined as:

$$H_{u,f} = \sum_{(u,f)\in} \frac{h_{f,i}}{h_{u,i}} \times a_i$$

where $h_{f,i}$ is the nutritional value of a particular food item, $h_{u,i}$ is the nutritional needs of users, and $a_{u,i}$ is the weight of each nutritional component in the weighted summation.

3.2.3 *Induced Motivation Model* (IMM) based on custom goal

The health score of food is different for different users and even on different days when the aim is to complete a dietary goal or keeping a healthy diet. In the process, a user may have a personal goal, such as losing two kilograms of weight in 30 days, or gaining four kilograms of weight in 25 days to train muscles. According to [8], the ideal perfect model for a user to gradually change their eating diet and eating choice, specifically, should first recommend tastier but less healthy food to the user, after which it would be possible for them to transit to food which is both tasty and healthy. Thus, we use IMM as the model of weight in the weighted summation of the taste part and health part. The mathematics model we are using here is a logistic model, with slight changes to the start and end and with logarithmic growth in the middle to satisfy both time efficiency and allow adjustment time for the user to adapt. In the evaluation, we use information of a virtual person to test whether the system meet the needs. The particular person is a normal healthy18 year-old male student with normal activities with an initial height of 180 centimeters and an initial weight of 75 kilograms, and with the goal of losing two kilograms of weight in 30 days. The custom IMM graph is shown in Figure 2:

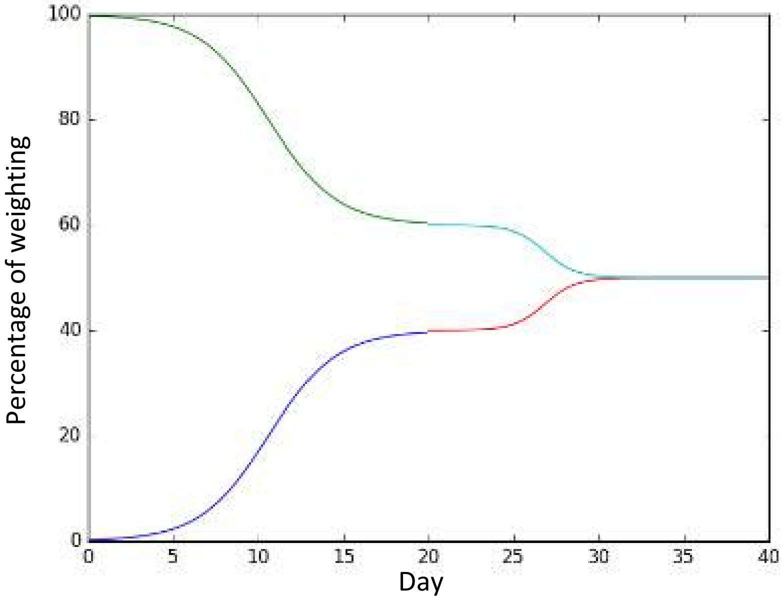


Figure 2 Custom IMM Graph

The model for this user is to reach his goal in 30 days and also allow him an adjustment time. In Figure 2, the upper graph represents the weight of the taste part in weighted summation and the bottom graph represents the weight of the health score: the x- and y-axes represent the number of days required to complete the goal and percentage, respectively. It is clear that the two lines in the graph are approaching 50% at day 30, including two adjustment periods during days 1-5, and days 15-25 when the health weight is about 40%.

3.3 Test and training dataset collection

There are few useable open-source large-scale datasets on food ratings; therefore, a three-week track lunch survey was sent out on school days. A survey listing all the courses served in the cafeteria was distributed online as a Google form. The survey reviewed participants' opinions based only on the taste of the foods they ate for lunch during the specified period expressed as a score ranging from 1 to 10.

The resulting matrix generated from the track survey contains responses from186 people and pertains to 64 courses of food. The respondents consist of 165 students aging from 15 to 18 and 21 adult teachers; thus, the matrix has a size of(186×64). Matrix A contained a total of 1235 entries. The system was evaluated by randomly dividing the original Matrix A into Matrix

$A_{Training}$ and Set S_{Test}. The size of Matrix $A_{Training}$ is (555×3) and the cardinality of the Set test is 212. All of the entries were useable and valuable ranging from 1 to 10 as integers.

3.4 Minor problems and solutions

Because the whole system is an ideal prototype, minor problems were generated such as components of choosing foods, popular problems of recommendation system. In addition, because of the limitations, which cannot do a tracking experiment of human participants, imposed as a result of participants' privacy concerns, it is almost impossible to track a real health condition. Therefore, this research offers some solutions to these problems and limitations.

I. Define a healthy and suitable diet and evaluate the efficiency

Each person may have their own thoughts on the definition of a healthy and suitable diet for them. However, the main idea of a healthy diet is "A healthy diet provides the body with essential nutrition: fluid, adequate essential amino acids from protein, essential fatty acids, vitamins, minerals, and adequate calories" quoted from [10]. According to the definition, using a customized goal for each user is a great way to present personal needs on nutrition. Along with the basic information reflecting a user's health condition, it is also easy to obtain the BMR to count a user's basic requirement in terms of calories. Although the number of research participants was limited, we set up a virtual user for which all of the basic information is specified to test whether the recommended food meets the user's needs.

II. Cold start and new user problems of recommendation system

Cold start and Newcomer are popular problems of recommendation systems. A newcomer, defined as a new user without any known ratings of food, is difficult to predict. We minimized the RMSE by using the average rating of most popular foods as a prediction of rating $P_{u,f}$ as:

$$P_{u,f} = \frac{1}{N_{f_p}} \sum R_{u\prime,f_p},$$

where f_p represents the foods on the most popular list.

4 Evaluation

The system designed in Section 3[1] was implemented on Matrix $A_{Training}$ and Set S_{Test}. In particular, we use four types of algorithms, i.e., naïve program, KNN algorithm, regular LFM method, and modified LFM algorithm, to test the recommendation system to find the best fit for this problem. Each of these algorithms was evaluated by using five pairs of training and test datasets. In the evaluation of the KNN algorithm, we test how k and similar functions affect the error. Our aim is to establish the extent to which overfitting components affect the RMSE and a best combination with k-features and other components. In addition, we test the time required to execute two different LFM methods in order to show their difference in terms of input data size. Lastly, there is the result from feedback of the blind test for recommended food. The results are as follows:

4.1 Naïve Program

As a control of the experiment, we used the average of all of entries as the prediction of any ratin g P$_{u,f}$. For an average rating of 6 rounded up from 5.798, the RMSE of the naïve program is 3.02 04.

4.2 KNN Algorithm

The KNN algorithm was evaluated by using the approach proposed in Section 3.2.1. The RMSE of the KNN algorithm using three different types of similarity functions with k=13 are provided in Table 1.

Table 1. RMSE of different similarity functions

	Euclidean	Cosine	Pearson
RMSE –Using training dataset	2.46493	2.49006	2.54757
RMSE – Using whole dataset	1.85630	2.28051	1.80219

There is a cross validation between the RMSE and the value of k, specifically ranging from 1 to 100, shown below in Figure 3.

[1] 1The main code is online and shared with github. https://github.com/ramonidea/DietImprovement/blob/master/DietImprovement.ipynb

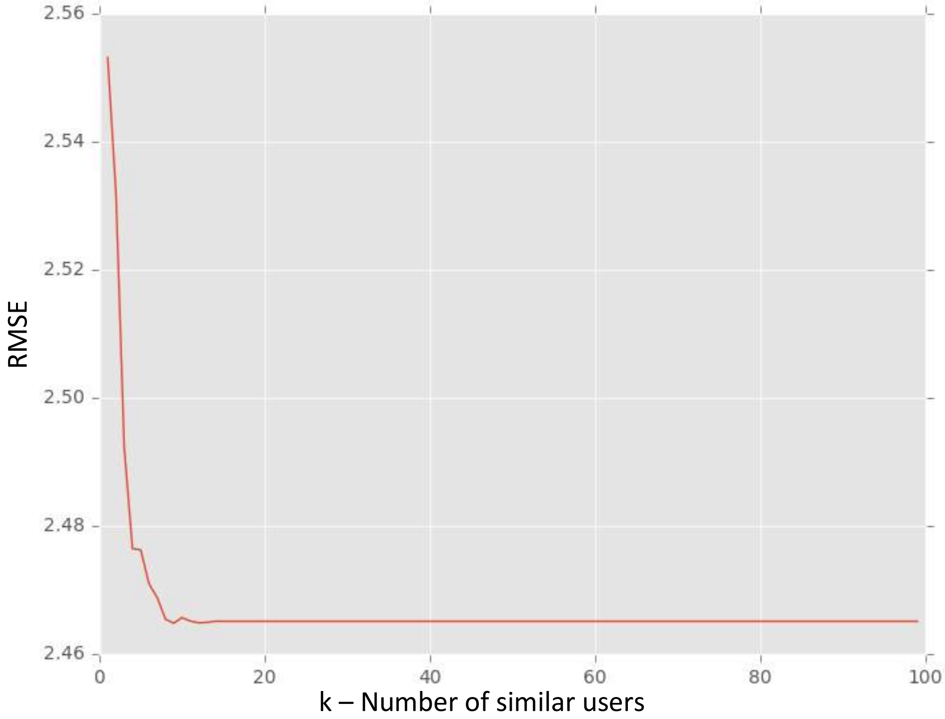


Figure 3 Cross Validation from KNN Algorithm

Therefore, the most accurate result is obtained by applying the Pearson similarity function in the KNN algorithm. These results show that KNN algorithms are more effective than the naïve program. When the value of k is greater than 13, the RMSE value is about the same. Considering about the smaller k chosen, the less execution time it will be. Thus, we chose k = 13 with Pearson similarity function as the best combination in this research. However, the performance does not equal that of other collaborative filtering methods, such as matrix decomposition. Additionally, the KNN method can be used to initialize alternative programs. Practically, in this case, we used the KNN method on the matrix to reduce the scale of the original matrix to increase the time efficiency of LFM.

4.3 Original vs. Modified LFM Algorithm

To compare the performance of the two algorithms, I ran each with identical overfitting parameters, steps=3500, α=0.0003 and β=0.02. The original LFM algorithm uses the whole test set as a matrix input and then requests the value from the result matrix set. Because the matrix is *extremely large and sparse*, the algorithm runs slowly (5:30) and with significant error (RMSE=1.00169). Since the modified LFM algorithm uses just thirteen of the nearest neighbors, the matrix input is *smaller and denser* and contains more useful information, resulting in both faster execution (13.1 seconds) and more accurate predictions (RMSE=0.07334).

The result contains different values for the number of features and the value of the overfitting components required to obtain a more accurate value.

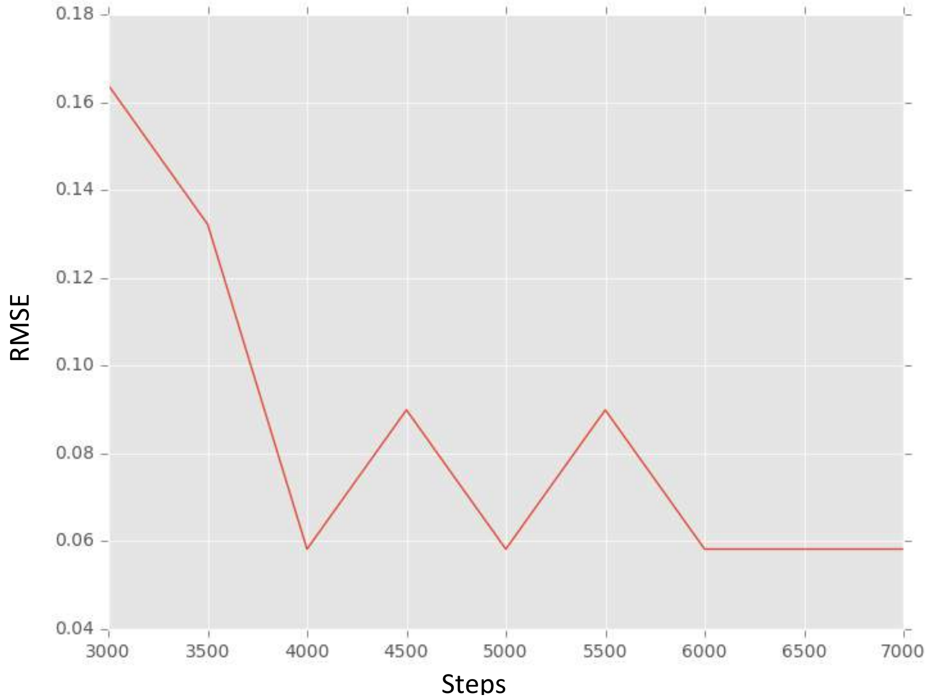


Figure 3 Cross Validation for LFM algorithm: Steps range from 3000 to 7000

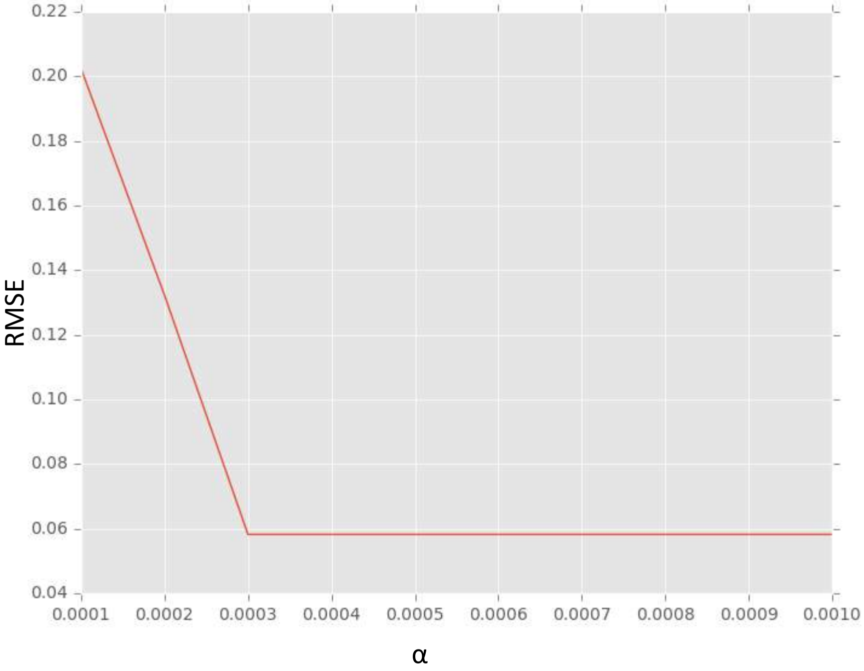


Figure 4 Cross Validation for LFM algorithm: α ranges from 0.0001 to 0.001

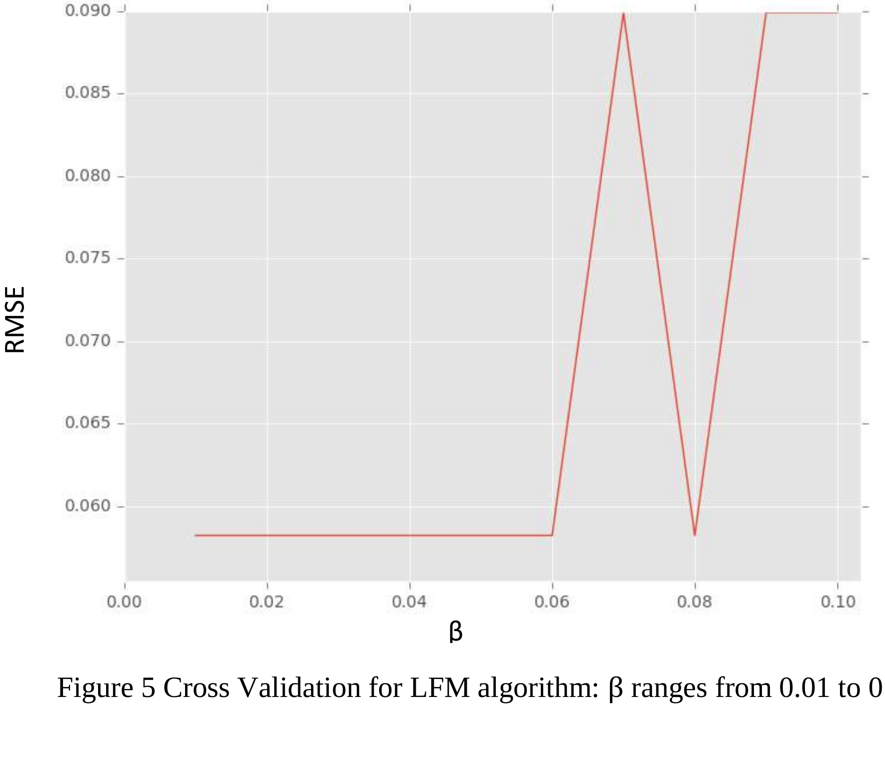


Figure 5 Cross Validation for LFM algorithm: β ranges from 0.01 to 0.1

Therefore, the best result among those illustrated in Figures 3, 4, and 5 is when the number of steps is 4000, α=0.0003, and β=0.02. These optimal components produce an RMSE for the test data of 0.07334.

In summary, the RMSE values representing the best performance of each of the different algorithms are listed in Table 2.

Table 2. Performance of different algorithm in predicting taste rating

Algorithm	Naïve method	KNN method	Original LFM algorithm	Modified LFM algorithm
RMSE	2.9631	2.46511	1.00169	0.07334

4.4 Achieving the Health Goal

The virtual user we created to represent normal health conditions is an 18-year-old male student with a normal level of activity and with an initial height of 180 centimeters and an initial weight of 75 kilograms. The goal that was specified for weight loss was 2 kilograms in 30 days; thus, his BMR was about 1860.55 and the daily calories needed to satisfy the goal were about 2435 calories per day of which about 1000 calories were permitted for lunch. In the list of recommended foods, 86.67% of the food was found to satisfy the user’s needs.

4.5 Real Test Result on Taste Algorithm

A post-survey was conducted to receive feedback on the accuracy of prediction. In the post-survey, the system generated six recommended foods and randomly selected six foods from the food list. This list of 12 foods was scrambled to present the survey as a blind test to make sure people answered this survey without personal influences. Ratings for a total of 376 foods were received. Among these 376 foods, 188 foods were from the recommended food list with the user’s food preferences predicted with an accuracy of 63.3%. The average rating for recommended foods is 7.87234, whereas the average ranting for random foods is 4.49468. This result clearly shows that users consider more foods in the recommended food list to be tasty.

5 Conclusion and Future Work

5.1 Conclusion

This research project aimed to develop a diet improvement system using recommendation system combining foods’ health score and taste score according to customized induced motivation model based on users’ health goal. The later evaluation test results helped to find a most efficient combination of different parameters and proved the validity of the hybrid and novel mixed algorithm used in the system. In addition, the results promote the health of the user; therefore, the diet improvement system based on the recommendation system using the hybrid matrix decomposition algorithm is effective to solve the problem.

5.3 Future work:

· Monitor health goal progress in human participants to validate induced motivation model.

· Try a larger dataset of foods, people, and ratings to test the performance of the system under "big data" conditions.

· Use a neural network and deep learning to improve ratings prediction.

· Study how social relationships between participants affect eating habits.

REFFERENCES

[1] Gunawardana, A., & Shani, G. (2015). Evaluating Recommender Systems. Recommender Systems Handbook, 265-308.

[2] Neumann, S., Thum, A., & Böttcher, C. (2012). Nearline acquisition and processing of liquid chromatography-tandem mass spectrometry data. Metabolomics, 9(S1), 84-91.

[3] Sarwar, B., Karypis, G., Konstan, J., & Reidl, J. (2001). Item-based collaborative filtering recommendation algorithms. Proceedings of the Tenth International Conference on World Wide Web - WWW '01.

[4] Matrix Factorization: A Simple Tutorial and Implementation in Python. (n.d.). Retrieved Jan, 2016, from http://www.quuxlabs.com/blog/2010/09/matrix-factorization-a-simple-tutorial-and-implementation-in-python/

[5] Schelter, S., Boden, C., & Markl, V. (2012). Scalable similarity-based neighborhood methods with MapReduce. Proceedings of the Sixth ACM Conference on Recommender Systems - RecSys '12.

[6] Ryza, S., Laserson, U., Owen, S., & Wills, J. (2015). Advanced analytics with Spark. Sebastopol, CA: O'Reilly Media.

[7] Takács, G., Pilászy, I., Németh, B., & Tikk, D. (2008). Matrix factorization and neighbor based algorithms for the netflix prize problem. Proceedings of the 2008 ACM Conference on Recommender Systems - RecSys '08.

[8] Miller, Neal E. "Motivation and Psychological Stress." The Physiological Mechanisms of Motivation (1982): 409-32. Web.

[9] Stephen, Farenga, J., and Ness Daniel. "Calories, Energy, and the Food You Eat." Questia. Web. Feb. 2016. <https://www.questia.com/article/1G1-143734185/calories-energy-and-the-food-you-eat>.

[10] "Nonessential Nitrogen Supplements And Essential Amino Acid Requirements." Nutrition Reviews 21.6 (2009): 183-84. Web.