EasyChair Preprint

№ 704

A Random Forest Regression-based Personalized Recommendation Method

Yingxue Ma and Mingxin Gan

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

December 27, 2018

A Random Forest Regression-based Personalized Recommendation Method

Yingxue Ma, Mingxin Gan*
University of Science and Technology Beijing, Beijing, China
ganmx@ustb.edu.cn

Abstract

Recently, recommendation methods based on the similarity between different users or objects have achieved remarkable success. However, the high similarity between users does not represent a similar preference in reality. In fact, what really reflects user preferences is user's subjective evaluation on items. In this paper, we propose a method to predict users' evaluations of films by using random forest regression model. As the user's evaluation on items depends on the characteristics of items and preferences reflected in history, we utilize the above two data as inputs to predict users' evaluations of films based on users' rating process simulated by random principle. The results show that the method proposed in this paper outperforms others in MAE.

Keywords: Random forest regression model, user preference, rating prediction
*Corresponding author

Introduction

In recent years, with the rapid development of data on the Internet, we are facing the circumstance of data overload in both e-commerce and daily life. The recommendation system helps people filter out information that users are not interested in, and alleviates the problem of information trek (Buhwan et al, 2010). The recommendation system has showed good performance in many application fields, books (Linden et al, 2003), movies (Nie et al, 2009), news (Wei et al, 2011; Prawesh et al, 2012), TV programs (Barragáns-Martínez et al, 2010), blog articles (Sun et al, 2009), tourism (Gavalas et al, 2014), friends (Backstrom et al, 2011; Jamali and Ester, 2010) and many others.

Collaborative filtering is a basic recommendation method, which suggests what users have been interested in before shows their preferences on new items (Sarwar et al, 2001; Cacheda et al, 2011; Biau et al, 2010; Moreno et al, 2014; Shi et al, 2014) It can be divided into two types: user-based and items-based (Burke, 2002; Sarwar et al, 2001). The calculation of similarity utilizes users' historical rating score, item description, item attribute and other information (Adomavicius and Tuzhilin, 2005). Hybrid recommendation is a combination of these two approaches (Burke, 2002). Some studies have shown that analyzing the potential relationships between users and objects and modeling them can enhance the recommendation effect effectively (Adomavicius and Tuzhilin, 2005; Hofmann, 2004). Another study branch uses networks and diagrams to build connections between users and objects, combining recommendations with network methods. Therefore, in recent years, researchers have being focus on the relation between users and items, utilizing users' social network to build the network between them, and employing the random walk model is to figure out user interest points (Chen et al, 2015; Tian and Jing, 2013; Ying et al, 2014).

However, the existing research methods have widespread limitations: unilateral using of user or item information. For example, the user-based collaborative filtering method relies on users' feature information, focusing on users' characteristics to find similar users, whereas it ignores the relationship between item information during users' selection process and factors affecting selection decisions. In this case, goods recommended to the users depends on the degree of similarity between users and items,

but actually there is a phenomenon in which the similarity of two users is extremely high while they are not similar at all. For example, one user scored x= (1,1,2,1,1) on a batch of movies, and another scored the same movies as y= (5,4,5,5,5). The similarity between the two scoring records is high, but obviously the first score is low and the second one is high. Apparently these two users have different preferences on the same movie; as a result, it is unreasonable to divide the two users into a similar group according to the similarity method. Therefore, it is significant to fully consider users' preferences in the recommendation system. Users usually browse and purchase according to their preferences and economic conditions, and it is important to grasp the different preferences of each user to ensure the accuracy of recommendations. Many studies establish user feature models by using user reviews and ratings, using natural language processing and topic modeling to mine user preference information (Gao et al, 2015; Zhang et al, 2017). User preferences can be excavated from interactions between users and items, and the users' evaluation can be predicted using methods such as clustering and linear regression (Tan et al, 2014). Nowadays, there are many studies which abstract the indicators by user evaluation to indicate their preference on items. Using multiple regression to predict user scores for variables extracted in user reviews, and build similar user groups to predict the ratings of target users by others' ratings. One of the disadvantages of these regression methods is that the users' scoring process for items is directly defined as a linear relationship, but in reality, we know that users' ratings of items are affected by many factors, which means the process cannot be completely quantified by a certain rule but as random.

Based on the above discussion, this paper proposes a random forest method to predict users' scores, based on item and user history characteristics. In the process of feature selection and decision tree construction, the users' scoring process is simulated based on the stochastic principle. According to the principle of minimum variance, the random forest regression method can be used to predict the users' score, and the mean absolute error (MAE) is used as an index for measuring the prediction effect.

Method

The premise of our approach is to make full use of information on both users and items (Figure 1). Many studies use the user history to predict user ratings while ignoring item features. This article attempts to integrate two different information, taking into account the objective characteristics of items and subjective users' ratings. Therefore, the model proposed in this paper predicts users' ratings from two perspectives: user's subjective preferences and objective characteristics of items. We give two clear definition:

Definition 1: user's subjective preference. It is the user's history rating that best reflects the user's preference in selecting items.

Definition 2: objective characteristics of items. It is a relatively stable property that describes some objective features of the item itself, which is fixed and cannot change as the user changes or the user's preference changes.

The CART tree processes these discrete data, establishes multiple decision trees for each user based on the principle of minimum variance loss, and integrates multiple decision trees into the forest; finally, establishes a random forest model and tests for all users. Detailed The process is shown in Figure 2.

Our method includes three steps: First, we quantify and encode the properties of the items which have been scored from a certain user's history and calculate the score of each item which the user is interested in; Second, in the process of constructing the decision tree, we select the features of split nodes and experimental samples at random and use the CART tree to process these discrete data, establish multiple decision trees for each user according to the principle of minimum variance loss, and integrate multiple decision trees into forests; Finally, we build a random forest model and test for all users, in other words, each user's rating is predicted through a complete forest. The detailed process is shown in Figure 2.

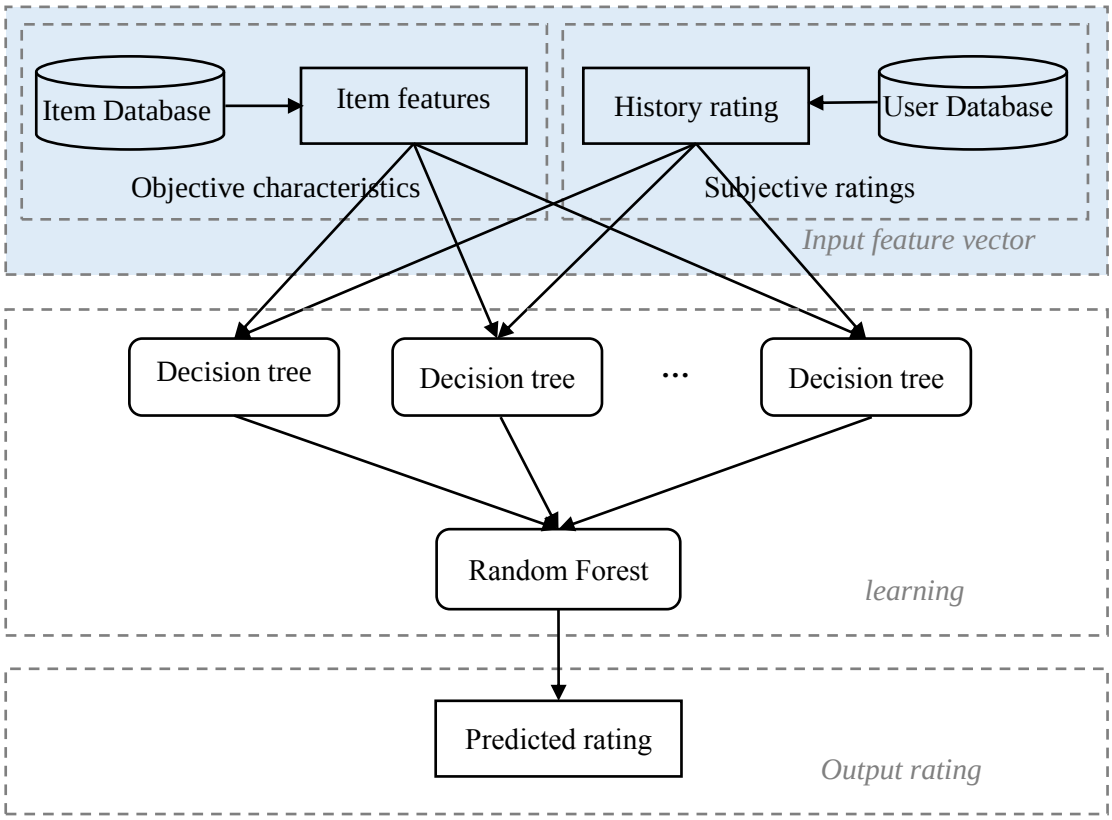

Figure 1: The logical model of the proposed method.

We take the high-dimensional vector that consists of features of items and history rating scores of users as the input. The item features consist of item's attributes which are objective inherence of items, and the rating scores which are subjective impressions given by the user. Therefore, two kinds of features are considered as the input of the method and used to predict the rating to the item by the user.

Algorithm Random forest regression

1: Partition the data into training set and test set
2: **for** *k* **from** *1* **to** *u*
3: **Train$_k$** (a training set for user *k*)
4: **for** *t* **from** *1* **to** *n_tree*
5: Select split nodes randomly from the movie characteristics of training set
6: Construct decision tree according to the minimum error principle
7: **end**
8: **Test$_k$** (a testing set for user *k*)
9: **for** *t* **from** *1* **to** *n_tree*
10: Traverse the decision tree from root node to terminal node according to the movie features in the test set
11: Obtain the predict results from terminal node
12: **end**
13: Calculate the average result and MAE for each user
14: **end**
15: Calculate the mean **MAE** for the algorithm

Figure 2: Rating prediction.

Regression tree construction

The regression tree is constructed by recursive call samples, whose nodes are divided into different levels, from the root node to the terminal node, and the formation of each split node is based on some certain split standards. Once the decision tree is built, a new sample can be predicted along a path from the root node to the terminal node based on the features. To implement the regression, we deal with discrete data and choose the CART decision tree as an independent tree of random forest.

The CART algorithm was proposed by Breiman (1984), when constructing a decision tree, all features are divided into two independent feature subsets and outputs are generated based on a split point, respectively. Considering all the features, traversing each feature under all possible values or split points, the data is divided into two parts, and each part has an output value to calculate the error (Figure 3). Then, we calculate the squared error of the two subsets and select the feature with the minimum error as segmentation point and generate two sub-nodes (Grömping, 2006).

$$\mathrm{R}_1(j,s)=\left\{x \mid x^j \le s\right\}, \quad \mathrm{R}_2(j,s)=\left\{x \mid x^j > s\right\}$$

We hope that some of the output values can minimize this error, and it is obvious that the output value is optimal when the output value is the mean of all the actual values on the corresponding part. We take each property as a variable, and all values of properties are possible for the split variable *j*. After setting a value for the splitting point, we obtain two regions divided by this property. In order to find values of two regions, we minimize the variance *V* in each region. We iterate over all variables and find the optimal split property *j*. In the optimal split property, we find the optimal segmentation value *s*, so as to gain the optimal pair (*j*, *s*) and two regions, which are the two sub-trees divided by the sample, whose values are c_1, c_2.

$$\mathrm{V}=\min\left[\min\sum_{x_j\in R_1(j,s)}(y_i-c_1)^2+\min\sum_{x_j\in R_2(j,s)}(y_i-c_2)^2\right]$$

In general, when the decision tree with better fitting effect, the tree's size will become larger, however, as the composition of random forest, overfitting is avoided to a certain extent, owing to random forests select samples and features based on stochastic principles. Therefore, the pruning problem should not be considered in decision tree construction.

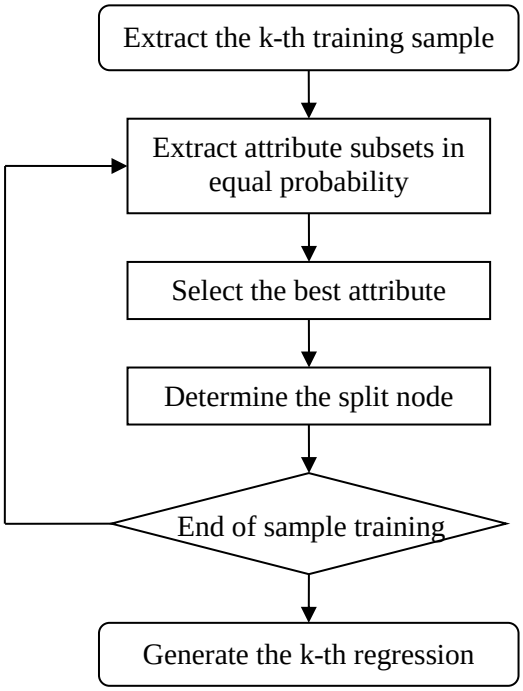

Figure 3: Construction of regression trees.

We believe that the degree of user preference for items is directly affected by the characteristics of the items and is reflected in user's rating records. Formally, for item o, the feature vector is $x_o = (o_1, o_2, o_3, \ldots o_n)$, where o_1 to o_n is n features of item o. For user u we collect all items selected by the user in history and all the items not have selected by the user. After this procedure for a user, we obtain a matrix $X_u = (x_{o1}, x_{o2}, x_{o3}, \ldots x_{oo})$ representing the property settings of all objects that the user might be interested in. For a user, we collect all historical interactive behavior of all items, and obtain a preference vector (rating score) $y_u = (y_{u1}, y_{u2}, y_{u3}, \ldots y_{uo})$, where y_{ui} for $1 \leqslant i \leqslant o$ is the user u preference item i. Repeating this procedure for all users, we obtain a predicted rating preference matrix $Y = (y_1, y_2, y_3, \ldots y_m)$, where m is the total number of users. We fit a regression as $\boldsymbol{Y}$ = Function ($\boldsymbol{X}$) by using these two matrixes**.**

Random forest regression

Multiple decision trees are integrated into a random forest. Random forest models are trained by users' historical data. Users' preferences and influence of scoring between users are learned internally, the entire model is used for user rating prediction. That is, treating each user as an independent sample as input to a random forest. The random forest is composed of a large number of decision trees, and the random forest based on CART tree called RF-CART, which was first proposed by Breiman (1984). It is proved that the number of decision trees in the random forest is the most important parameter affecting the effect of evaluation. The randomness of random forests is reflected in two aspects: the sample selection for each decision tree is random; the variables selected for each feature subset are random. Since the construction of each decision tree is random, the regression results obtained by different trees are different. The result of averaging the regression results of each decision tree is the regression result of random forests. Randomness of random forest avoids the over-fitting of data to a certain extent, so it is not necessary to consider the pruning problem during constructing. The basic idea of the random forest is: First, use Bootstrap principle to extract k samples from the original training set, and the total number of all samples is the same as the original training set; Second, establish k decision tree models for k samples and obtain k prediction results; Finally, the ultima result of each record is determined from the mean of the predictions of the k decision trees (Figure 4).

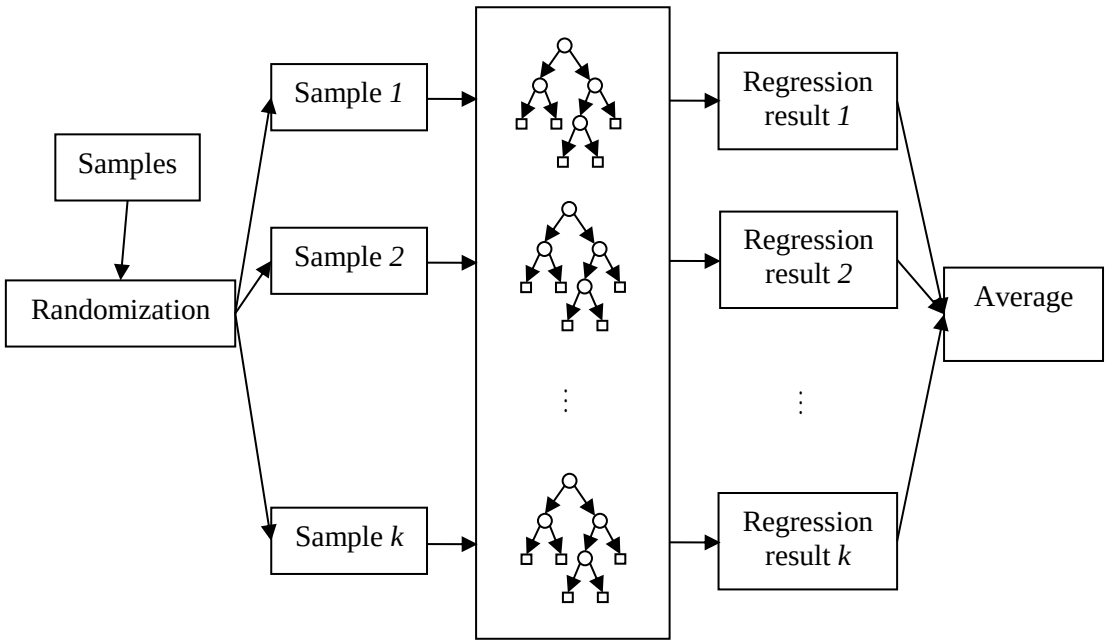

Figure 4: The process of rating prediction.

There are several important parameters in a random forest model. The number of decision trees in the forest has most effect on the performance of a model. In general, the more sub-trees the better performance, but the calculation speed of the model will slow down. In addition, the maximum number

of features in a single decision tree is also one of the important parameters. There are three ways to set the maximum number of features: using all the features to generate a decision tree, without limiting the number of features; selecting the square root of the total number of features as the maximum number of features; choosing 20% of the total number of features as the maximum number of features, this value can be extended to x% based on actual needs. Increasing the maximum number of features can generally improve the performance because it allows more options to be considered at each node, but it reduces the diversity of individual decision trees. Obviously, increasing the maximum number of features will reduce the speed of the algorithm. For some problems, simply increasing the maximum number of features does not improve performances, owing to the optimal parameters require multiple experiments to determine actual problems. The maximum depth of decision tree is usually chosen from the range of 10 to 100. When the amount of data or the number of features is relatively small, the depth of the decision tree cannot be limited, but if the model has a large number of features, the maximum depth needs to be determined based on the distribution of the data.

Method for comparison

We compare the proposed method with three other methods. First, the user's ratings are discrete. We use Random Forest Classification method to predict users' score according to the characteristics of items and users' rating record. When establishing a forest classification decision tree, the segmentation attributes are selected using the principle of maximum entropy. The voting result of each tree is the final result of the random forest classification. Second, we attempt a Collaborative Filtering method to predict users' ratings and divide the data into a training set and a test set. The user similarity is calculated based on the ratings of the target user and all other users. Take the top 10 users with the largest similarity as the target user's neighbors. In the test set, the average score of the 10 neighbor users' ratings on the specified item is taken as the target user's predicted rating on that item. Finally, we use Random Guess method to generate random number in the range of 1 to 5.

Results

Dataset

The data set used in the experiment was MovieLens (100k) collected by the GroupLens team (https://www.grouplens.org). The data set includes ratings of 943 users on 1682 movies, and scores are integers from 1 to 5, with a total of 100, 000 rating records for all users. The data set includes the genre and year information of the movies, and the data is processed to convert the features of the movie into a hot-coded vector containing only 0-1 variables, which are used to represent the characteristics of the movies. The experiment uses 10 cross validation experiments to ensure the generalization ability of the model. We randomly divide 100,000 records of raw data into ten approximately equal-sized subsets. In each validation, we use nine subsets as training data to train the random forest model and remain the rest of the subsets as evaluation test data.

The movie attribute is obtained from the movie title. All the values of the movie in the year attribute are collected. The movie year attribute is expanded to a high-dimensional sparse vector. The total dimension of the vector is the total number of all years. The form of this vector becomes the One-Hot encoding format, and each element of the vector is either 0 or 1. "1" indicates that the movie is changed to the corresponding attribute value on the attribute of the year, and "0" indicates that the attribute value of the movie child year is not changed. The conversion tool we use is the function of OneHotEncoder in the sklearn toolkit in Python. We use the same method to deal with the movie's category, except that the same movie may have a category coverage; in other words, there may be more than one "1" in the category vector of the same movie, not the One-Hot in the strict sense. At this point, the of each movie can be represented by a higher-dimensional vector formed by connecting these two vectors as characteristics of the items. At the same time, we collect the users' ratings on all movies, and use the score of all users in one movie as the item's subjective feature of the movie that is users' preference.

MovieID	1995	1996	…	2001	Adventure	Animation	…	Comedy	Fantasy	U_1	U_2	U_3	…	U_n
46098	0	0	…	1	1	0	…	1	0	2	0	5	…	3

Figure 5: Combination of subjective and objective features.

Evaluation criteria

The evaluation criteria adopted in this article is the mean absolute error (MAE). The MAE refers to the distance between the predicted ratings and the actual users' ratings. It is a commonly used measurement model for predicting. The forecasted quality can be measured intuitively, and the smaller the MAE, the better the prediction performance.

$$\text{MAE}=\frac{\sum_{i=1}^{n}|\,p_i - r_i\,|}{|\text{N}|}$$

Among them, p_i refers to the user's prediction score for item *i*, while r_i refers to user's actual rating for item *i*, and *N* represents the total number of items predicted.

Performance comparison

In this paper, a random forest regression method is proposed to predict users' ratings, each user's rating record and related characteristics is viewed as a sample, each sample from the input to the output in prediction process traverses a forest. The experiment aims to compare the prediction performance of random forest regression with other methods, and evaluate the quality of data processing and regression prediction method. At the same time, the best prediction results for different parameters are compared to determine the optimal parameter combination for the random forest regression method.

We conducted 10-fold cross-validation experiments to evaluate the performance of our method. The mean of all error produced by each method can be compared with the MAE of different prediction methods. The results were obtained by using the same data in random forest classification, regression and so on. For the random forest approach, the regression method is 6.7% lower than the MAE of the classification method. As is shown in Figure 4, the MAE of the random forest regression method can reach 0.782 (variance 0.013), while the random forest classification MAE reaches 0.849, slightly larger than the MAE of the random forest regression method, but obviously more accuracy than Collaborative Filtering (1.236) and Random Guess (1.527).

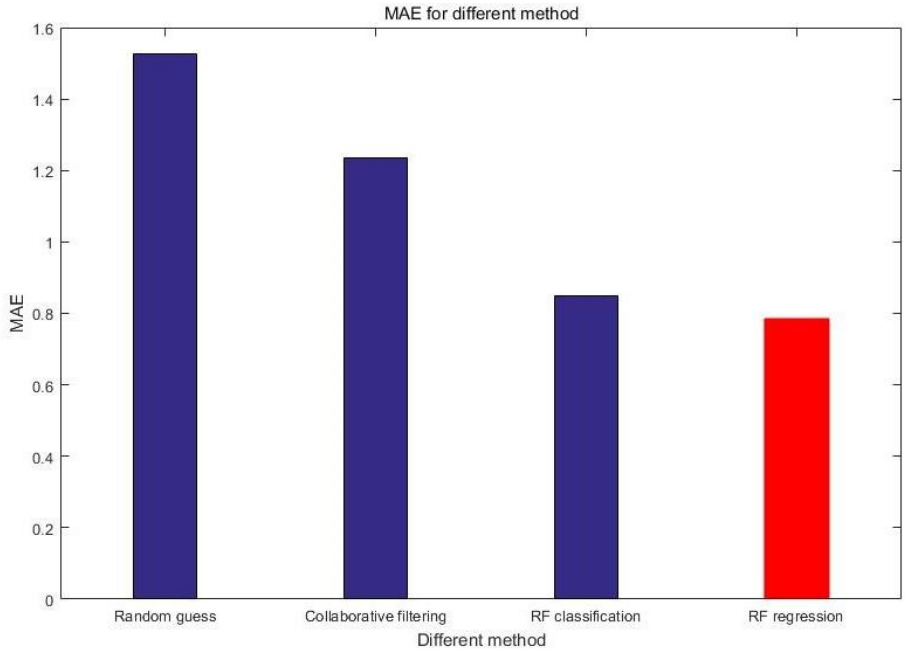

Figure 6: Methods Comparison on MAE.

There are three main parameters in our method: the number of decision tress in the random forest, the maximum number of features and the maximum depth of the decision tree. By default, these parameters

are set to: the number of trees = 10, the maximum number of features = 'Auto' and the maximum depth = None. Therefore, we then conducted an adjustment experiment to observe the effect of these important parameters on the prediction effect of the method. Firstly, we set the number of decision trees in the random forest in the range of 10 to 200, and evaluated the performance of our method in the 10 cross-validation experiment on the Movielens data set. The resulting of the performance on MAE is shown in Figure 5, which suggests that the more the number of decision trees, the smaller the prediction error. However, as the number of decision trees increases, the speed of error gradually decreases. It can be seen that the optimal number of decision trees on this data set is in range of 10 to 30.

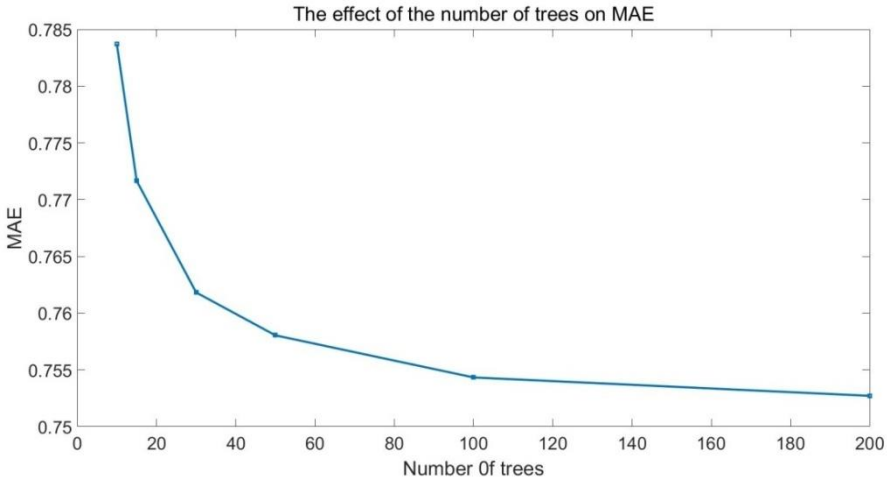

Figure 7: The relationship between the effect of random forest regression and the number of decision trees.

We then studied the influence of the maximum number of features selected when constructing the decision tree on the final prediction effect. We experimented three commonly used setting methods. Select all feature numbers, 20% of the total number of features, and the square number of total roots of all features. The prediction errors we obtained are 0.784, 0.783 and 0.786 respectively. The performance on different numbers of features exhibits stable patterns. These observations suggest that one can simply select the parameter of the maximum features in any way depending on the actual situation. Usually, the maximum depth of a single decision tree is generally specified in the range of 10 to 100. The maximum depth we set in the experiment is 10, 30, 50, 70, 90. The results show that when the maximum depth of the decision tree is 90, the prediction error is minimal, and this conclusion is also consistent with the general rule that the maximum depth affects the effect of the random forest method. Finally, we determine that the "number of decision trees" is 200, and the "best feature" takes 20% of all the features, and "the maximum depth of a single decision tree" takes the best combination of 90 parameters to validate our prediction method, this parameter set can be interpreted as that in the entire experimental data set, when each user's rating record is taken as a sample, use this random forest with above parameters predicts the output value of each sample to reach the current minimum error, and the MAE is reduced by 3.301%, reached 0.749.

Conclusion

In our study, we combine item features and user's preference reflected in history records and develop a random forest regression model to predict users' ratings. Compared to the random forest classification method, our proposed regression method achieved better performance. Then we analyze the important parameters in the regression method, and a set of better parameters selected for prediction. Finally, we achieved the lowest MAE for the proposed method. The significance of this study is to try to use a new data processing method and regression method to predict the user's score with minor error. User preferences can be analyzed on the basis of predicting user ratings or develop better recommendations combined with the traditional recommendation methods. In summary, this article contributes to the following research in related fields:

The data processing and prediction methods proposed in this paper can be widely used on recommendations for different types of items, as long as features of both items and users' preference are available. For example, book recommendations in the library, collecting book features and user borrowing records allows you to predict user borrowings for target books; Such as commodity recommendation in e-commerce, collecting product features and user's browsing, collecting, and buying records can predict the user's behavior toward product.

A random forest approach was applied to predict user's rating and reach smaller error with simple operation. Compared to traditional methods, we do not define the relationship of the regression process, while directly get the mapping result from input to output avoids possible errors in relation description with a simple operation process. Provides new ideas for the solution of problems that do not focus on process description.

However, there is a small limitation in this article, we did not get the user's feature information, such as age, gender and occupation. Subsequent research may be more accurately based on the user's complete information.

References

Adomavicius, G., &Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge & Data Engineering, 17(6), 734-749.

Balabanović, M., &Shoham, Y. (1997). Fab: content-based, collaborative recommendation. Communications of the Acm, 40(3), 66-72.

Barragáns-Martínez, A. B., Costa-Montenegro, E., Burguillo, J. C., Rey-López, M., Mikic-Fonte, F. A., &Peleteiro, A. (2010). 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.

Biau, G., Cadre, B., &Rouviére, L. (2010). Statistical analysis of k-nearest neighbor collaborative recommendation. Annals of Statistics,38(3), 1568-1592.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984), Classification and Regression Trees, Belmont, CA: Wadsworth.

Burke, R. (2002). Hybrid recommender systems: survey and experiments. User Modeling and User-Adapted Interaction, 12(4), 331-370.

Cacheda, F., &Formoso, V. (2011). Comparison of collaborative filtering algorithms:limitations of current techniques and proposals for scalable, high-performance recommender systems. Acm Transactions on the Web, 5(1), 1-33.

Chen, Z., Xia, F., Jiang, H., Liu, H., & Zhang, J. (2015). Aver: random walk based academic venue recommendation. 579-584.

Gao, Y. P., Yu, W. Z., Zhao, P. F., Zheng, Z. L., & Zhang, R. (2015). Analyzing reviews for rating prediction and item recommendation. Journal of east China normal university (natural science edition), 2015(3), 80-90.

Gavalas, D., Konstantopoulos, C., Mastakas, K., &Pantziou, G. (2014). Mobile recommender systems in tourism. Journal of Network & Computer Applications, 39(1), 319-333.

Grömping, U. (2006), "Relative Importance for Linear Regression in R: The Package relaimpo," Journal of Statistical Software, 17, 1. Available at http://www.jstatsoft.org/ v17/ i01/ .

Hofmann, T. (2004). Latent semantic models for collaborative filtering. ACM.

Jeong, B., Lee, J., & Cho, H. (2010). Improving memory-based collaborative filtering via similarity updating and prediction modulation. Information Sciences, 180(5), 602-612.

Kim, H. N., Ha, I., Lee, K. S., Jo, G. S., & El-Saddik, A. (2011). Collaborative user modeling for enhanced content filtering in recommender systems. Decision Support Systems, 51(4), 772-781.

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Computing, 7(1), 76-80.

Moreno, A., Ariza-Porras, C., Lago, P., Jiménez-Guarín, C. L., Castro, H., &Riveill, M. (2014). Hybrid model rating prediction with linked open data for recommender systems. Communications in Computer & Information Science, 475, 193-198.

Nie G., Xia H.,& Li X. (2009). An Ontology-based Approach on Intelligent Recommendation in Movie Field. Proceedings of the 6th International Conference onInnovation and Management, 1489–1494.

Prawesh, S., & Padmanabhan, B. (2012). Probabilistic news recommender systems with feedback. (pp.257-260).

Sarwar, B., Karypis, G., Konstan, J., &Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. International Conference on World Wide Web (Vol.4, pp.285-295). ACM.

Sarwar, B., Karypis, G., Konstan, J., &Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. International Conference on World Wide Web (Vol.4, pp.285-295). ACM.

Shi, Y., Larson, M., &Hanjalic, A. (2014). Collaborative filtering beyond the user-item matrix:a survey of the state of the art and future challenges. Acm Computing Surveys, 47(1), 1-45.

Sun, A. R., Cheng, J., & Zeng, D. D. (2010). A novel recommendation framework for micro-blogging based on information diffusion. Social Science Electronic Publishing, 12(7).

Tan, Sh., Zhang, B. Q., & Li, Z. X., (2014). An Algorithm for Recommendation Based on the Iterative Weighted Regression Method. Mathematical theory and application, (3), 38-47.

Tian, G., & Jing, L. (2013). Recommending scientific articles using bi-relational graph-based iterative RWR. ACM Conference on Recommender Systems (pp.399-402). ACM.

Wei, D., Zhou, T., Cimini, G., & Wu, P. (2011). Effective mechanism for social recommendation of news. Physica A Statistical Mechanics & Its Applications, 390(11), 2117-2126.

Ying, J. C., Kuo, W. N., Tseng, V. S., & Lu, H. C. (2014). Mining user check-in behavior with a random walk for urban point-of-interest recommendations. Acm Transactions on Intelligent Systems & Technology, 5(3), 1-26.

Zhang, H. L., Liu, J. Y., Yang, S. N., & Xu, J.(2017). Research on the evaluation model based on network user comments. Modern book intelligence technology, 1(8), 48-58.