International Journal of u- and e- Service, Science and Technology
Vol.9, No. 2 (2016), pp.101-108
http://dx.doi.org/10.14257/ijunesst.2016.9.2.11

Research on Personalization Algorithm based on Collaborative Filtering

Liu Feng
Heilongjiang University of Technology, Jixi 158100, china,
Liufeng8038@163.com

Abstract

For the issue of single CF algorithm performing low recommendation precision, we propose an adaptive Adaptive-Boost.RT ensemble learning algorithm. First, the base regression predictor is formed by minimizing the error function of user's predicting ratings via gradient descent algorithm. Then, we introduce an adaptive error parameter, which has statistical property and can be adjusted automatically by the predict error, instead of original parameter. Experiments results demonstrate that this ensemble learning algorithm can improve performance of single CF model significantly.

***Keywords**: Recommendation Systems, Collaborative Filtering, Clustering, Ensemble Learning*

1. Introduction

Of all existing model-based collaborative filtering algorithms, most are discussed to generate recommendations with single model. Although some algorithms have made pretty good effects, the recommendation precision is restricted [1]. In addition to the improved accuracy of recommendations, the generalizing ability of such models is also a key challenge to the recommendation system. Integrated learning is one method able to increase effectively model prediction precision and generalization performance. It's one research direction in machine learning field in recent years. It has basic idea like: for primitive problem, it creates a series of models, through combination of those models. It gets better precision and generalization effect than single classifier model.

Research findings reveal that in collaborative filtering, Ensemble learning can obtain better prediction result than single model. In Netflix contest, Bell et al [2] built one Ensemble learning model which has 107 algorithms to increase the forecast result. To fuse better experimental results, they used linear regression, creating model on each sub-model with different parameters. It turned out that Ensemble learning was much better than the improved single model. [3] presented Boosting Ensemble learning method based on Ada-Boost, producing recommendation results in the way of ordering. The algorithm integrates many weak ordering algorithms, getting the final recommendation results by joining results of those weak learning methods. Steffen Pauws et al[4] suggested the method based on decision rules for music recommendations; it proved experimentally superior over merely content-based collaborative filtering method. Michael Jahrer et al[5] developed one heterogeneous Ensemble learning method which contains19 models such as SVD method, neighboring approach, restricted Boltzmann machine and asymmetry factor model. They were combined through residual training. The method realized higher prediction precision than any one of 19 sub models according to experimental results.

Here we propose one heterogeneous Ensemble learning algorithm based on the improved Adaptive-Boost.RT. The prediction algorithm is simple but effective regressive method. The regression formula is obtained with gradient descent method to minimize user rating error function. Adaptive-Boost algorithm assigns the same weight to each training sample in the initial period; then after training the regression model is got. In the

ISSN: 2005-4246 IJUNESST

next iteration, based on model's prediction errors of training samples, the weight of each sample is updated, increasing the weight of samples which have bigger training errors, so as to assure that the regression model will make possibly sample prediction correct in the next training. In the process, the weight in each iteration needs normalization, making total weight equal to 1. At last, Ensemble learning machine is acquired by weighing all regression models. Individual model which has lower prediction errors takes heavier weight in the integration process. Meanwhile, in the learning course, we use a deviation factor α which has statistical features to replace relative error parameter φ in the original algorithm. The factor α can self-adjust according to different prediction errors, to make the modification of sample weight in more conformity to prediction result. It demonstrated that the proposed algorithm can effectively improve the recommendation precision of individual model.

2. Ensemble Learning Summary

2.1 Basic Conception

Ensemble learning is a new research filed in machine learning. It dated the earliest back to researches by Hansen and Salamon [6]. Integrated learning solves primitive problems by training multiple learning machines. That's why it can effectively increase the generalized learning capability of the system. In view of huge potential and application prospects, Ensemble learning has been listed by expert T. G. Dietterich in machine learning field das one of the four research directions in the current machine learning; and it has received wide applications in fields like speech recognition, text mining and image processing.

Ensemble learning, in conceptual sense, has narrow and broad type. The narrow Ensemble learning involves individual learning machines which are of the same kind, i.e. homogeneous learning machine, which uses BP neural network or linear regression model. The only difference is those individual learning machines have different parameters. On the early stage of Ensemble learning research, the narrow Ensemble learning was often seen. Sollich and Krogh [7]used the narrow learning for neural network integration in 1996. The classic bagging and boosting algorithms were introduced on it as well. Regarding the broad Ensemble learning, as long as several learning machines are utilized to solve problems, it's Ensemble learning [8]. Opitz and Maclin [9] firstly defined the broad Ensemble learning in 1999. Since individual learning machine uses models of different kinds, the broad learning is advantageous over the narrow and accepted by more and more researchers. For learning machine, it has classifying problem and regression problem. So the research on Ensemble learning includes classifying and regression. Here we talk about regression problem. The learning machine is regression model. Figure 1 is a schematic diagram of ensemble learning.

2.2 Individual Generation Method

The differences among individual learning machines affect directly the effect of the final integration method. The main objective of individual generation method is how to generate the machine with bigger differentiation. In Ensemble learning, the method to generate individual learning machine includes the one based on changing sample distribution, the one based on feature selection, the one based on cross validation and the one based on random disturbance.

Hereunder we introduce the method based on changing sample distribution.

It is recalled one based on data partitioning. It follows the principle: produce datasets which are largely different in distribution by varying the distribution of samples, to utilize these training sets for modeling to get greatly different individual learning machines. Methods for changing sample distribution include Bagging [10]and Boosting [11] method. They are two most popular approaches for generating individuals. In the following we'll introduce them in details.

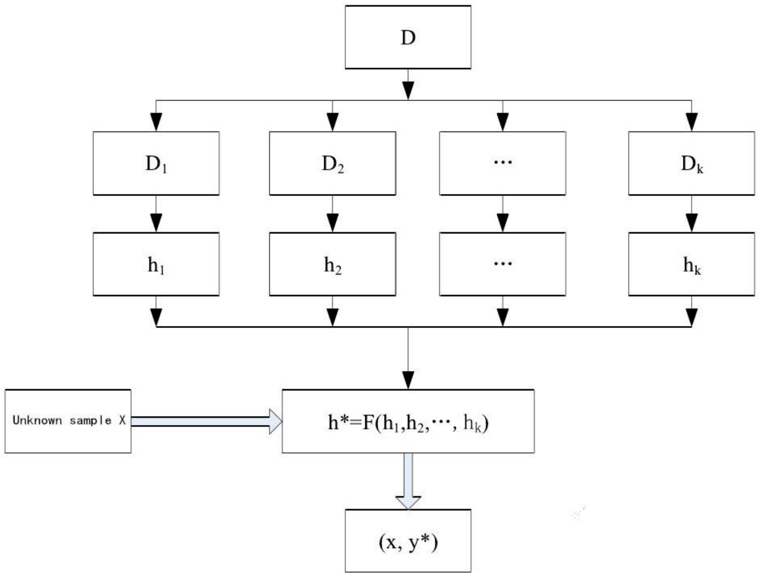

Figure 1. Ensemble Learning Diagram

2.2.1 Bagging algorithm. Bagging algorithm was proposed by Breiman. It adopted one Bootstrap method to generate samples, i.e. using random sampling method to extract samples from original data to produce one new dataset which has the exactly same size with the original one. Such methods refer to sampling without repeating, sampling with replication and their combined method. In Bagging algorithm, it uses sampling with repeating, because in acquired datasets after sampling, the same sample may appear many times; after that, create model for each training subset after sampling to get respective learning machine; finally integrate a few learning machines to generate one integrated composite learning machine M; before the arrival of unknown sample X, use M to predict it.

Bagging algorithm is as follows:

Bagging algorithm[12]

Input: Training set D, number K of individual classifier, a learning scheme (such as decision tree, neural network, etc.)

Output: Complex classifier C^*

Steps:

1 for i=1 to k

2 Through the D's back sampling to get Bootstrap sample set D_i .

3 Using D_i derived classifier C^*

4 end for

2.2.2 Boosting Algorithm. The other very popular individual generative method for changing sample distribution is Boosting, which was firstly proposed by Schapir and then improved by Freund [13]. Different from bootstrap of Bagging, where each individual learning machine's sample generation is mutually independent and establishing every such machine can be a concurrent process; while Boosting algorithm chooses samples in a specific order; the distribution of training samples in next round is affected by the training result in the last round; generally falsely predicted samples will get higher weight as to make it more likely that they will appear in the next training data.

Adaptive boost algorithm flow is as follows:

The currently hot Boosting algorithm is Adaptive-Boost, which is firstly introduced by Freund and Schapire [14]. Take classifying problem for instance. Adaptive-Boost algorithm maintains a weight distribution of training sample sets, whose initial weights are the same; after training, a classifier is got; then update sample weights as per the correct or false classification of training samples by the classifier; lower the weight of correctly categorized instances and hoist the weight of falsely classified ones so that in next time, dataset will focus learning on examples which were previously hardly classified; lastly, the integration classifier is acquired through weighted voting of classifier collection; individual learning machines with lower false prediction rate will gain higher weight in the final voting [15].

Adaptive boost algorithm

Input: Training set D, number K of individual classifier, a learning scheme (such as decision tree, neural network, etc.)

Output: An integrated model

Steps:

1 The weight of each sample in the D is initialized to 1/d, and d is the number of samples in D.
2 for i=1 to k, do
3 According to the weight of the sample from the D, to get D_i ;
4 Using data set D_i to train the model M_i ;
5 Calculate the error rate of M_i error (M_i)
6 if error(M_i)>0.5 then
7 The weight is re-initialized to 1/d;
8 Turn to step 3) and try again;
9 end if
10 for each correct classification of the sample in D_i
11 Weight of sample* $error(M_i)/(1-error(M_i))$
12 Normalized weight of each sample
13 end for

3. An Improved Adaptive-Boost Collaborative Filtering Algorithm

3.1. Design of base Class Learning Algorithm

When the Ensemble learning is used for modeling, firstly it's necessary to determine base class learning algorithm. Here we use a simple but effective regression algorithm, who is devised as follows:

In the beginning we need to define in one recommendation system, user set U and item set I; $r_{u,i}$ stands for user u's rating of item i. Suppose we can use a triple $(u,i,r_{u,i}), k=1,2,...,m$ to represent m ratings generated in the system. Define |U| as user quantity and |I| as item quantity. In real cases, since the value of |I| is quite huge, a user can only evaluate partial items, causing very sparse data, i.e. $m >> |U| \times |I|$. Normally, it's very difficult to find a user who rates all items. Actually effective recommendations are hardly generated because many users rate too fewer items. We predict that the algorithm will eventually produce one rating collection $(u,i,r_{u,i}), k=1,2,.....,|U| \times |I|$ as follows. In the proposed Ensemble learning regression algorithm here, missing rating data are obtained after continuous minimization of error function in the iteration process.

Definition of $\hat{r}_{ui}$ R represents the user u to project i's predictive score, It is shown in expression (1).

$$\hat{r}_{ui} = b_{ui} + \sum_{\substack{v \in N(u) \\ i \in R(v)}} w_{uv}(r_{vi} - b_{vi}) \tag{1}$$

Where, $u = 1,...,U; i = 1,..., I$, w_{uv} represents the impact weight of the user's V rating on the user's u. b_{ui} represents the initial predictive score for the project i on user U. N (u) is a collection of user u nearest neighbor. R (v) is a collection of items evaluated by the user v.

N (u) is a nearest neighbor user set by scoring similarity measure. Here, in order to verify the effectiveness of the ensemble learning algorithm. In this paper, the most commonly used Pearson in collaborative filtering is to measure the similarity of users, and select the highest similarity among the former N users as the user U's nearest neighbor set.

In formula 1, first initialized the user u to the project i score b_{ui} , such as formula 2:

$$b_{ui} = \mu + b_u + b_i \tag{2}$$

Where, μ is the average value of all the users for all items in score matrix, the non null score, b_u is the average score for the user u, b_i is the average score for the project i.

Defining indicator function:

$$1_{ui} = \begin{cases} 1 \ \ if \ \ r_{u,i} \ exist \\ 0 \ \ \ Otherwise \end{cases}$$

The score values of μ , b_u and b_i can be calculated by formula 3 and 5:

$$\mu = \frac{\sum_u \sum_i r_{ui} \times 1_{ui}}{\sum_u \sum_i 1_{ui}}, u = 1,...,U; i = 1,..., I \tag{3}$$

$$b_u = \frac{\sum_i r_{ui} \times 1_{ui}}{\sum_i 1_{ui}} - \mu, u = 1,...,U \tag{4}$$

$$b_i = \frac{\sum_u (r_{ui} - (b_u + \mu)) \times 1_{ui}}{\sum_u 1_{ui}} - \mu, i = 1,..., I \tag{5}$$

The total error of the prediction score can be expressed as:

$$E(W) = \sum_u \sum_{i \in R(u)} \gamma_{ui} (\hat{r}_{ui} - r_{ui})^2 \tag{6}$$

3.2. Design of the Improved Adaptive-Boost. RT Algorithm

After base class learning algorithm is confirmed, we can use Adaptive-Boost Ensemble learning method for modeling. Adaptive-Boost method assigns one weight distribution to training sample sets, whose initial weights are generally identical; after training, a classifier is got; next update sample weights as per the correct or false classification of training samples by the classifier to increase the weight of falsely classified samples as to categorize them correctly in later training of classifying models; at last, the integration classifier is acquired through weighted voting of classifier collection; individual learning machines with lower false prediction rate will gain higher weight in the final voting. When learning task is a regression problem, the classifier is changed to relative regression model and sample classifying accuracy ratio to regression prediction error for judgment.

Adaptive-Boost.RT is an Ensemble learning algorithm used in the regression model [16], one variant of Adaptive-Boost algorithm. Compared with traditional Adaptive-Boost algorithm, Adaptive-Boost.RT has these merits:

(1) it introduces error threshold φ, based on which we can sort training data samples to good and bad type and assign to them different weights;

(2) the update of weighting parameters is unlike Adaptive-Boost.R2 [17]; when the prediction error is lower, it will focus more on learning difficultly predictable samples;

(3) the final output result of Adaptive-Boost.RT is the weighted mean value of all weak learning machines' outputs; while others' is largely output mean value of weak learning machines; and that's contributive to better performance of the algorithm.

Adaptive-Boost.RT algorithm involves two parameters in the learning process: iteration and error threshold φ. Only when a sample' relative error is above φ, its weight will be added. In the improved Adaptive-Boost.RT proposed here, we use deviation coefficient α to substitute the relative error threshold φ in the original algorithm. In each iteration of Ensemble learning, we calculate the mean value of predicted error and standard error. Only when sample's predicted error subtraction mean is above α times the standard error, we raise the sample's weight so as to make intensive training of it. Compared with the original method, improving the parameter will bring two good qualities:

(1) unlike φ, α adjusts its value as per predicted error to make such adjustment of sample weight in better agreement with prediction result;

(2) when computing α, we utilize the statistical feature of prediction error, instead of making it beforehand a fixed threshold.

Improved Adaptive-Boost. RT ensemble learning algorithm

Input:

(1) M user - item score triad $(u, i, r_{u,i})_k, k = 1, 2, ..., m$

(2) By design of base class learning algorithm to generate user rating forecast

(3) Ensemble learning iteration number T

(4) Error factor α

Initialization:

(1) set the number of iterations for t =1

(2) the same weight is paid by the uniform distribution.

Iteration: when t < T

(1) Use expression(1) to predict the user's score on the project $\{\hat{r}_{ui}^t\}$

(2) Computational prediction error $\{\hat{r}_{ui}^t - r_{ui}^t\}$ and Standard deviation σ^t

(3) Calculation error rate:

$$v^t = (\sum_{(u,i)\in Q} \gamma_{ui}^t)^2, Q = \{(u,i) : |\hat{r}_{ui}^t - r_{ui}^t| - \varepsilon^t > \alpha.\sigma^t\}$$

(4)Update the weight of each score:

$$\gamma_k^{t+1} = \frac{\gamma_k^t}{N_t} \times \begin{cases} v^t & if \quad k \notin Q \\ 1 & Otherwise \end{cases}$$

(5) Set t=t+1

Output:

Weighted all of the model predictions, the final score was obtained:

$$\hat{r}_k = \frac{\sum_t r_k^t.\log(1/v^t)}{\sum_t \log(1/v^t)}$$

4. Experiment Design and Analyst

4.1. Experimental Data

To validate the performance of the algorithm, as usual we choose MovieLens dataset as testing data. We chose randomly 100 users' ratings to evaluate the model performance from rating dataset about 3900 movies by 6040 users. Of that, 70% ratings are used for training and the rest 30% for testing. For example a user rated 20 movies, of which 14 movies used to test the model and the rest 6 movies used to prove performance of the model. Although merely 100 users are chosen for modeling data, their adjacent users are searched out from the whole user dataset.

4.2Experiment Design and Discussion

Set the size of one adjacent user set N=40; iteration of Ensemble learning is 10. The model performance indicator is mean absolute error (MAE). Fig. 1 shows curve variations of three models' performance in each iteration: single collaborative filtering model, standard Adaptive-Boost.RT Ensemble learning model and the improved Adaptive-Boost.RT.

From the figure 2, we find after with the Ensemble learning method, those models perform better than before in each iteration; moreover, regarding the standard Adaptive-Boost.RT method, the advanced Adaptive-Boost.RT improves much more, whose MAE decreases from 1.105 to 0.948, with performance hoisted by 14.28%, after first-round iteration; after ten iterations, its MAE declines further to 0.745, with overall performance going better by 32.58%. According to result comparison between Ensemble learning and the single learning model, we notice that the proposed strategy here reduces clearly the prediction error of single learning model.

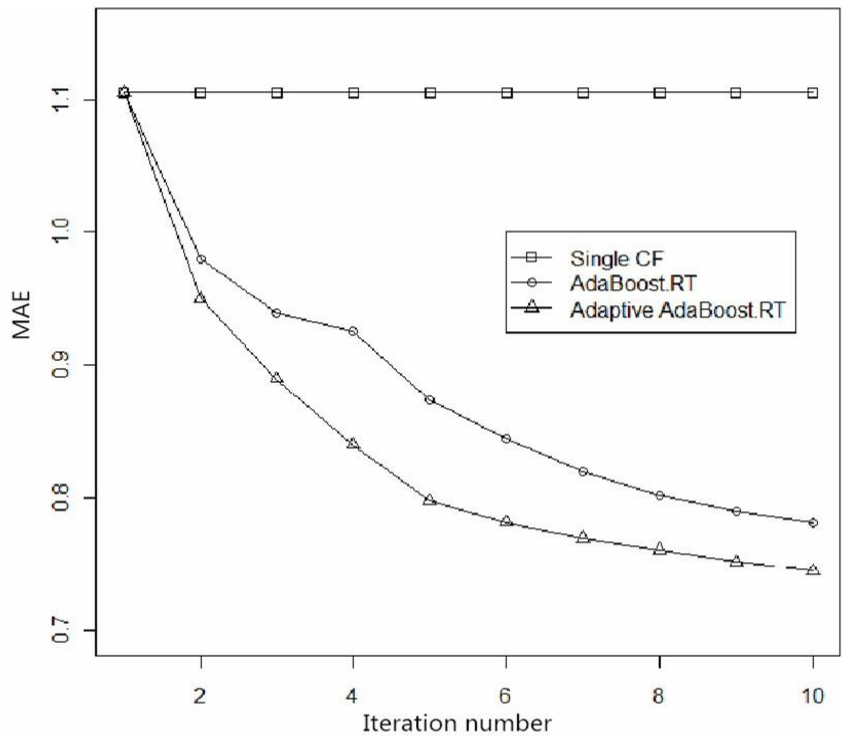

Figure 2. Performance Comparison of Ensemble Learning Algorithm and Single Model

5. Conclusion

In order to address the problem of lower recommendation accuracy by the single learning model, we introduced an improved Adaptive-Boost.RT Ensemble learning algorithm. Firstly we minimized the prediction error by means of gradient descent method and got the regression formula by solving a system of equations; next in the algorithm, we used the deviation coefficient α with statistical property to take place of relative error parameter φ in the original algorithm; α would make self-adjustments as per prediction error to let adjusted sample weight consistent with prediction result. Through experiments, the Ensemble learning algorithm proved its ability to improve apparently the recommendation precision of the single model. Besides, with the use of Ensemble learning, the time grew for building the model. However, in consideration of the actual recommendation system, the model-based collaborative filtering can be implemented off-line. Hence the cost of time consuming may be acceptable.

References

[1] H. Ma, G. Zhang and P. Li, "Overview of collaborative filtering recommendation algorithm", Miniature computer system, vol. 30, no. 7, **(2009)**, pp. 56-65
[2] Y. Koren R. M. Bell and C. Volinsky, "The bellkor solution to the netflix prize", **(2007)**.
[3] Y. Freund, R. Iyer, R. E. Schapire and Y. Singer, "An efficient boosting algorithm for combining preferences", Journal of Machine Learning Research, vol. 4, **(2003)**, pp. 933–969,
[4] M. Tiemann and S. Pauws, "Towards ensemble learning for hybrid music recommendation, Proceedings of RecSys., **(2007)**.
[5] M. Jahrer and A. Töscher, "Combining Predictions for Accurate Recommender systems", Proceeding of 16th Intenational Conference on Knowledge Discovery and Data Mining, **(2010)**.
[6] T. G. Dietterich, "Machine learning research: four current directions", AI Magazine, vol. 18, no. 4, **(1997)**, pp. 97-136.
[7] P. Sollich and A. Krogh, "Learning with ensembles: how over-fitting can be useful", Advances in Neural Information Processing Systems 8, Cambridge, MA: MIT press, **(1996)**, pp. 190-196.
[8] Z. H. Zhou, J. Wu and W. Tang, "Ensembling neural networks: many could be better than all", Artificial Intelligence, vol. 137, no. 1-2, **(2002)**, pp. 239-263.
[9] D. Opitz and R. Maclin, "Popular ensembles methods: an empirical study", Journal of Artificial Intelligence Research, vol. 11, **(1999)**, pp. 169-198.
[10] L. Breiman, "Bagging Predictors", Machine Learning, vol. 24, no. 2, **(1996)**, pp. 123-140
[11] R. E. Schapire, "The Strength of Weak Learn Ability", Machine Learning, vol. 5, no. 2, **(1990)**, pp. 197-227
[12] W. H. Micheline and K. Jia, "Data mining: concepts and techniques", Ming Fan, Meng Xiaofeng. Beijing: Mechanical Industry Press, **(2007)**, pp. 238-262
[13] Y. Freund, "Boosting a weak algorithm by majority", Information and Computation, vol. 121, no. 2, **(1995)**, pp. 256-285
[14] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an Application to boosting", Journal of Computer and System Sciences, vol. 55, no. 1, **(1997)**, pp. 119-139
[15] J. Baining, "Feature selection algorithm research in machine learning", Ocean University of China, **(2009)**.
[16] D. L. Shrestha and D. P. Solomatine, "AdaBoost.RT: A Boosting Algorithm for Regression Problems", Proceeding of the International Joint Conference on Neural Network, Budapest, **(2004)**.
[17] H. Drucker, "Imp roving regressors using boosting techniques", Proceedings of the Fourteenth International Conference onMachine Learning, (**1997)**; San Francisco.

Author

Liu Feng, He received his B.S degree in computer Science from Heilongjiang University of Science and Technology. He received his M.S degree in computer Science from Liaoning Technical University. He is an Associate professor in Heilongjiang University of Technology. His research interests include database and algorithm.

