Proceedings of International Joint Conference on Neural Networks, Dallas, Texas, USA, August 4-9, 2013

Extreme Learning Machine Combining Matrix Factorization for Collaborative Filtering

Tianfeng Shang, Qing He, Fuzhen Zhuang and Zhongzhi Shi

Abstract— **Collaborative Filtering (CF) is one of the most popular techniques for information filtering in recommendation systems. Currently, there are many linear and nonlinear regression algorithms for CF. However, to our knowledge, these regression algorithms may not give satisfactory results in some practical applications. In this paper, Extreme Learning Machine (ELM), which is famous with its fast speed and good performance in generalization, is firstly employed to build a nonlinear regression model for CF, namely ELM for CF (ELMCF) algorithm. Then by combining ELM and Weighted Nonnegative Matrix Tri-Factorization (WNMTF), which can alleviate the data sparsity problem of the user-item matrix, a new nonlinear regression model is proposed, namely Extreme Learning Machine Combining Matrix Factorization for Collaborative Filtering (CELMCF) algorithm, to construct regression based CF algorithms and improve the performance of recommendation systems. Experiments are conducted on several benchmark datasets from different application domains. Experimental results show that the proposed CELMCF algorithm outperforms some state-of-the-art regression based CF algorithms (including ELMCF algorithm, Linear Regression for CF (LRCF) algorithm and Memory based CF (MemCF) algorithm) more efficiently with the competitive effectiveness.**

I. INTRODUCTION

Collaborative Filtering (CF) is one of the most important information filtering approaches in recommendation systems. From the technical point of view, CF recommendation systems mainly include three categories: memory based CF recommendation systems, model based CF recommendation systems and hybrid CF recommendation systems. Memory based CF recommendation systems firstly compute similarity between users or items on the user-item rating matrix, and then generate predictions or recommendations for objective users. Model based CF recommendation systems establish user behavior model over users' behaviors in the past, and predict objective users' behaviors in the future. Hybrid CF recommendation systems combine CF techniques with other recommendation techniques to analyze users' behaviors.

As an important branch of model based CF recommendation systems, regression based recommendation algorithms [21] have obtained more and more attention. Many of them are widely used and made great success in CF recommendation systems, such as linear regression [9], polynomial regression [24], additive regression [4], ordinal regression [2], logistic regression [12], pairwise preference regression [13], temporal regression [1], et al. These "one-to-one" regression based recommendation algorithms are simple and effective.

However, according to our research, there are several problems on the "one-to-one" item regression approach that limit the performance improvement of regression algorithms in recommendation systems. The first one is the loss of considerable nonlinear information. The approximated linear or simple nonlinear functions will inevitably lead to the loss of some nonlinear information of the original recommendation problem. The second one is the inappropriate ensemble weights. In "one-to-one" item regression approach, there are many different calculation methods of ensemble weights. And it is a troublesome question on how to choose the optimal ensemble method, because inappropriate weights will lead to bad recommendation performance. The third one is too many regressors in the "one-to-one" item regression approach. Given I items, the number of regressors generated for prediction and recommendation is $O(I^2)$ which is a very high complexity, especially for large datasets in practical applications.

In this paper, to solve these problems, a new regression model is proposed in the nonlinear "all-to-one" approach. Similar to $f_{I_a,i}$, it builds a multiple-input single-output nonlinear regression model based on Extreme Learning Machine (ELM) and the trained regressors can give recommendation directly. To alleviate data sparsity problem and obtain better performance, Weighted Nonnegative Matrix Tri-Factorization (WNMTF) is employed to fill the sparse user-item matrix. Thus a new nonlinear regression model is proposed based on WNMTF and ELM to improve the performance of CF recommendation systems.

The rest of the paper is organized as follows. Section 2 summarizes some related work on regression algorithms in recommendation systems. Section 3 introduces preliminary knowledge, including basic techniques of CF and some important theories of ELM. Based on WNMTF and ELM, a new regression approach for collaborative filtering is proposed in Section 4. Experiments on four benchmark datasets are showed in Section 5, and experimental results show that the proposed approach can get better performance than some state-of-the-art regression approaches. Section 6 presents conclusions and future work.

Tianfeng Shang, Qing He, Fuzhen Zhuang and Zhongzhi Shi are with The Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China. Tianfeng Shang is also with University of Chinese Academy of Sciences, Beijing, 100049, China. (email: {shangtf, heq, zhuangfz, shizz}@ics.ict.ac.cn).

This work is supported by the National Natural Science Foundation of China (No.61175052, 61203297, 60933004, 61035003), National High-tech R&D Program of China (863 Program) (No.2013AA01A606, 2012AA011003) and National Program on Key Basic Research Project (973 Program) (No.2013CB329502).

II. Related Work

At present, CF based recommendation systems are very popular, and researchers have paid more and more attentions to them. As basic technologies of data mining, regression algorithms are widely used in CF based recommendation systems and they can achieve good performances.

Linear Regression: Kunegis et al. [9] improve the standard memory-based collaborative filtering rating prediction algorithm using the Pearson correlation by adapting user ratings using linear regression. Montanés et al. [12] take the recommendation problem as a time series mining problem, and propose an approach to collaborative tag recommendation based on a logistic regression learning process. Purushotham et al. [14] propose a recommendation method based on social matrix factorization (SMF) and latent Dirichlet allocation (LDA), where SMF is to analyze the structure of social network and LDA can be seen as a collaborative topic regression model to process the item information.

Nonlinear Regression: Zhu et al. [24] propose a polynomial regression based recommendation model in order to alleviate the poor predictions generated by directly using the neighbors' rating. In their model, training sample set is firstly generated by sampling technique, and then it builds the polynomial regression models between items, finally these polynomial regression models are integrated according to appropriate weights: similarity or model error. Frank et al. [4] present two regression schemes, both based on forward stagewise additive modeling (FSAM) [5]. The first method uses FSAM in conjunction with an ensemble of simple linear regression models. In the second method, FSAM is used in conjunction with the k-means clustering method. Mild et al.[11] firstly build various regression models (including linear regression, ridge regression and logistic regression) between items, and propose a new weight calculation method for model selection. Park et al. [13] firstly propose the objective function, so called personalized pairwise loss, and build predictive feature-based regression models that leverage all available information of users and items, such as user demographic information and item content features, to tackle cold-start problems. Yu et al. [22] show that there are some common problems in recommendation systems: one is that the typical collaborative algorithm loses some important parameter when it predicts the ratings, because there might be a strong similarity between the users who give very different ratings. Another is that the classification information of resources is not used. To solve these problems, they have proposed a recommendation algorithm combining the user-based classified regression and the item-based filtering. Vucetic et al. [21] propose a regression-based approach to collaborative filtering on numerical ratings data that searches for similarities between items, builds a collection of experts in the form of simple linear models, and combines them efficiently to provide preference predictions for an active user. It can be said that regression algorithms have achieved great success in recommendation systems.

Sequence Regression: Chang et al. [2] take ordinal ratings into account, and employ singular value decomposition (SVD) [23] and support vector ordinal regression (SVOR) [16] to build prediction classifiers with data of ordinal scale. Brenner et al. [1] use a temporal regression technique to model the short-term evolution of the probability that an item is rated before each test period (in a user-independent way), and then forecast these probabilities on the test week.

These regression based collaborative filtering algorithms have made great success in many applications. However, these "one-to-one" approach based regression recommendation algorithms have some disadvantages, such as generating too many regressors, intractable ensemble weights, et al. In this paper, an "all-to-one" approach based regression recommendation algorithm by combining extreme learning machine and matrix factorization techniques are proposed to improve the performance of recommendation systems.

III. Preliminary Knowledge

In this section, we mainly give some brief introduction on two basic techniques: Collaborative Filtering (CF) and Extreme Learning Machine (ELM).

A. Collaborative Filtering

Recently, CF based recommendation systems have gradually become the most important recommendation systems. Nowadays, they are widely used and make great success in many applications, especially in e-commerce web sites, such as $www.amazon.com$, $www.netflix.com$, $www.ebay.com$, et al. Besides, lots of researchers pay more and more attentions to CF techniques of recommendation systems. Generally speaking, CF techniques mainly include three categories [20]: memory based algorithms, model based algorithms, and hybrid CF algorithms. Memory based CF algorithms [10] firstly compute similarity between users or items on the user-item rating matrix, and then generate predictions or recommendations for objective users. Model based CF algorithms [21] establish user behavior model over users' behaviors in the past, and predict objective users' behaviors in the future. Hybrid CF algorithms [19] combine CF techniques with other recommendation techniques to analysis users' behaviors.

Early recommendation systems are generally rule-based systems, and then some content-based and CF recommendation systems are developed gradually. One of the first CF recommendation systems is Tapestry [6], whose developers coined the phrase "Collaborative Filtering (CF)". CF recommendation systems make information filtering more efficient and more accurate, and people can get more useful information from huge amounts of data in much shorter time.

In CF recommendation systems, the fundamental assumption is that if users u and v have similar ratings on some items, they will also give similar ratings on other items. Based on the above assumption, the basic form of CF recommendation systems can be described as follows. In a typical CF recommendation system, there are two lists: one is list of U users, $\{u_1, u_2, ..., u_U\}$, and the other is list of I

items, $\{i_1, i_2, ..., i_I\}$. The users' ratings on items can either be explicit indications, such as real numbers or like/dislikes, or implicit indications, such as purchases or click-throughs. All the ratings constitute a user-item rating matrix, usually a very sparse matrix.

For example, Table I is an example of a user-item matrix, in which each element denotes one user rating on one item.

TABLE I
AN EXAMPLE OF A USER-ITEM MATRIX

	i_1	i_2	i_3	i_4	i_5
u_1	1	5		3	4
u_2	4		4	1	
u_3		3	2		4
u_4	2		5		?

In model based CF recommendation systems, regression algorithms are mainly to construct regressors $f_{j,i}$ between different users or different items. These regressors $f_{j,i}$ can be linear or nonlinear or some other complex functions. Finally, the regression algorithms for CF give recommendation by integrating these regressors with different weights.

B. Regression Based Recommendation Algorithms

In a typical CF scenario, there is a matrix D consisting U rows (or users) and I columns (or items). Given scores r_a of an active user on items from I_a, the problem of predicting its score on item i could be solved by learning a nonlinear map

$$f_{I_a,i} : \mathbf{R}^{|I_a|} \rightarrow \mathbf{R} \tag{1}$$

where $f_{I_a,i}$ is a nonlinear function to predict the score on item i given scores on items from I_a.

However, because the nonlinear function is usually very complex and time-consuming, most of current regression algorithms in recommendation systems are in linear "one-to-one" item approach, that is, in practical recommendation systems, some linear functions $f_{j,i}$ where $j \in I_a$ is an item are firstly built, and then $f_{I_a,i}$ is approximated by integrating these linear functions

$$f_{I_a,i} = \sum_{j \in I_a} \alpha_j f_{j,i} \tag{2}$$

where α_j is the ensemble weight of each linear "one-to-one" regressor.

C. Extreme Learning Machine

Extreme Learning Machine (ELM) is first articulated by G.-B. Huang in his 2004 paper [8],and its structure is shown in Fig.1. It is originally derived from the single-hidden layer feed forward neural network (SLFN). If the feature mapping of hidden layer satisfies the universal approximation condition, the hidden layer of SLFN need not be iteratively tuned, which is the difference of ELM from the common SLFN. One of typical implementation of ELM is to generate the weight matrix randomly between the input layer and the hidden layer, and then calculate the output weights by the least-square method. ELM not only tends to reach the smallest training error but also the smallest norm of output weights. This feature brings better generalization performance of ELM.

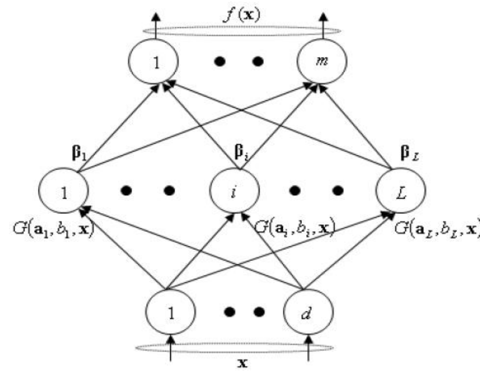


Fig. 1. Illustration of ELM.

The output of an ELM with L hidden nodes can be represented by

$$f_L(\mathbf{x}) = \sum_{i=1}^{L} \beta_i g_i(\mathbf{x}) = \sum_{i=1}^{L} \beta_i G(\mathbf{a}_i, b_i, \mathbf{x}), \mathbf{x} \in \mathbf{R}^d, \beta_i \in \mathbf{R}^m \tag{3}$$

where g_i denotes the output function $G(\mathbf{a}_i, b_i, \mathbf{x})$ of the i^{th} hidden node. For N arbitrary distinct samples $(\mathbf{x}_i, \mathbf{t}_i) \in \mathbf{R}^d \times \mathbf{R}^m$, ELM with L hidden nodes can approximate these N samples with zero error means that there exist $(\mathbf{a}_i, b_i)$ and β_i such that

$$\sum_{i=1}^{L} \beta_i G(\mathbf{a}_i, b_i, \mathbf{x}_j) = \mathbf{t}_j, j = 1, \cdots, N \tag{4}$$

The above N equations can be written compactly as

$$\mathbf{H}\beta = \mathbf{T} \tag{5}$$

where

$$\mathbf{H} = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} G(a_1, b_1, x_1) & \cdots & G(a_L, b_L, x_1) \\ \vdots & \ddots & \vdots \\ G(a_1, b_1, x_N) & \cdots & G(a_L, b_L, x_N) \end{bmatrix}_{N \times L}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m} \quad \text{and } \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix}_{N \times m}$$

The smallest norm least-squares solution of Equation (5) is

$$\widehat{\beta} = \mathbf{H}^{\dagger}\mathbf{T} \tag{6}$$

where $\mathbf{H}^{\dagger}$ is the Moore-Penrose generalized inverse of matrix $\mathbf{H}$. So the regression estimation value matrix $\mathbf{Y}$ can be given by

$$\mathbf{Y} = \mathbf{H}\widehat{\beta} = \mathbf{H}\mathbf{H}^{\dagger}\mathbf{T} \tag{7}$$

Thus, ELM algorithm can be described in Algorithm 1.

Algorithm 1 Extreme Learning Machine (ELM)

Input: a training set $\aleph = \{(\mathbf{x}_i, \mathbf{t}_i) | \mathbf{x}_i \in \mathbf{R}^d, \mathbf{t}_i \in \mathbf{R}^m, i = 1, \cdots, N\}$, hidden node number L and hidden node transfer function $G(\mathbf{a}_j, b_j, \mathbf{x}), j = 1, \cdots, L$.
Output: a trained instance of ELM.

1) Randomly generate hidden node parameters $(\mathbf{a}_j, b_j), \mathbf{a}_j \in \mathbf{R}^d, b_j \in \mathbf{R}$;
2) Calculate hidden node output matrix $\mathbf{H}$;
3) Calculate output weight vector $\widehat{\beta} = \mathbf{H}^{\dagger}\mathbf{T}$.

IV. Collaborative Filtering by Combining Matrix Factorization and Extreme Learning Machine

Given a sparse user-item matrix $X \in \mathbb{R}_{+}^{U \times I}$, Collaborative Filtering (CF) aims to fill the matrix X by machine learning models which minimize the following objective as Equation (8).

$$\begin{aligned} \min f(X) &= \sum_{i=1}^{U} \sum_{j=1}^{I} W_{ij}(X_{ij} - f_j(i))^2 \\ &= \| W \otimes (X - F) \|_F^2 \end{aligned} \tag{8}$$

where $\otimes$ is Hadamard product (element wise product between matrices). $\| \cdot \|_F$ is Frobenius norm. F is the matrix for all the $U \times I$ regressors $f_j(i)$. $f_j(\cdot)$ is some kind of machine learning models, here is the regressor for item j, and $f_j(i)$ is the recommendation value on item j for user i. W is the indicate matrix, W_{ij} is 1 if X_{ij} is not empty, otherwise W_{ij} is 0.

In this section, Extreme Learning Machine is adapted as the regressors, followed by Matrix Factorization Combined Extreme Learning Machine for Collaborative Filtering.

A. Extreme Learning Machine for Collaborative Filtering

In Equation (8), ELM can be used as the regressor to generate recommendation for active users, and this approach can be called Extreme Learning Machine for Collaborative Filtering (ELMCF) algorithm. Given a sparse matrix D_s with U users and I items, a full matrix D_f can be got by setting an initial value to each empty element of D_s. The initial value can be calculated by averaging other known values of its row, column or all the matrix.

For each column of D_f, an ELM regressor can be built. For item i, construct the input matrix P_i and output vector T_i according to D_f, where

$$D_f = \begin{bmatrix} d_{1,1} & \cdots & d_{1,I} \\ \vdots & \ddots & \vdots \\ d_{U,1} & \cdots & d_{U,I} \end{bmatrix} \Rightarrow T_i = \begin{bmatrix} d_{1,i} \\ \vdots \\ d_{U,i} \end{bmatrix}, P_i = \begin{bmatrix} d_{1,1} & \cdots & d_{1,i-1} & d_{1,i+1} & \cdots & d_{1,I} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ d_{U,1} & \cdots & d_{U,i-1} & d_{U,i+1} & \cdots & d_{U,I} \end{bmatrix}$$

Similar to standard ELM algorithm, ELMCF algorithm gets hidden node output matrix H_i according to P_i. Thus, it can get

$$\mathbf{H}_i \beta = \mathbf{T}_i \tag{9}$$

The solution of Equation (9) is

$$\widehat{\beta} = \mathbf{H}_i^{\dagger} \mathbf{T}_i \tag{10}$$

The training error and test error are calculated by Equation (11) and Equation (12).

$$E_{training} = \| \mathbf{H}_i \widehat{\beta} - \mathbf{T}_i \| = \| \mathbf{H}_i \mathbf{H}_i^{\dagger} \mathbf{T}_i - \mathbf{T}_i \| \tag{11}$$

$$E_{test} = \| \mathbf{H}_i' \widehat{\beta} - \mathbf{T}_i' \| = \| \mathbf{H}_i' \mathbf{H}_i^{\dagger} \mathbf{T}_i - \mathbf{T}_i' \| \tag{12}$$

From the above analysis, ELMCF algorithm can be described as Algorithm 2. Compared with the traditional "one-to-one" linear regression algorithms for CF, ELMCF algorithm has several distinct advantages as follows.

- Make the regression more reasonable. In the linear or nonlinear "one-to-one" regression algorithms for CF, it often occurs that the same input is with some different outputs in the training samples. However, based on the nonlinear "all-to-one" regression approach, ELMCF algorithm can effectively avoid this phenomenon.
- The number of regressors is reduced from $O(I^2)$ to $O(I)$. In "one-to-one" approach, it needs to build regressor between each item, so for each item, $(I-1)$ regressors are needed to build, thus the total number of regressors is $I \times (I-1)$. However, in ELMCF algorithm, it only needs to build one regressor for each item, that is, totally I regressors.
- No regressor ensemble is needed any more. In "one-to-one" approach, the final recommendation on item i is calculated by ensemble of all the "one-to-one" regressors on the item. However, in ELMCF algorithm, the recommendation on item i can be directly achieved by ELM regressor $f_i(\cdot)$.

B. Matrix Factorization Combined Extreme Learning Machine for Collaborative Filtering

In the above section, we have introduced ELMCF algorithm to improve the performance of recommendation systems. However, according to our research, there are still two potential problems in ELMCF algorithm, namely the data sparsity problem of CF algorithm and the choice of hidden node number in ELM algorithm.

Algorithm 2 Extreme Learning Machine for Collaborative Filtering (ELMCF)

Input: a training set $\aleph = \{\mathbf{x}_i | \mathbf{x}_i \in \mathbf{R}^I, i = 1, \cdots, N\}$, hidden node number L and hidden node transfer function $G(\mathbf{a}_j, b_j, \mathbf{x}), j = 1, \cdots, L$.
Output: I trained ELM regressors, one for each item.

1) For each empty element of matrix D_s, fill it by averaging the known value of its row, column or all the matrix.
2) **for** $i = 1$ to I **do**
3) Construct input matrix P_i and output vector T_i for item i.
4) Build ELM regressor $f_i(\cdot)$ for item i.
5) **end for**

In practice, the user-item matrix used for collaborative filtering is extremely sparse and the performances of ELMCF algorithm are challenged. Many approaches have been proposed to alleviate the data sparsity problem. Dimensionality reduction techniques, such as Matrix Factorization, map the users or items to a reduced feature space whiling preserving significant information of user-item matrix. One approximates those observed values by a low-rank matrix (using weighted matrix factorization techniques). Unobserved values are predicted according to the learned low-rank matrix.

Besides, the number of hidden layer nodes is one of the most important parameters in ELM algorithm. Too many hidden layer nodes will make ELM overfitting, on the contrary, too few hidden layer nodes will significantly reduce the training accuracy of ELM. How to choose the optimal number of hidden layer nodes is very important for ELM algorithm. Generally, one can make the choice by several different available approaches [18], such as trial-and-error strategy, heuristic search and exhaustive search, et al. In this paper, we resort to the basic trial-and-error strategy to optimize the number of hidden layer nodes, and details refer to Section Experiments.

In this section, we will firstly solve the data sparsity problem by combining ELMCF and Weighted Nonnegative Matrix Tri-Factorization (WNMTF) [15], [3], [17], [7], which aims to minimize the following objective as Equation (13)

$$\begin{aligned} J(X) = \|W \otimes (X - FSG^T)\|_F^2, \\ s.t.\ F \geq 0, G \geq 0, S \geq 0, F^T F = I, G^T G = I \end{aligned} \tag{13}$$

The following update rules are used to compute G, F and S in every iterator.

$$F_{ij} \leftarrow F_{ij} \frac{((W \otimes X)GS^T)_{ij}}{((W \otimes (FSG^T))GS^T)_{ij}} \tag{14}$$

$$S_{jk} \leftarrow S_{jk} \frac{(F^T(W \otimes X)G)_{jk}}{(F^T(W \otimes (FSG^T))G)_{jk}} \tag{15}$$

$$G_{lk} \leftarrow G_{lk} \frac{((W \otimes X)^T FS)_{lk}}{((W \otimes (FSG^T))^T FS)_{lk}} \tag{16}$$

Initialize G, F and S as the following steps. (1) Do K-means clustering of columns of X and obtain the cluster memberships as G and set $G \leftarrow G + 0.2$. (2) Do K-means clustering of rows of X and obtain the cluster memberships as F and set $F \leftarrow F + 0.2$. (3) Initialize $S \leftarrow F^T XG$.

Here, we use WNMTF as a preprocessing step to initialize a full low-rank user-item matrix. Then ELM algorithm is adapted to make recommendations on the learned user-item matrix. Thus, by combining WNMTF with ELMCF algorithm, WNMTF Combined ELM for CF (CELMCF) algorithm can be proposed as shown in Algorithm 3.

Algorithm 3 WNMTF Combined ELM for CF (CELMCF)

Input: a training set $\aleph = \{\mathbf{x}_i | \mathbf{x}_i \in \mathbf{R}^I, i = 1, \cdots, N\}$, hidden node number L, hidden node transfer function $G(\mathbf{a}_j, b_j, \mathbf{x}), j = 1, \cdots, L$, user clustering number K_U and item clustering number K_I.
Output:I trained ELM regressors, one for each item.

1) For each empty element of matrix D_s, fill it by averaging the known value of its row, column or all the matrix.
2) Do K-means clustering of columns to obtain the cluster memberships as G. Set $G \leftarrow G + 0.2$.
3) Do K-means clustering of rows to obtain the cluster memberships as F. Set $F \leftarrow F + 0.2$.
4) Initialize $S \leftarrow F^T XG$.
5) **while** not stopping criterion **do**
6) Update G, F and S according to Equation (16), Equation (14) and Equation (15).
7) **end while**
8) Construct a full low-rank matrix $\widehat{X} = FSG^T$.
9) **for** $i = 1$ to I **do**
10) Construct input matrix P_i and output vector T_i for item i according to $\widehat{X}$.
11) Build ELM regressor $f_i(\cdot)$ for item i.
12) **end for**

V. Experiments

In this section, several experiments are conducted to verify the effectiveness of the proposed ELMCF algorithm and CELMCF algorithm. Generally speaking, experiments are conducted to compare the performance of several CF algorithms, including Memory based CF (MemCF) algorithm [20], Linear Regression for CF (LRCF) algorithm [21], ELMCF algorithm and CELMCF algorithm.

For MemCF algorithm, user-based top-N and item-based top-N recommendation algorithms [20] are chosen to predict unobserved values and generate recommendation in the experiments.

For LRCF algorithm, least squares method is used for collaborative filtering algorithm. It firstly establishes the linear models between users or items, and computes the recommendation by combining these linear models.

A. Data Preparation

In the experiments, there are four benchmark datasets including Movielens dataset, BookCrossing dataset, Jester dataset and Tencent Weibo dataset. Some descriptions of each dataset are shown in Table II. For convenience of calculation, BookCrossing dataset, Jester dataset and Tencent Weibo dataset are sampled to construct the relatively dense matrices respectively, and the data preprocessing results are also shown in Table II.

TABLE II

SUMMARY OF DATASETS

Name	Description	Profile	Data preprocessing result	Availability
Movielens	Ratings of movies on scale from 1 to 5	1,682 users 943 movies and 100,000 ratings	All ratings	www.grouplens.org
BookCrossing	Ratings of books on scale from 1 to 10	278,858 users, 271,379 books and 1,149,780 ratings	1,545 users, 1,729 books and 34,352 ratings	www.grouplens.org
Jester	Continuous ratings of jokes on scale from -10.0 to 10.0	73,421 users, 100 jokes and over 4.1 million continuous ratings	6916 users, 100 jokes and over 553 thousand continuous ratings	www.grouplens.org
Tencent Weibo	Ratings on followees with value of 1 or -1	2,320,895 followers, 6,095 followees and 73,209,277 ratings	6,808 followers, 1,274 followees and 2,779,948 ratings	www.kddcup2012.org

B. Metrics

In the experiments, the performance of all the algorithms are measured by Mean Absolute Error (MAE), which is one of the most popular metrics in recommendation systems.

In statistics, MAE is a quantity used to measure how close forecasts or predictions are to the eventual outcomes, and it can be calculated by Equation (17).

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|e_i| = \frac{1}{N}\sum_{i=1}^{N}|f_i - y_i| \quad (17)$$

C. Experiments Based on Benchmark Datasets

In this section, some experiments are conducted on the four benchmark datasets to verify the performance of CELMCF algorithm. For each dataset, experiments are implemented on totally four algorithms. For Memory based CF (MemCF) algorithm, Linear Regression for CF (LRCF) algorithm and ELMCF algorithm, the sparse matrix is filled by averaging all the values of original matrix. For CELMCF algorithm, the sparse matrix is filled by matrix factorization techniques.

To test the performance of the proposed algorithms on different data sparsity, 20%, 50% and 80% ratings of each dataset are randomly selected as training set and the rest as testing set respectively. All the regression models are built in two basic approaches, that is, item based regression approach and user based regression approach. For each approach, the random selection is carried out 10 times independently, and the average MAEs are reported in Table III and Table IV respectively.

From Table III and Table IV, it can be seen that the proposed CELMCF algorithm can get the best performance of all the four algorithms for the four benchmark datasets in all the cases, and LRCF algorithm tends to get the worst performance.

Another interesting phenomenon is when the training dataset size is small, the training errors and the testing errors of all the four algorithms tend to be large. When the training dataset size becomes large, the training errors and the testing errors become small gradually. It is obvious for machine learning algorithms because the more training samples there are, the more accurate the built model will be.

Besides, the testing errors tend to be larger than the training errors for most of cases. It can be easily understood for the reason that the training samples do not have the completely same distribution space with the testing samples. So the model built according to training samples may be not completely suitable to all the testing samples.

It's worth noting that, in Table III and Table IV, CELMCF algorithm has different advantages on different benchmark datasets. For Tencent Weibo dataset, CELMCF algorithm can get much better training and testing performance than any other algorithms, especially than MemCF algorithm and LRCF algorithm.

In summary, it can be concluded that CELMCF algorithm can get better performance than three other algorithms on the four benchmark datasets. However, different algorithms are suitable for different datasets in data mining.

D. Impact of Parameters

In this section, we will investigate the impact of two important parameters of CELMCF algorithm, including dimensionality of low-rank approximations d in matrix factorization and number of hidden nodes n of ELM, to reveal the relationship between them. The dimensionality of low-rank approximations is set by the grid $\{5, 10, 20, 40, 80\}$. The number of hidden nodes is set by the grid $\{5, 10, 20, 40, 80\}$. Therefore, for each problem we try $5 \times 5 = 25$ combinations of parameters (d, n). Due to the limited space, only the experimental results of Movielens dataset are given in Fig. 2.

From Fig. 2, it can be seen that for both user and item based CELMCF algorithms, when the training dataset size is small, they tend to get their best performance with less hidden nodes, that is, $n = 5$ for Fig. 2(a) and $n = 10$ for Fig. 2(d). When the training dataset size becomes large, they get the best performance with more hidden nodes, that is, $n = 80$ for Fig. 2(b) and Fig. 2(c), and $n = 40$ for Fig. 2(e) and Fig. 2(f). Besides, the parameter sensitivity of used based CELMCF algorithm is similar to that of item based CELMCF algorithm. And number of hidden nodes n tends to have more impact on CELMCF algorithm than rank of matrix approximation d.

VI. CONCLUSIONS AND FUTURE WORK

In recommendation systems, regression algorithms are widely used in model based Collaborative Filtering (CF) approach. However, these regression based CF recommendation algorithms have several inherent flaws, including the serious loss of nonlinear information, the system error caused by regressor ensemble and too many regressors. All

TABLE III
MAE OF ITEM BASED REGRESSION

Dataset	Training Data Size	20%		50%		80%	
	Type of Error	Training	Testing	Training	Testing	Training	Testing
Movielens	MemCF	0.7712	0.8035	0.7461	0.7824	0.7153	0.7652
	LRCF	0.9104	0.9481	0.9066	0.9162	0.8152	0.8461
	ELMCF	0.7404	0.7821	0.7024	0.7528	0.6837	0.7340
	CELMCF	0.6734	0.6926	0.6570	0.6735	0.6406	0.6533
Jester	MemCF	0.7025	0.7206	0.6754	0.6901	0.6457	0.6755
	LRCF	0.8927	0.9017	0.8731	0.8864	0.8626	0.8634
	ELMCF	0.7164	0.7438	0.6925	0.7223	0.6232	0.6469
	CELMCF	0.6984	0.7165	0.6436	0.6825	0.5938	0.6016
BookCrossing	MemCF	0.6654	0.6791	0.6217	0.6548	0.6127	0.6452
	LRCF	0.6703	0.7056	0.6338	0.6713	0.6239	0.6676
	ELMCF	0.6429	0.6742	0.6035	0.6497	0.5663	0.5907
	CELMCF	0.5626	0.6051	0.5245	0.5775	0.5110	0.5209
Tencent Weibo	MemCF	0.2917	0.3548	0.2588	0.3024	0.2377	0.2529
	LRCF	0.5538	0.7013	0.4818	0.6012	0.4539	0.5067
	ELMCF	0.1845	0.2475	0.1746	0.2156	0.1667	0.1909
	CELMCF	0.1195	0.1297	0.1119	0.1053	0.1063	0.1077

TABLE IV
MAE OF USER BASED REGRESSION

Dataset	Training Data Size	20%		50%		80%	
	Type of Error	Training	Testing	Training	Testing	Training	Testing
Movielens	MemCF	0.7719	0.8012	0.7553	0.7822	0.7378	0.7523
	LRCF	0.8855	0.9519	0.9191	0.9236	0.8018	0.8137
	ELMCF	0.8056	0.8374	0.7730	0.8037	0.7578	0.7631
	CELMCF	0.7255	0.7479	0.7101	0.7248	0.6890	0.6948
Jester	MemCF	0.6403	0.6616	0.6358	0.6429	0.6163	0.6251
	LRCF	0.8038	0.8329	0.7920	0.8163	0.7869	0.7832
	ELMCF	0.5920	0.6361	0.5892	0.6011	0.5586	0.5755
	CELMCF	0.5906	0.6097	0.5726	0.5813	0.5561	0.5563
BookCrossing	MemCF	0.6229	0.6321	0.6085	0.6123	0.5885	0.5801
	LRCF	0.6014	0.6463	0.5926	0.6204	0.5832	0.5915
	ELMCF	0.5431	0.5763	0.5306	0.5455	0.5254	0.5273
	CELMCF	0.5125	0.5369	0.5092	0.5209	0.4906	0.5052
Tencent Weibo	MemCF	0.3017	0.3348	0.2815	0.3129	0.2877	0.3092
	LRCF	0.5438	0.5813	0.5386	0.5647	0.5239	0.5367
	ELMCF	0.2145	0.3075	0.1711	0.2659	0.1671	0.2109
	CELMCF	0.1535	0.1997	0.1442	0.1764	0.1423	0.1607

these problems will inevitably reduce the performance of recommendation systems.

In this paper, based on Extreme Learning Machine (ELM), which is originally derived from the single-hidden layer feedforward neural networks and famous for its good generalization performance, ELMCF algorithm is proposed to improve the performance of CF recommendation systems. To alleviate the data sparsity problem, Weighted Nonnegative Matrix Tri-Factorization (WNMTF) is merged into ELMCF algorithm, namely WNMTF Combined ELM for CF (CELMCF) algorithm, to initialize the unobserved ratings of the original user-item matrix. Experimental results on several benchmark datasets from different application domains have shown that the proposed CELMCF algorithm outperforms some state-of-the-art regression based CF algorithms and can get significantly better performance in some cases.

In the future, we intend to focus on improving CELMCF algorithm to enhance its scalability and applicability. For example, weighted nonnegative matrix factorization technique can be further studied to propose more effective missing value filling strategy for collaborative filtering. Besides, with the advent of the age of big data, data analysis by serial algorithms usually can not be completed within acceptable duration. So we intend to design feasible parallel CLEMCF algorithm and implement it based on MapReduce mechanism in the Hadoop framework, which is a simple but powerful parallel programming technique, to provide effective tools for big data recommendation.

REFERENCES

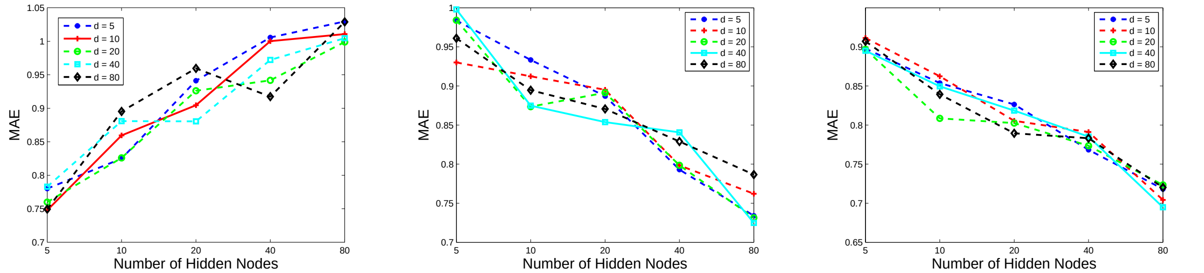
[1] BRENNER, A., PRADEL, B., USUNIER, N., AND GALLINARI, P. Predicting most rated items in weekly recommendation with temporal regression. In *Proceedings of the Workshop on Context-Aware Movie Recommendation*, ACM, pp. 24–27, 2010.

[2] CHANG, T., HSIAO, W., AND CHANG, W. An ordinal regression model with svd hebbian learning for collaborative recommendation. *Journal of Information Science and Engineering*, forthcoming, 2013.

[3] DING, C., LI, T., PENG, W., AND PARK, H. Orthogonal nonnegative matrix tri-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp. 126–135, 2006.

[4] FRANK, E., AND HALL, M. Additive regression applied to a large-scale collaborative filtering problem. *AI 2008: Advances in Artificial Intelligence*, pp. 435–446, 2008.

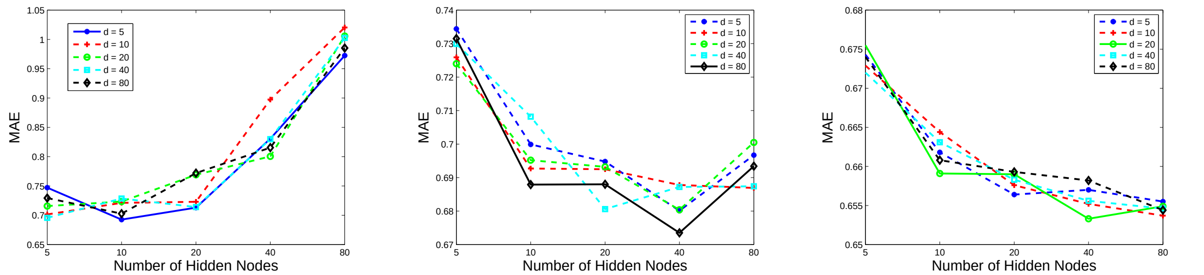
[5] FRIEDMAN, J., HASTIE, T., AND TIBSHIRANI, R. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder



(a) User based CELMCF algorithm with training data size 20%

(b) User based CELMCF algorithm with training data size 50%

(c) User based CELMCF algorithm with training data size 80%



(d) Item based CELMCF algorithm with training data size 20%

(e) Item based CELMCF algorithm with training data size 50%

(f) Item based CELMCF algorithm with training data size 80%

Fig. 2. Impact on Movielens dataset of two different parameters, including rank of matrix approximation d and number of hidden nodes n. Fig. (a)-(c) are user based CELMCF algorithms, and Fig. (d)-(f) are item based CELMCF algorithms.

by the authors). *The annals of statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[6] GOLDBERG, D., NICHOLS, D., OKI, B., AND TERRY, D. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[7] GU, Q., ZHOU, J., AND DING, C. Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In *SIAM SDM*, pp. 199–210, 2010.

[8] HUANG, G., ZHU, Q., AND SIEW, C. Extreme learning machine: a new learning scheme of feedforward neural networks. In *Proceedings of IEEE International Joint Conference on Neural Networks*, vol. 2, pp. 985–990, 2004.

[9] KUNEGIS, J., AND ALBAYRAK, S. Adapting ratings in memory-based collaborative filtering using linear regression. In *IEEE International Conference on Information Reuse and Integration*, pp. 49–54, 2007.

[10] LINDEN, G., SMITH, B., AND YORK, J. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing*, IEEE, vol. 7, no. 1, pp. 76–80, 2003.

[11] MILD, A., AND NATTER, M. Collaborative filtering or regression models for internet recommendation systems?. *Journal of Targeting, Measurement and Analysis for Marketing*, vol. 10, no. 4, pp. 304–313, 2002.

[12] MONTANÉS, E., QUEVEDO, J., DIAZ, I., AND RANILLA, J. Collaborative tag recommendation system based on logistic regression. *ECML PKDD Discovery Challenge*, pp. 173–188, 2009.

[13] PARK, S., AND CHU, W. Pairwise preference regression for cold-start recommendation. In *Proceedings of the third ACM conference on Recommender systems*, ACM, pp. 21–28, 2009.

[14] PURUSHOTHAM, S., LIU, Y., AND KUO, C. Collaborative topic regression with social matrix factorization for recommendation systems. In *Proceedings of the 29th International Conference on Machine Learnin*, pp. 759–766, 2012.

[15] SEUNG, D., AND LEE, L. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, vol. 13, pp. 556–562, 2001.

[16] SHASHUA, A., AND LEVIN, A. Ranking with large margin principle: Two approaches. *Advances in neural information processing systems*, vol. 15, pp. 937–944, 2003.

[17] SREBRO, N., JAAKKOLA, T., ET AL. Weighted low-rank approximations. In *Proceedings of the 20th International Conference on Machine Learning*, vol. 20, pp. 720–727, 2003.

[18] STATHAKIS, D. How many hidden layers and nodes?. *International Journal of Remote Sensing*, vol. 30, no. 8, pp. 2133–2147, 2009.

[19] SU, X., GREINER, R., KHOSHGOFTAAR, T., AND ZHU, X. Hybrid collaborative filtering algorithms using a mixture of experts. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, IEEE Computer Society, pp. 645–649, 2007.

[20] SU, X., AND KHOSHGOFTAAR, T. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, pp. 1–19, 2009.

[21] VUCETIC, S., AND OBRADOVIC, Z. Collaborative filtering using a regression-based approach. *Knowledge and Information Systems*, vol. 7, no. 1, pp. 1–22, 2005.

[22] YU, C., XU, J., AND DU, X. Recommendation algorithm combining the user-based classified regression and the item-based filtering. In *Proceedings of the 8th International Conference on Electronic Commerce*, ACM, pp. 574–578, 2006.

[23] ZHANG, S., WANG, W., FORD, J., MAKEDON, F., AND PEARLMAN, J. Using singular value decomposition approximation for collaborative filtering. In *The Seventh IEEE International Conference on E-Commerce Technology*, IEEE, pp. 257–264, 2005.

[24] ZHU, H., LUO, Y., WENG, C., AND LI, M. A collaborative filtering recommendation model using polynomial regression approach. In *The 4th China Grid Annual Conference*, IEEE, pp. 134–138, 2009.