Chinese Journal of Electronics
Vol.21, No.4, Oct. 2012

A Novel Collaborative Filtering Using Kernel Methods for Recommender Systems*

CAO Jie[1], WU Zhiang[1], ZHUANG Yi[2], MAO Bo[1] and YU Zeng[1]

(1.*Jiangsu Provincial Key Laboratory of E-Business, Nanjing University of Finance and Economics, Nanjing 210003, China*)

(2.*College of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China*)

Abstract — Recommender systems form an essential part of e-business systems. Collaborative filtering (CF), a widely used technique by recommender systems, performs poorly for cold start users and is vulnerable to shilling attacks. Therefore, a novel CF using kernel methods for prediction is proposed. The method is called Iterative kernel-based CF (IKCF), for it is an iterative process. First, mode or mean is used to smooth the unknown ratings; second, discrete or continuous kernel estimators are used to generate predicted ratings iteratively and to export the predicted ratings in the end. The experimental results on three real-world datasets show that, with IKCF as a booster, the prediction accuracy of recommenders can be significantly improved especially for sparse datasets. IKCF can also achieve high prediction accuracy with a small number of iteration.

Key words — Recommender systems, Collaborative filtering, Kernel methods, Prediction.

I. Introduction

A tremendous progress of recommender systems has been witnessed in the past several years, since data in various formats increases explosively on the Web. Recommender systems are changing the way people interact with Web. From e-commerce sites like Amazon.com to news and information sites like dig and Slashdot, recommender systems help people choose from diverse products and mess data by providing a more personalized information access experience[1]. Currently, Collaborative filtering (CF) is the most widely used technique in recommender systems. Tapestry and GroupLens are the two earliest CF-based recommender systems for mail and news respectively[2]. Amazon employs CF algorithm for book recommendation[3]. Facebook also utilizes CF algorithm for advertisement recommendation[4].

In data mining, recommender systems are able to be regarded as the prediction task which is of high importance to business applications[5]. Predicting the preference of users is the key step of CF, and CF-based recommenders often utilize the opinions of its k nearest neighbors (kNN) to identify the user's interested content from an overwhelming set of potential choices. kNN-based CF performs poorly for so-called cold start users who have expressed only a few ratings. Meanwhile, the recommender systems using kNN-based CF are vulnerable to shilling attacks. In such attacks, malicious users create biased rating profiles to manipulate the recommendation output of the system[6].

To solve the inherent problems of kNN-based CF, we propose a novel CF algorithm, called Iterative kernel-based collaborative filtering (IKCF), which predicts the preference of users based on kernel method. . The kernel method was first used in Support vector machines (SVM), and it has been applied to many applications such as classification, principal component analysis, regression, machine learning and missing value imputation[7,8]. In IKCF method, the prediction phase is an iteratively process using the kernel method. First, the missing ratings are smoothed by mode or mean values. Second, the predicted values are generated iteratively based on the known ratings with discrete or continuous kernel functions and the predicted ratings are given in the end. This iteration process will stop if certain criterion is satisfied. Since the missing ratings are smoothed firstly, the accuracy of prediction for cold start users is improved greatly. Moreover, the predicted values are generated iteratively, which will weaken the shilling attackers. Experimental results also show that the proposed IKCF leads to higher prediction accuracy than two kinds of kNN-based CF.

The rest of the paper is structured as follows. Related work is given in Section II. In Section III, we utilize an illustrative example to paraphrase the difference between IKCF and traditional kNN-based CF. Section IV presents the novel IKCF algorithm. The efficiency of the proposed algorithm is illustrated with various kinds of experiments in Section V. Section VI summarizes the whole paper.

II. Related Work

Existing CF algorithms are able to be categorized into the

*Manuscript Received Sept. 2011; Accepted Nov. 2011. This work is supported by the National Natural Science Foundation of China (No.71072172, No.61103229, No.60003047), Industry Projects in the Jiangsu Science & Technology Pillar Program (No.BE2011198), Jiangsu Provincial Key Laboratory of Network and Information Security (Southeast University) (No.BM2003201) and the Program of Natural Science Foundation of Zhejiang Province (No.Y1090165, No.Y1110644, No.Y1110969).

neighborhood-based approach and the latent factor models[9]. Neighborhood-based CF algorithms contain User-based CF (UCF) and Item-based CF (ICF), according to whether neighbors are being computed for the users or items. The kNN algorithm is adopted by both UCF and ICF. For example, if UCF wants to predict the rating of user u for item j ($\widehat{rating_{u,j}}$), it first computes a set of similar neighbors on a user-item matrix $\boldsymbol{N}$, in which element $N(v,i)$ is the rating of user v for item i. Many other methods for similarity measures are also proposed, such as cosine, correctional cosine, and Pearson correlation coefficient (PCC). The predicted rating $rating_{u,j}$ can be generated by combining similarity measures of kNN and their ratings on item j.

Different from the above-mentioned kNN-based CF, this paper utilizes kernel methods for the prediction phase of CF. Kernel function is widely used to build prediction models, such as in Refs.[10, 11] and [12]. Kernel function selection is the key step of applying the kernel method to prediction. Usually, predicted ratings in recommender systems are ordering discrete values, such as MovieLens, Epinions, Jester Joke, Netflix, and so on. Currently there is not too many open recommendation dataset, in which the values are continuous. However, for the sake of integrality, a cube model containing ordering, non-ordering discrete and continuous ratings is presented. Three kinds of kernel functions for dealing with ordering, non-ordering discrete and continuous ratings are also proposed in this paper.

III. An Example

In this section, we utilize an example to demonstrate the difference of our approach from traditional kNN-based CF.

Example 1 We construct a 5∗5 User-Movie matrix from MovieLens datasets. An entry in (u,j) of this matrix represents the rating user u on the j-th movie. The symbol ? represents the unknown rating.

UID/MID	1	2	3	4	5
1	5	3	4	3	3
2	4	?	?	?	?
13	3	?	?	5	1
16	5	?	?	5	?
21	5	?	?	?	2

If we want to make recommendations for UID-2, we should predict the ratings UID-2 on MID-2 to MID-5. The kNN-based CF works as follows. Assume UID-1 and UID-13 are two nearest neighbors of UID-2. The predicted values of (2, 2) and (2, 3) is determined by (1, 2) and (1, 3), because the values of (13, 2) and (13, 3) are unknown. But the predicted values of (2, 4) and (2, 5) are determined by (1, 4), (13, 4) and (1, 5), (13, 5). Since UID-2 only has one rating, any similarity measure is hard to find its real similar neighbors, which decrease the prediction accuracy.

Different from the kNN-based CF, IKCF fills all unknown ratings in User-Movie matrix with the mode of that column. For example, we fill (2, 4) and (21, 4) with 5. Then, ordering discrete kernel function is employed to update new predicted ratings based on the predicted ratings in the last iteration. This process is repeated until certain stopping criterion is satisfied. Details on IKCF are discussed in Section IV.

IV. Algorithm Design

Without loss of generality, a user-item-rating cube $R = (R_{u,i,r})_{n\times m\times w}$ can be generated, where n, m and w are the number of dimensions of users, items and ratings, respectively. Assume there are three types of ratings in the cube R (non-ordering and ordering discrete ratings, and continuous ratings). When the rating r is set, the cube R becomes a $n\times m$ matrix $\boldsymbol{R}_{u,i}(r)$. And there is only one type of values in each matrix $\boldsymbol{R}_{u,i}(r)$, $r = 1, 2, \cdots, w$. Fig.1 depicts the user-item-rating cube R and its three kinds of matrixes.

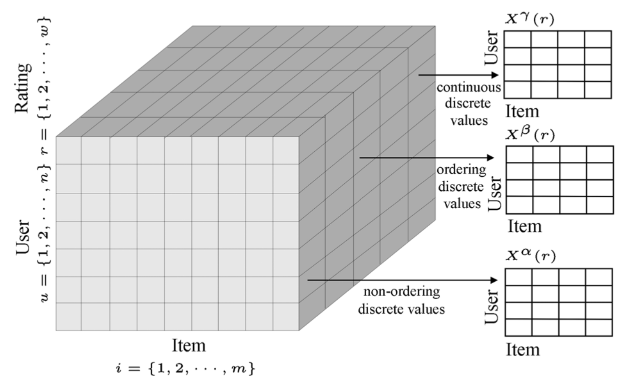

Fig. 1. The user-item-rating cube R

1. Three types of kernel functions

For a certain r, let $X(r)$ be all known values in the matrix $\boldsymbol{R}_{u,i}(r)$, $\boldsymbol{X}_u(r)$ be the u-th ($u = 1, 2, \cdots, n$) row of $\boldsymbol{X}(r)$, and $\boldsymbol{Y}(r)$ be a $n\times 1$ vector with unknown values remained. According to the type of values in $\boldsymbol{R}_{u,i}(r)$, $\boldsymbol{X}(r)$ and $\boldsymbol{Y}(r)$ are separated into three types: $X^{\alpha}(r)$ and $Y^{\alpha}(r)$ for non-ordering discrete values; $X^{\beta}(r)$ and $Y^{\beta}(r)$ for ordering discrete values; $X^{\gamma}(r)$ and $Y^{\gamma}(r)$ for continuous values. We first present three types of kernel functions, upon which estimators to predict the unknown ratings are constructed. Non-ordering discrete values, ordering discrete values and continuous values kernel functions are defined as follows.

Definition 1 (Non-ordering discrete kernel function)

Let $X^{\alpha}_{u,i}(r)$ be the i-th ($i = 1, 2, \cdots, m$) component of $X^{\alpha}_u(r)$ and $d_{u,x} = \sum_{i=1}^{m} I(X^{\alpha}_{u,i}(r), X^{\alpha}_{x,i}(r))$ denote the number of disagreeing components between any two rows of $X^{\alpha}(r)$, where $I(X^{\alpha}_{u,i}(r), X^{\alpha}_{x,i}(r))$ is an indicator function taking value 1 if $X^{\alpha}_{u,i}(r) \neq X^{\alpha}_{x,i}(r)$ and 0 otherwise. The non-ordering discrete kernel function is defined as:

$$\begin{aligned} K_{\alpha,u,x,\lambda} =& K(X^{\alpha}_u(r), X^{\alpha}_x(r), \lambda) \\ =& \prod_{i=1}^{m} l(X^{\alpha}_{u,i}(r), X^{\alpha}_{x,i}(r), \lambda) \\ =& 1^{m-d_{u,x}}\lambda^{d_{u,x}} \\ =& \lambda^{d_{u,x}} \end{aligned} \tag{1}$$

where λ is a smoothing parameter of which the range is in (0,1), and $l(X^{\alpha}_{u,i}(r), X^{\alpha}_{x,i}(r), \lambda) = \lambda$ if $X^{\alpha}_{u,i}(r) \neq X^{\alpha}_{x,i}(r)$ and 1 otherwise.

Definition 2 (Ordering discrete kernel function)

Let $\delta_{u,x} = \sum_{i=1}^{m} |X_{u,i}^{\beta}(r) - X_{x,i}^{\beta}(r)|$ denote the L_1-distance between any two rows of $X^{\beta}(r)$. The ordering discrete kernel function is then defined as:

$$\begin{aligned} K_{\beta,u,x,\lambda} =& K(X_u^{\beta}(r), X_x^{\beta}(r), \lambda) \\ =& \prod_{i=1}^{m} \lambda^{|X_{u,i}^{\beta}(r) - X_{x,i}^{\beta}(r)|} \\ =& \lambda^{\delta_{u,x}} \end{aligned} \tag{2}$$

Definition 3 (Continuous kernel function)

For any two rows $X_u^{\gamma}(r)$ and $X_x^{\gamma}(r)$ in $X^{\gamma}(r)$, let $K_{rbf} = \exp(-\|X_u^{\gamma}(r) - X_x^{\gamma}(r)\|^2/h^2)$ denote the Radial basis function (RBF) and $K_{poly} = (< X_u^{\gamma}(r), X_x^{\gamma}(r) > +1)^b$ denote the polynomial kernel. The continuous kernel function is the combination of RBF and polynomial kernel:

$$K_{\gamma,u,x,h} = \rho K_{rbf} + (1-\rho) K_{poly} \tag{3}$$

where b is the degree of the polynomial, h is the width of the RBF and ρ is the optimal mixed coefficient $(0 \le \rho \le 1)$.

For the continuous vector, it is difficult to determine the parameters b, h and ρ a simultaneously due to the exponential time complexity. However, Jordaan[13] shown experimentally that only a small proportion of RBF is needed to take the ability of interpolation into $K_{\gamma,u,x,h}$. Moreover, Zhu *et al.*[10] also demonstrated that using higher degrees of polynomials or larger widths of RBF is not necessary. Therefore, we suggest set $b = 2$, $h = 0.2$, $\rho = 0.05$ in Eq.(3).

2. Iterative kernel-based collaborative filtering (IKCF)

IKCF is an iterative process and it assigns all unknown ratings in $R_{u,i}(r)$ with the mode or the mean of that column in the first round. Then, three kinds of kernel estimators are used to obtain the predicted values in the t-th round based on the values in the previous round. Therefore, the kernel estimators are the key ingredients of IKCF.

Let the $n \times 1$ vector $\boldsymbol{Y}(r)$ be the target vector and $Y_u(r)$ be the u-th component of $\boldsymbol{Y}(r)$. Since $\boldsymbol{Y}(r)$ contains both known and unknown components, a bool vector $\boldsymbol{\delta}$ is used to represent this difference. Then, the relationship of $\boldsymbol{X}(r)$ and $\boldsymbol{Y}(r)$ can be formulated as Eq.(4):

$$(\boldsymbol{X}(r), Y_u(r), \delta_u), \quad u = 1, 2, \cdots, n \tag{4}$$

where $\delta_u = 0$ when $Y_u(r)$ is known, and $\delta_u = 1$ when $Y_u(r)$ is unknown.

In the t-th round, let $\hat{Y}_x^t(r)$ denote the predicted value of the x-th unknown value in the target vector $\boldsymbol{Y}(r)$. Then, the unknown ratings in the t-th round are predicted with three kinds of kernel estimators based on the values in the $(t-1)$-th round. The definitions of three kernel estimators are described as follows.

Definition 4 (Non-ordering discrete kernel estimator)

Let $D_y^{\alpha} = \{0, 1, \cdots, c_u - 1\}$ denote the range of $Y_u^{\alpha}(r)$, and $n^{-1} \sum_{u=1}^{n} l(Y_u^{\alpha,t-1}(r), y, \lambda) K_{\alpha,u,x,\lambda}$ be the joint density of $\boldsymbol{X}^{\alpha}(r)$ and $\boldsymbol{Y}^{\alpha}(r)$, where $l(Y_u^{\alpha,t-1}(r), y, \lambda) = \lambda$ if $Y_u^{\alpha,t-1}(r) \neq y$ and 1 otherwise. The kernel estimator, $\hat{Y}_x^{\alpha,t}(r)$, of the x-th unknown value in the non-ordering discrete target vector $\boldsymbol{Y}^{\alpha}(r)$ is defined as:

$$\hat{Y}_x^{\alpha,t}(r) = \frac{\sum_{u=1}^{n} \sum_{y \in D_y^{\alpha}, y \neq Y_u^{\alpha,t-1}(r)} y l(Y_u^{\alpha,t-1}(r), y, \lambda) K_{\alpha,u,x,\lambda}}{\sum_{u=1}^{n} K_{\alpha,u,x,\lambda}} \tag{5}$$

Definition 5 (Ordering discrete kernel estimator)

Similar to non-ordering discrete kernel estimator, let $D_y^{\beta} = \{0, 1, \cdots, c_u - 1\}$ denote the range of $Y_u^{\beta}(r)$. The kernel estimator, $\hat{Y}_x^{\beta,t}(r)$, for the ordering discrete target vector $\boldsymbol{Y}^{\beta}(r)$ is defined as:

$$\hat{Y}_x^{\beta,t}(r) = \frac{\sum_{u=1}^{n} \sum_{y \in D_y^{\beta}, y \neq Y_u^{\beta,t-1}(r)} y l(Y_u^{\beta,t-1}(r), y, \lambda) K_{\beta,u,x,\lambda}}{\sum_{u=1}^{n} K_{\beta,u,x,\lambda}} \tag{6}$$

where $l(Y_u^{\beta,t-1}(r), y, \lambda) = \lambda$ if $Y_u^{\beta,t-1}(r) \neq y$ and 1 otherwise.

Definition 6 (Continuous kernel estimator)

The kernel estimator, $\hat{Y}_x^{\gamma,t}(r)$, for the continuous target vector $\boldsymbol{Y}^{\gamma}(r)$ is defined as:

$$\hat{Y}_x^{\gamma,t}(r) = \frac{n^{-1} \sum_{u=1}^{n} Y_u^{\gamma,t-1}(r) K_{\gamma,u,x,h}}{n^{-1} \sum_{u=1}^{n} K_{\gamma,u,x,h} + n^{-2}} \tag{7}$$

In Eqs.(5), (6) and (7), the value of $Y_u^{\theta,t-1}(r)$ ($\theta = \alpha$, β or γ) is equal to $Y_u^{\theta}(r)$ if $\delta_u = 0$ and $\hat{Y}_u^{\theta,t-1}(r)$ if $\delta_u = 1$. That is

$$Y_u^{\theta,t-1}(r) = \begin{cases} Y_u^{\theta}(r), & \text{if } \delta_u = 0 \\ \hat{Y}_u^{\theta,t-1}(r), & \text{if } \delta_u = 1 \end{cases} \quad \text{where } \theta = \alpha, \beta \text{ or } \gamma \tag{8}$$

Based on the Eqs.(5), (6), (7), the pseudo-code of the IKCF is shown in Table 1.

Table 1. Pseudo-code of the IKCF

Input: R: the user-item-rating cube
$maxIter$: the max number of iteration
ϵ: the threshold of stopping criterion
Output: Recommendation list for any user
1: **if** (the type of values in $R_{u,i}(r)$ is non-ordering discrete or ordering discrete)
2: all unknown values are filled with the mode of that column
3: **else if** (the type of values in $R_{u,i}(r)$ is continuous)
4: all unknown values are filled with the mean of that column
5: **end of if** //initialization
6: **for** $t \leftarrow 1$ *to maxIter* //iteratively predicting the unknown values
7: **if** (the type of values in $R_{u,i}(r)$ is non-ordering discrete)
8: compute $\hat{P}R^t$ by Eq.(5)
9: **else if** (the type of values in $R_{u,i}(r)$ is ordering discrete)
10: compute $\hat{P}R^t$ by Eq.(6)
11: **else if** (the type of values in $R_{u,i}(r)$ is continuous)
12: compute $\hat{P}R^t$ by Eq.(7)
13: **end of if**
14: **if** ($\|\hat{P}R^t - \hat{P}R^{t-1}\|_{\infty} < \varepsilon$)
15: **break**
16: **end of if**
17: **end of for**
18: Generate a recommendation list based on the predicted ratings.

In line 1 to line 5, the unknown ratings in $R_{u,i}(r)$ are initialized. Line 6 ∼17 are the iterative process of IKCF. After unknown ratings in $R_{u,i}(r)$ are predicted, we can make decisions on recommendation for any user. Generally speaking, there are two methods for decisions making in recommendation. (1) Select top N target items with higher ratings for users. This method has to generate recommendation results even though some items may have low ratings and useless to the user. (2) Define a threshold value, and if the rating is greater than the threshold value, the item is selected. This method can also produce useful recommendation for the user.

Norm, a concept in mathematics, is employed to be the convergence condition. A norm is a function that assigns a strictly positive length to all vectors in a vector space, and all types of norm are proven to be equivalent[14]. To simplify the calculation, we utilize the infinite norm as the convergence condition. Let PR denote a set of all unknown values in matrix $\boldsymbol{R}_{u,i}(r)$, and the $|PR|$ is size of PR. Let $\hat{P}R^t = (\hat{P}R_1^t, \hat{P}R_2^t, \cdots, \hat{P}R_{|PR|}^t)$ denote the predicted values of all unknown ratings in the t-th iteration. The infinite norm between $\hat{P}R^t$ and $\hat{P}R^{t-1}$ is then given by Eq.(9).

$$\|\hat{P}R^t - \hat{P}R^{t-1}\|_\infty = \text{Max}(|\hat{P}R_1^t - \hat{P}R_1^{t-1}|, |\hat{P}R_2^t - \hat{P}R_2^t|, \cdots, |\hat{P}R_{|PR|}^t - \hat{P}R_{|PR|}^{t-1}|) \tag{9}$$

V. Experiments

This section reports the experimental results of various versions of CF applied on the real datasets. These results indicate that the proposed IKCF can achieve superior performances comparing to UCF and ICF on sparse datasets.

1. Experimental design

The proposed IKCF is implemented by C++ program language in Visual Studio environments. Standard UCF and ICF are also implemented as two fundamental versions of CF. The experiment system is running on an Intel Core2 2.33 GHz CPU with 3GB RAM with Windows XP SP3 system.

There are a number of datasets for recommendation, among which three typical data sets are chosen. Jester Joke data set released by UC Berkeley contains 4.1 million continuous ratings (−10.00 to +10.00) of 100 jokes from 73,496 users[15]. The MovieLens data set consists of 100,000 ratings on 1682 movies by 943 users[16]. All the ratings are integer values between one and five where one is the lowest (disliked) and five is the highest (most liked). Article ratings dataset of Epinions datasets is utilized where the ratings represent how much a certain user rates a certain textual article written by another user[17]. Jester Joke is a dense dataset, and Epinions dataset is very sparse. The dense degree of MovieLens dataset is moderate.

Three above-mentioned datasets contain ordering discrete values. To the author's knowledge, there is no open recommendation dataset of which the values are continuous. Although we only use the ordering discrete estimator for prediction, the experimental results illustrate the effectiveness of IKCF. Meanwhile, we utilize Normalized mean absolute error (NMAE) to measure the error in recommendation:

$$NMAE = \frac{MAE}{(\sum_{|PR|} P_i)/|PR|},$$

where
$$MAE = \frac{\sum_{|PR|} |P_i - \hat{P}_i|}{|PR|} \tag{10}$$

NMAE is in [0, 1]. Small value of NMAE indicates a precise recommendation.

2. Experimental results

The first experiment compares NMAE performance of IKCF, UCF and ICF for all users in three datasets. In IKCF, we set λ to 0.6 and the number of iterations to 8. As Fig.2 shows, IKCF is more precise than UCF and ICF in three datasets, and especially for the sparse dataset (*e.g.* Epinions), NMAE of IKCF is far lower than that of UCF and ICF. It appears that because the similarity measure of UCF and ICF cannot reflect the truth for the sparse dataset.

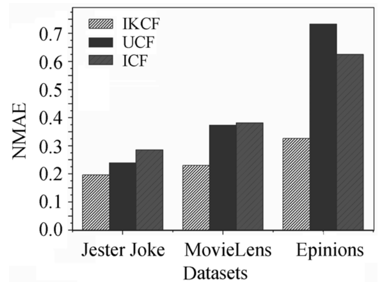

Fig. 2. NMAE Performance comparison for all userse

The second experiment studies the impact of the number of iteration of IKCF. We set λ to 0.6 and record the NMAE of IKCF from the first iteration to the tenth iteration. The variation of NMAE in each round is shown in Fig.3. Combining the results shown in Fig.2 and Fig.3, NMAE of IKCF in the first round is worse than that of UCF and ICF because the unknown values are predicted by mode in IKCF. From the second round, the performances of IKCF are better than UCF and ICF. When the number of iteration reaches 4 to 6, IKCF performs drastically better than UCF and ICF. However, the performances of IKCF tend towards stability with the continuing increase of the number of iteration.

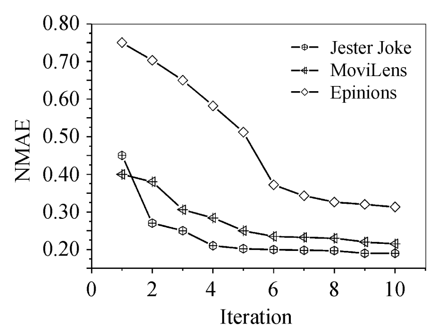

Fig. 3. The number of iteration of IKCF on NMAE

The third experiment investigates the impact of the smoothing parameter λ. We set the number of iterations to 8, and vary the value of λ from 0.1 to 0.9 and the interval is set to 0.1. Fig.4 shows the NMAE of IKCF for three datasets varies with the increase of λ. Three curves exhibit similar form. The three curves hit a trough when the value of λ is intermediate. From the results shown in Fig.4, it is appropriate that λ be set between 0.5 to 0.7 in IKCF. If we want to find the rigorous value of λ, the cross-validation approach may be utilized. In our experiments, we set λ to 0.6 which leads to the satisfactory performance of IKCF.

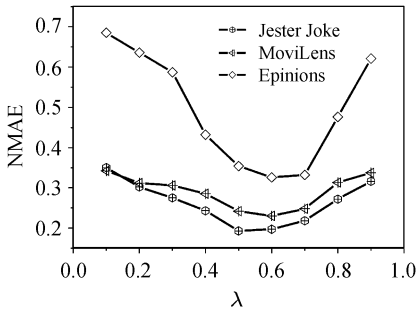

Fig. 4. The impact of λ of IKCF on NMAE

The last experiment ascertains the performance of IKCF for cold-start users. In Epinions dataset, we consider users with less than 5 ratings as cold start users the number of which is 24,000. But in MovieLens dataset, there are 2 users with less than 10 ratings and more than 100 users with less than 20 ratings among 943 users. So we consider users with less than 20 ratings as cold start users. Since the Jester Joke dataset is too dense, we do not select cold start users from it. As shown in Fig.5, the performances of IKCF are significantly better than that of UCF and ICF. In addition, ICF performs better than UCF because it is difficult to find neighborhoods for the cold start user.

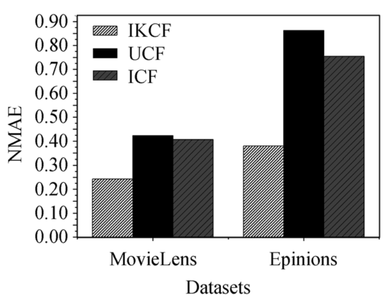

Fig. 5. NMAE Performance Comparison for cold start users

In summary, the proposed IKCF substantially improves the performance of prediction especially on sparse datasets. This improvement is achieved by filling the mode for unknown ratings in the first round and utilizing kernel estimators to predict unknown ratings iteratively. IKCF clearly outperforms traditional CF (both user-based and item-based) in terms of precision. Since cold start users have only a few ratings, the improvement of NMAE using IKCF compared to UCF and ICF is much more for cold start users than for all users.

VI. Conclusion

It is observed that several inherent problems including cold start users and vulnerability to shilling attacks cannot solved by traditional kNN-based CF. We present IKCF, a novel iterative CF using kernel methods for prediction. The proposed infinite norm is tested suitable as the stopping criterion. For the sake of integrality, a cube model including various formats of ratings is presented. The cube model is the extension of user-item matrix and is pervasive for any recommender system. Our experiments compare IKCF with UCF and ICF on three real-world datasets including Jester Joke, MovieLens and Epinions. Meanwhile, the impact of the number of iteration and the smoothing parameter λ are investigated. IKCF can achieve high prediction accuracy with a small number of iteration, and tend towards stability with the continuing increase of the number of iteration. We also suggest to use the cross-validation to select the value of λ because carefully determining the intermediate value of λ can improve the prediction accuracy further. Experimental results show that IKCF performs better than them on sparse datasets. Also, IKCF can significantly improve the prediction accuracy for cold start users.

References

[1] J. Riedl, B. Smyth, "Introduction to special issue on recommender systems", *ACM Transactions on the Web*, Vol.5, No.1, Article 1, pp.1–2, 2011.

[2] F. Cacheda, V. Carneiro, D. Fernandez and V. Formoso, "Comparison of collaborative filtering algorithms: limitations of current techniques and proposals for scalable, high-performance recommender systems", *ACM Transactions on the Web*, Vol.5, No.1, Article 2, pp.1–33, 2011.

[3] G. Linden, B. Smith and J. York, "Amazon.com recommendations: item-to-item collaborative filtering", *IEEE Internet Computing*, Vol.7, No.1, pp.76–80, 2003.

[4] L.J. Fang, H. Kim, K. LeFevre and A. Tami, "A privacy recommendation wizard for users of social networking sites", *Proceedings of the 17th ACM Conference on Computer and Communications Security*, Chicago, IL, USA, pp.630–632, 2010.

[5] J.J. Wu, H. Xiong and J. Chen, "COG: Local decomposition for rare class analysis", *Data Mining and Knowledge Discovery*, Vol.20, No.2, pp.191–220, 2010.

[6] Z.A. Wu, J. Cao, B. Mao and Y.Q. Wang, "Semi-SAD: applying semi-supervised learning to shilling attack detection", *The 5th ACM Conference on Recommender Systems*, Chicago, IL, USA, pp.289–292, 2011.

[7] C.A. Micchelli, M. Pontil, "Learning the kernel function via regularization", *Journal of Machine Learning Research*, Vol.6, pp.1099–1125, 2005.

[8] J.B. Li, L.J. Yu and S.H. Sun, "Refined kernel principal component analysis based feature extraction", *Chinese Journal of Electronics*, Vol.20, No.3, pp.467–470, 2011.

[9] Y. Koren, "Factor in the neighbors: scalable and accurate collaborative filtering", *ACM Transactions on Knowledge Discovery from Data*, Vol.4, No.1, pp.1–24, 2009.

[10] X.F. Zhu *et al.*, "Missing value estimation for mixed-attribute data sets", *IEEE Transactions on Knowledge and Data Engineering*, Vol.23, No.1, pp.110–121, 2011.

[11] S.C. Zhang, Z.J. and X.F. Zhu, "Missing data imputation by utilizing information within incomplete instances", *The Journal of Systems and Software*, Vol.84, No.3, pp.452–459, 2011.

[12] Y.S. Qin *et al.*, "POP algorithm: kernel-based imputation to treat missing values in knowledge discovery from databases", *Expert Systems with Applications*, Vol.36, No.2, pp.2794–2804, 2009.

[13] E.M. Jordaan, "Development of robust inferential sensors: industrial application of support vector machines for regression", *Ph.D. Thesis*, Technical University Eindhoven, 2002.

[14] R.A. Horn, C.R. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.

[15] Jester Joke. *http://www.ieor.berkeley.edu/~goldberg/jester-data*, 2011.

[16] GroupLens Research. *http://www.grouplens.org/node/73*, 2011.

[17] P. Massa and P. Avesani, *Trust Metrics in Recommender Systems, Computing with Social Trust: Human-Computer Interaction Series*, Springer, pp.259–285, 2009.

CAO Jie received the Ph.D. degree from Southeast University, China, in 2002. He is currently a professor and the director of Jiangsu Provincial Key Laboratory of E-Business at Nanjing University of Finance and Economics. He has been selected in the Program for New Century Excellent Talents in University (NCET). His main research interests include cloud computing, business intelligence and data mining. He has published one book and more than 40 refereed papers in various conferences and journals. He has taken charge of over 20 national projects and obtained Jiangsu Provincial Scientific and Technological Progress Second Prizes Awarded (2008).

WU Zhiang (corresponding author) received the Ph.D. degree from Southeast University, China, in 2009. He is currently an associate professor of Jiangsu Provincial Key Laboratory of E-Business at Nanjing University of Finance and Economics. He is the member of the CCF and ACM. Dr. Wu is a recipient of the Excellent Student Paper Award of IFIP Intl. Conf. on Network Parallel Computing (2007). His recent research interests include recommender systems, cloud computing and data mining. Dr. Wu has authored two books and more than 20 refereed papers in international journals and conferences. (Email: zawuster@gmail.com)

ZHUANG Yi is currently an associate professor at the College of Computer and Information Engineering in Zhejiang Gongshang University where he joined as faculty member since May 2008. Dr. Zhuang is a recipient of the CCF Doctoral Dissertation Award conferred by Chinese Computer Federation in 2008 and IBM Ph.D. Fellowship 2007-2008. He obtained his Ph.D. degree in computer science from Zhejiang University in Mar. 2008. Dr. Zhuang is a member of ACM, IEEE and CCF. Dr. Zhuang has published more than 30 high-quality papers and co-authored two books or chapters, and 3 patents have been issued. Dr. Zhuang is engaged in the two aspects of research issues: index techniques (multimedia or uncertain) and cloud computing, *etc.*

MAO Bo received the Ph.D. degree from Royal Institute of Technology (KTH), Sweden in 2011 and the B.S. and M.S. degrees from Xi'an Jiaotong University and Southeast University in China in 2005 and 2008, respectively. He is currently an associate professor of Jiangsu Provincial Key Laboratory of E-Business at Nanjing University of Finance and Economics. His recent research interests include 3D visualization and data mining. Dr. Mao has published over 20 refereed papers in international journals and conferences.

YU Zeng received B.S. and M.S. degrees from the Department of Mathematics, School of Sciences, China University of Mining and Technology in 2008 and 2011, respectively. He is currently working as a researcher in Jiangsu Provincial Key Laboratory of E-Business at Nanjing University of Finance and Economics. His current research interests include recommender systems and data mining.