University of the Peloponnese
Department of Informatics and Telecommunications
Software and Database Systems Laboratory
https://soda.dit.uop.gr

Experimental results for considering Item Adoption Eagerness Information in Collaborative Filtering's Rating Prediction

Technical Report TR-19003

Dionisis Margaris, Dimitris Spiliotopoulos, Costas Vassilakis,
margaris@di.uoa.gr, dspiliot@uop.gr, costas@uop.gr,

July, 2019
Tripoli, Greece

1. Introduction

In this technical report, we present the experimental findings from applying an algorithm that considers Item Adoption Eagerness Information in the rating prediction formulation process, in order to increase rating prediction quality in Collaborative Filtering (CF).

To this end, the algorithm is applied to seven datasets, which are widely used in recommender system (RS) research.

In short, the algorithm moderates the weight that each individual rating $r_{V,i}$ of user V on an item i is taken into account when formulating a rating prediction $p_{U,i}$ on that specific item for another user U, by considering the item adoption eagerness information; this aspect is reflected on the particular item's adoption phases that the registered rating $r_{V,i}$ and the prediction $p_{U,i}$ to be formulated fall in. More specifically, the weight assigned to the rating $r_{V,i}$ in the context of recommending item i to user U is

$$IAE\_factor(U,V,i) = \begin{cases} \mathrm{EA}, & if\ \ r_{U,i}\ and\ r_{V,i} \in \mathrm{EA}_i \\ \mathrm{LA}, & if\ \ r_{U,i}\ and\ r_{V,i} \notin \mathrm{EA}_i \\ \mathrm{DIFF}, & \mathrm{otherwise} \end{cases} \qquad (1)$$

In formula (1) the *EA* is a constant that is used when both users' ratings on item *i*, belong to the item's *EA* lifetime phase; and -similarly- *LA* is a constant employed when both users' ratings on item *i*, belong to its *Late Adoption* lifetime phase. The *DIFF* constant is employed when the two ratings belong to different lifetime phases of item *i* (Early and Late).

In the presented experiments, the optimal values for the parameters that are used in the algorithm are investigated.

2. Algorithm Tuning and Performance Evaluation

In this section, we report on the experiments that were designed to:

1. Determine the optimal parameters values for the *EA*, *LA* and *DIFF* parameters, used in the *IAE_factor* function of the presented algorithm.
2. Compute the prediction improvement, introduced by the presented algorithm, due to the consideration of the *item adoption eagerness* information in the CF rating prediction computation process.

In order to determine the optimal parameters values, we experimentally explored the parameter value solution space, by iteratively selecting parameter value assignments and examining the effect that the particular parameter value assignments have on rating prediction quality. To quantify rating prediction quality, we employed two widely used error metrics, namely the Mean Absolute Error (MAE), and the Root Mean Squared Error (RMSE). The use of two different metrics allows us to gain more detailed insight on the prediction accuracy achieved by each parameter setting, since the MAE metric handles all error scales in a uniform fashion, whereas the RMSE metric penalizes more severely larger errors. To compute the algorithm's prediction error, in terms of MAE and RMSE, we exercised the standard "hide one" technique [1,2,3]: each user's last rating in the database was hidden and then its value was predicted on the basis of the values of other, non-hidden ratings. We also performed a second experiment where, for each user, a random rating was hidden, and again its value was predicted on the basis of the values of other, non-hidden ratings. The results obtained from the two experiments were in close agreement (the differences observed were less than 1.8% in all cases), therefore for conciseness purposes we report only on the results of the first experiment. All our experiments were run on seven datasets. Five of these datasets are obtained from Amazon [4,5] and two from MovieLens

Table 1: Datasets Summary

Dataset name	**#Users**	**#Items**	**#Ratings**	**Avg. #Ratings / User**	**Density**	**DB size (in text format)**
Amazon "Videogames" [4,5]	8K	50K	157K	19.4	0.039%	4 MB
Amazon "CDs and Vinyl" [4,5]	41K	486K	1.3M	31.6	0.006%	32MB
Amazon "Movies and TV" [4,5]	46K	134K	1.3M	28.0	0.021%	31MB
Amazon "Books" [4,5]	295K	2.3M	8.7M	29.5	0.001%	227MB
Amazon "Digital Music" [4,5]	6K	35K	86K	13.9	0.040%	2MB
MovieLens "Latest 100K – Recommended for education and development" [6,7]	700	9K	100K	142.8	1.587%	2MB
MovieLens "Latest 20M – Recommended for new research" dataset [6,7]	138K	27K	20M	144.9	0.537%	486MB

[6,7]; the Amazon datasets are relatively sparse, while the MovieLens datasets are relatively dense (a dataset DS is deemed to be very sparse if $d(DS) \ll 1\%$, where $d(DS)$ is the density of the dataset, defined as $d(DS) = \frac{\#ratings}{\#users * \#items}$ [8]). We choose to test both sparse and dense datasets, in order to establish that the proposed algorithm can be used in every dataset.

The seven datasets used in our experiments are summarized in Table 1 and have the following characteristics:

1. They are up to date (published between 1996 and 2016).
2. They are widely used as benchmarking datasets in CF research.
3. They contain each rating's timestamp, necessary in the proposed algorithm.
4. They differ in regards to the type of item domain of the dataset (videogames, movies, music and books) and size (ranging from 2MB to 486MB in plain text format).

Each dataset was initially preprocessed, and users found to have less than 10 ratings were dropped, since predictions formulated for users with few ratings are known to demonstrate high error levels [1,9]. This procedure did not have any effect on the MovieLens dataset, since it includes only users that have submitted at least 20 ratings.

For our experiments we used a machine equipped with six Intel Xeon E7 - 4830 @ 2.13GHz CPUs, 256GB of RAM and one 900GB HDD with a transfer rate of 200MBps, which hosted the datasets and ran the rating prediction algorithms.

In the remainder of this section, we present and discuss the results obtained from applying the algorithm presented above on these seven datasets.

2.1 Determining the optimal algorithm's parameters

The goal of the first experiment is to determine the optimal values regarding the *EA*, *LA* and *DIFF* parameters, used in the presented algorithm (c.f. equation 1).

Recall from section 1 that the relevant parameters of the presented algorithm are:

1. *EA*, moderates the weight of the a NN's rating to the prediction formulated for a user *U* in the case that both users have evaluated the item in its *EA* lifetime phase,
2. *LA* moderates the weight of the a NN's rating to the prediction formulated for a user *U* in the case that both users have evaluated the item in its *LA* lifetime phase and
3. *DIFF*, moderates the weight of the a NN's rating to the prediction formulated for a user *U* in the case that the NN's rating falls within the *EA* period of item *i* and the prediction falls within the *LA* period, or vice versa.

From equations (4) and (5) we can observe that the final outcome of a rating prediction calculation depends on the ratios $\frac{EA}{LA}$ and $\frac{LA}{DIFF}$, rather than on the absolute values of *EA*, *LA* and *DIFF*. Therefore, in the conducted experiments we have fixed the value of *LA* to 1 and varied the values of *EA* and *DIFF*, exploring their domain space. Regarding the values of the *EA* parameter, we consider only values that are greater than or equal to1.0, under the rationale that two *EA* evaluations on the same item are more important than two *LA* ones, since, while early (late) adopters are bound to share the same mentality and criteria with other early (late) adopters, the probability of having two early adopter ratings is significantly smaller than having two late adopter ones. Respectively, regarding the values of the *DIFF* parameter, we consider only values that are less than or equal to 1.0, under the rationale that the utility of using an early (late) adoption period rating –which has been given following some specific mentality and criteria- to predict a late (early) adoption period rating -which would be formulated following different mentality and criteria- is low. In order to find the optimal setting for the aforementioned parameters, we explored different combinations of values for these parameters. In total, more than 70 value combinations were examined, however, in the rest of this paper we report only on the most indicative ones, for conciseness purposes.

Figure 1 illustrates the average MAE reduction, of all the datasets summarized in Table 1, under different parameters values combinations, when similarity is measured using the PCC similarity metric.

In Figure 1, we can observe that the setting delivering the highest reduction in the MAE is when the *DIFF* parameter is set to 0.5 and the *EA* parameter is set to 2.0, achieving an average MAE reduction of 3.7%.

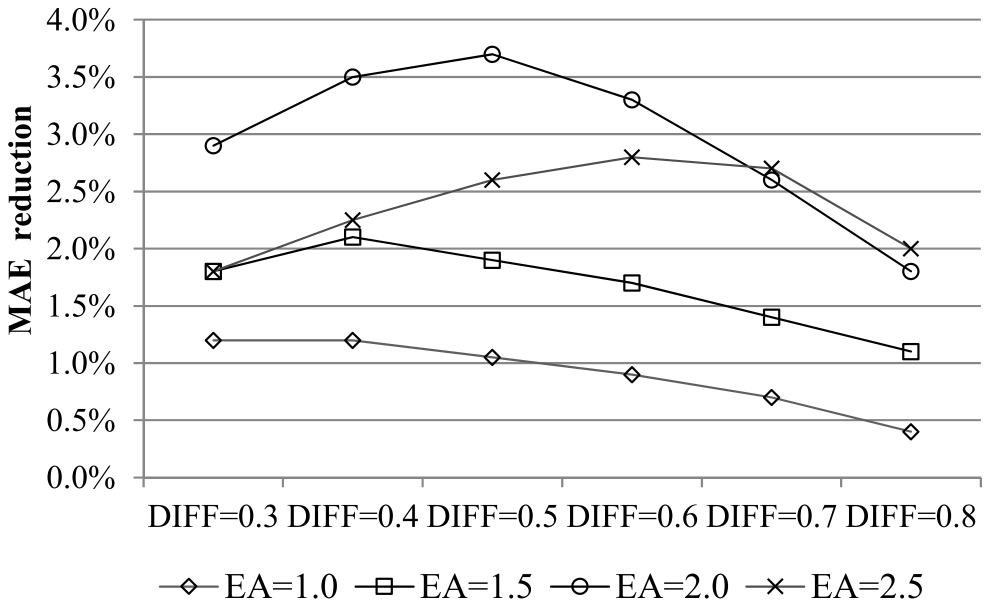

Figure 1: MAE reduction under different EA and DIFF parameter value combinations, using the PCC similarity metric

Figure 2 pictures the average RMSE reduction of all the datasets summarized in Table 1, under different parameter values combinations, again when similarity is measured using the PCC similarity metric.

In Figure 2, we can observe that the setting attaining the highest reduction in RMSE, is again when the *DIFF* parameter is set to 0.5 and the *EA* parameter is set to 2.0. This setting achieves an average RMSE reduction of 3.18%. Generally, the reductions achieved for the RMSE metric follow the same pattern observed for the MAE metric in Figure 1. The magnitude of the RMSE reduction is smaller than that of the MAE; the latter fact indicates that the algorithm manages to correct mostly small errors.

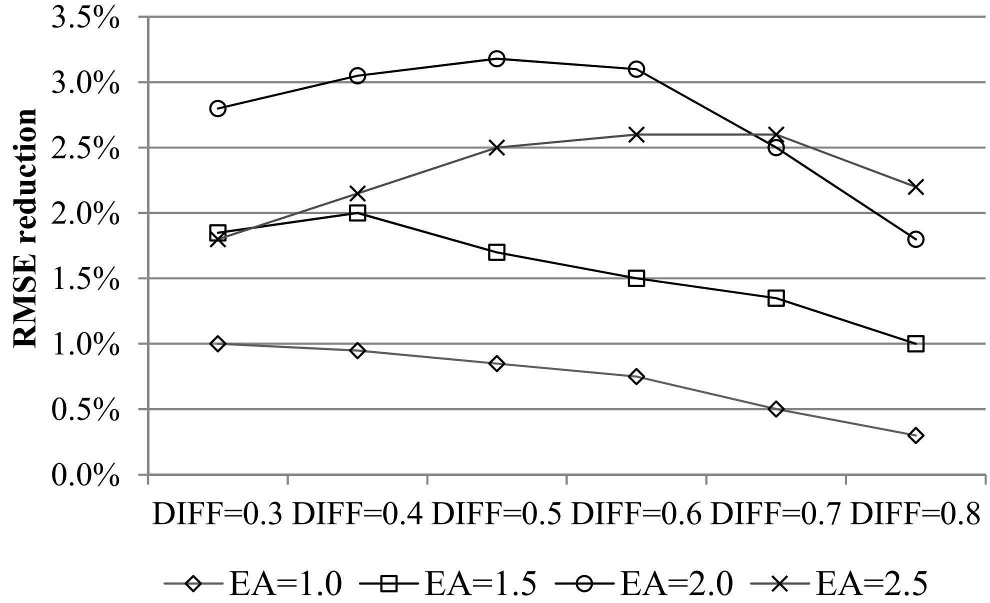

Figure 2: RMSE reduction under different *EA* and *DIFF* parameter value combinations, using the PCC similarity metric

Figure 3 depicts the average MAE reduction of all the datasets summarized in Table 1, under different parameter values combinations, when similarity is measured using the CS similarity metric.

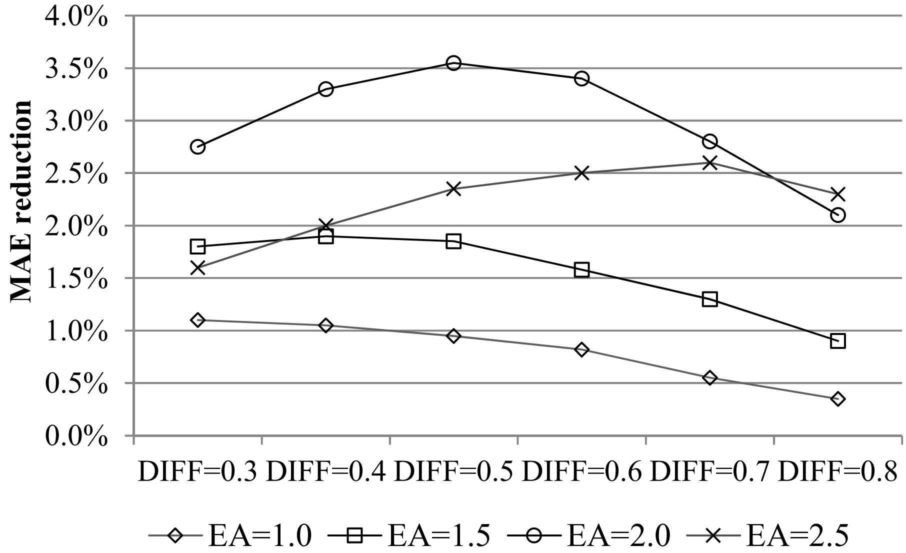

Figure 3: MAE reduction under different *EA* and *DIFF* parameter value combinations, using the CS similarity metric

In Figure 3, we notice that the setting combination, delivering the highest reduction in MAE, is again when the *DIFF* parameter is set to 0.5 and the *EA* parameter is set to 2.0, achieving an average MAE reduction of 3.55%.

Figure 4 illustrates the average RMSE reduction of all the datasets summarized in Table 1, under different parameters values combinations, when similarity is measured using the CS similarity metric. In Figure 4, we can observe that the setting attaining the highest reduction in the MAE is again when the *DIFF* parameter is set to 0.5 and the *EA* parameter is set to 2.0; this setting achieves an average RMSE reduction of 3.02%.

Similarly to the case when PCC similarity is used, the patterns of the MAE and the RMSE reduction under the CS similarity metric are highly alike. Again, the reduction achieved for the RMSE metric lags behind that of the MAE, indicating that the algorithm corrects mostly small errors.

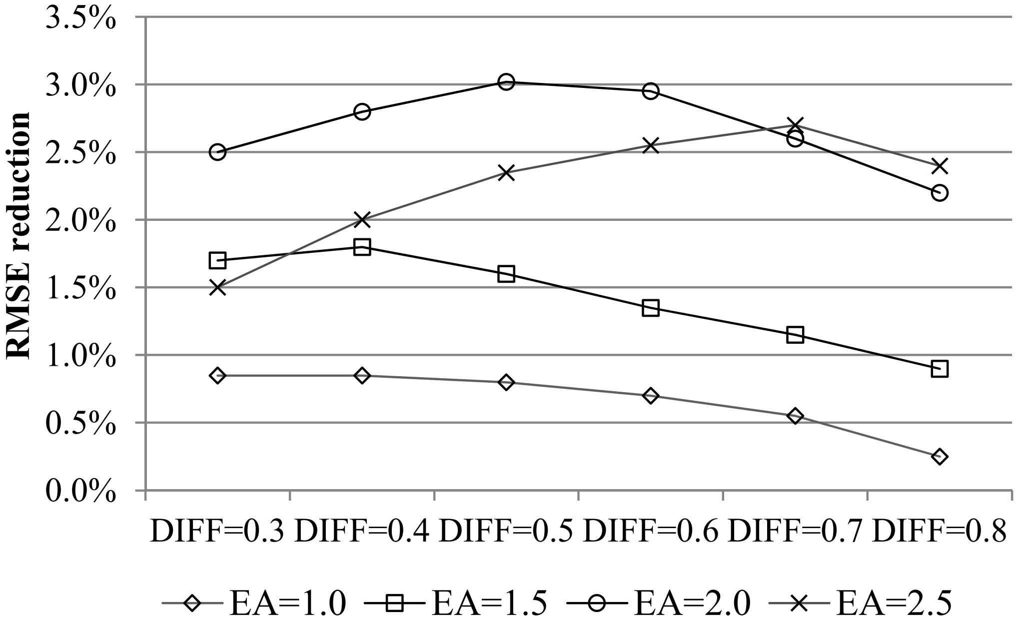

Figure 4: RMSE reduction under different *EA* and *DIFF* parameter value combinations, using the CS similarity metric

The datasets used in the experiment are summarized in Table I, and the results obtained are listed in the following subsections. In the results presentation subsections, cells with a gray background indicate cases where the rating prediction accuracy of the proposed algorithm surpasses that of the plain CF algorithm, while cells with bold typeface indicate that the respective cell corresponds to the optimal performance (rating prediction accuracy or coverage) achieved.

2.2 Performance evaluation

In this section, we compare the results produced by the presented algorithm, with the ones produced by the CF variability algorithm, proposed in [3]. This algorithm was chosen for the comparison since it (i) is a state-of-the-art algorithm (proposed in 2018), targeting the improvement of prediction accuracy in the context of CF, (ii) does not need extra information, regarding the users or the items (e.g. item categories or user social relationships) and (iii) does not deteriorate the prediction coverage. We note here that no other algorithm addresses the particular aspect of user behavior considered by the proposed algorithm; thus, in the absence of such an algorithm, the comparison is made with an algorithm that exploits similar features of ratings (i.e. temporal features).

Considering the optimal values of the *EA* and *DIFF* parameters, based on the results presented in the previous subsection, we can clearly see that the settings of 2.0 and 0.5, respectively, proved to be the optimal ones, hence in the experiments presented hereafter we will use these particular settings.

Figure 5 depicts the improvement in the MAE achieved by the proposed algorithm, when compared to the CF variability algorithm, proposed in [3], taking the performance of the plain CF algorithm as a baseline and using the PCC as the similarity metric, since this is the one tested in [3].

Clearly, the proposed algorithm surpasses the performance of the CF variability algorithm, in all the datasets tested, with its MAE reduction being 63.4% higher than that achieved by the CF variability algorithm (3.7% against 2.26% in absolute figures). At individual dataset level, the performance edge of the proposed algorithm against the CF variability algorithm ranges from 37% to 414%.

Figure 6 depicts the respective improvement in the RMSE achieved by the proposed algorithm, when compared to the CF variability algorithm, proposed in [3], again taking the performance of the plain CF algorithm as a baseline, and under the PCC as the similarity metric.

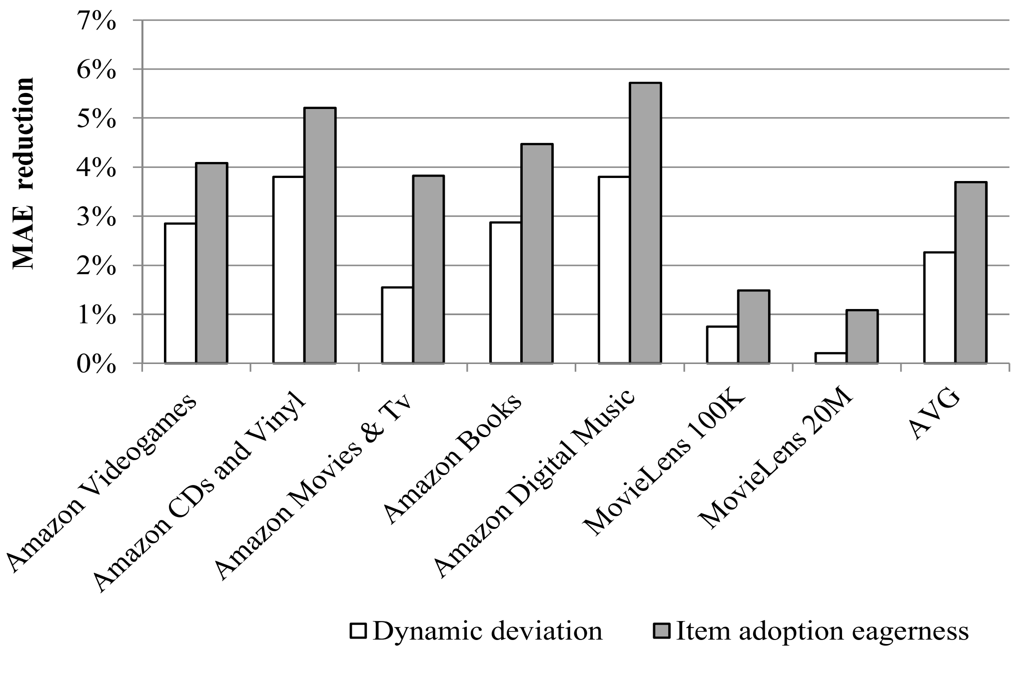

Figure 5: MAE reduction achieved by the proposed algorithm, in comparison to the CF variability algorithm, proposed in [3]

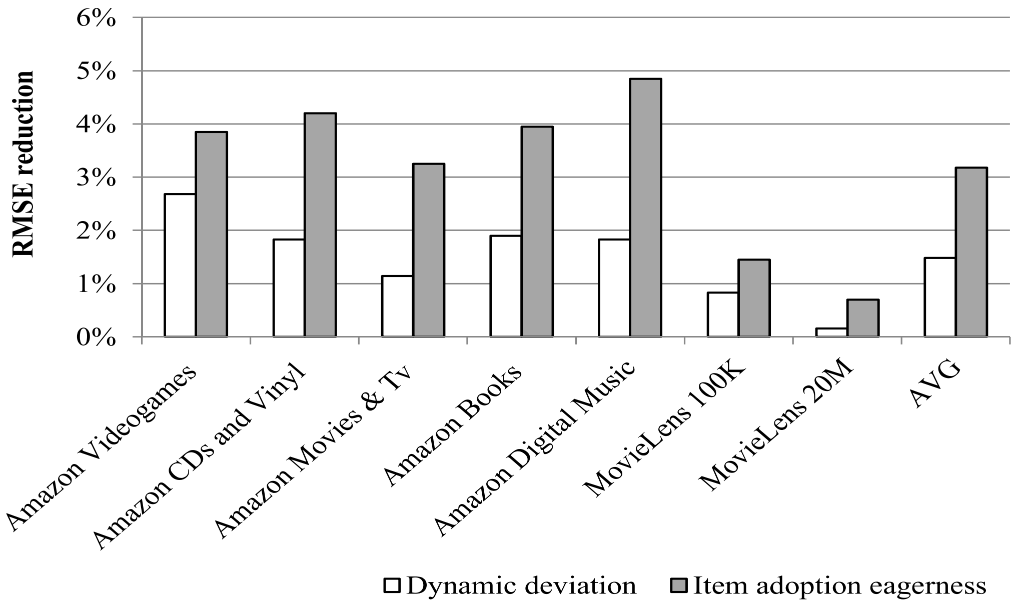

Figure 6: RMSE reduction achieved by the proposed algorithm, in comparison to the CF variability algorithm, proposed in [3]

Again, the proposed algorithm is shown to surpass the performance of the CF variability algorithm in all datasets tested, with its RMSE reduction being 114.6% higher than that achieved by the CF variability algorithm (3.18% against 1.48% in absolute figures). At individual dataset level, the performance edge of the proposed algorithm against the CF variability algorithm ranges from 44% to 317%.

Finally, we compare the performance of the proposed algorithm against the algorithm presented in [2], which is a state-of-the-art algorithm exploiting temporal, within-user history information, to achieve prediction error reduction in the context of CF-based rating predictions, and has also been shown to surpass the performance of other state-of-the art algorithms. The proposed algorithm achieves an average MAE improvement of 3.7% over all tested datasets, while the respective gains of the algorithm presented in [2] are 2.99%. While the relative difference is limited to 23.7%, it is stressed here that the algorithm presented in [2] requires and exploits additionally information from social networks regarding the influence levels among users, which are not always available. Additionally, the algorithm presented in [2] exhibits a coverage drop, which is considerable in the context of sparse datasets; on the other hand, the proposed algorithm fully maintains the coverage levels.

3. Conclusions

In this report we have presented the experimental findings from applying an algorithm that incorporates, in the rating prediction computation process, the aspect of the users' eagerness to adopt new items and technologies, in order to improve prediction accuracy. The results indicate that the above algorithm introduces considerable prediction accuracy gains.

4. References

[1]. M. Ekstrand, R. Riedl and J. Konstan (2011). Collaborative Filtering Recommender Systems. Foundations and Trends in Human-Computer Interaction, 4(2), 81-173.

[2]. D. Margaris and C. Vassilakis (2018). Exploiting Rating Abstention Intervals for Addressing Concept Drift in Social Network Recommender Systems. Informatics, 5(2), 21.

[3]. D. Margaris and C. Vassilakis (2018). Improving Collaborative Filtering's Rating Prediction Accuracy by Considering Users' Rating Variability. Proceedings of the 4th IEEE International Conference on Big Data Intelligence and Computing, 1022-1027.

[4]. Amazon (2019). Amazon product data, http://jmcauley.ucsd.edu/data/amazon/ links.html.

[5]. J.J. McAuley, C. Targett, Q. Shi and A. Van den Hengel (2015). Image-Based Recommendations on Styles and Substitutes. Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, 43-52.

[6]. MovieLens (2019). MovieLens datasets, http://grouplens.org/datasets/movielens/.

[7]. F. Harper and J. Konstan (2016). The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems, 5(4), 19.

[8]. J.B. Schafer, D. Frankowski, J. Herlocker and S. Sen (2007). Collaborative Filtering Recommender Systems. Lecture Notes in Computer Science, 4321, 291-324.

[9]. K.Yu, A. Schwaighofer, V. Tresp, X. Xu and H.P. Kriege (2004). Probabilistic Memory-Based Collaborative Filtering. IEEE Transactions on Knowledge Data Engineering, 16(1), 56-59.