Adapting to Natural Rating Acquisition with Combined Active Learning Strategies

Mehdi Elahi[1], Francesco Ricci[1], and Neil Rubens[2]

[1] Free University of Bozen-Bolzano, Bozen-Bolzano, Italy
mehdi.elahi@stud-inf.unibz.it, fricci@unibz.it
http://www.unibz.it

[2] University of Electro-Communications, Tokyo, Japan
neil@hrstc.org
http://www.uec.ac.jp

Abstract. The accuracy of collaborative-filtering recommender systems largely depends on the quantity and quality of the ratings added to the system over time. Active learning (AL) aims to improve the quality of ratings by selectively finding and soliciting the most informative ratings. However previous AL techniques have been evaluated assuming a rather artificial scenario: where AL is the only source of rating acquisition. However, users do frequently rate items on their own, without being prompted by the AL algorithms (natural acquisition). In this paper we show that different AL strategies work better under different conditions, and adding naturally acquired ratings changes these conditions and may result in a decreased effectiveness for some of them. While we are unable to control the naturally occurring changes in conditions, we should adaptively select the AL strategies which are well suited for the conditions at hand. We show that choosing AL strategies adaptively outperforms any of the individual AL strategies.

Keywords: Recommender systems, active learning, combined-strategies, rating elicitation.

1 Introduction and State of the Art

In this paper we focus on collaborative filtering (CF) Recommender Systems (RS) [5] that recommend items to a user based on the collective ratings of other users. The CF rating prediction accuracy does depend on: the characteristics of the prediction algorithm, and the number, the distribution, and the quality of the ratings known by the system. In general, the more informative about the user's preferences the collected ratings are, the higher is the recommendation accuracy.

In previous research, briefly described here, it has been therefore stressed that it is important to keep acquiring new and useful ratings from the users by adopting Active Learning strategies [9]. Different techniques have been proposed, some being non-personalized such as choosing popular items or items with diverse ratings (higher entropy of ratings), and some personalized such as using a decision tree to choose what to ask based on previous ratings of the users. These studies focussed especially on the cold start stage, i.e., where a new user is added to the system [10,8,11,4]. However, both the system and users do not remain in the cold start stage for long, as users start to interact with the system and the system starts to evolve. In realistic settings an active learning rating elicitation strategy can't be viewed as the only source, or tool, for collecting new ratings from the users. Users may freely and voluntary enter new ratings; yet all the previously mentioned elicitation strategies were evaluated in isolation from this natural ongoing system usage.

The need for evaluating the behavior of RS as it evolves is starting to receive its due attention. In particular, scholars are starting to evaluate system's performance (accuracy, diversity and robustness) from the temporal perspective where the users are rating items and the database is growing (although without considering the application of any active learning approaches) [1,6]. This is radically different from the typical evaluations where the rating dataset is decomposed into the training and test sets without considering the timestamps of the ratings. [1,6] have shown that in such scenarios the accuracy of the system might not improve even though ratings are added to the system.

Previous works have considered that ratings are acquired either actively during the cold start stage [10,8,11,4], or in a natural manner throughout the system's lifetime [1,6]. However, in practice, ratings may be acquired in both ways. In addition, recently [3] have shown that actively requesting and acquiring new ratings is useful for the overall system's performance at any stage of the system's evolution, and not just at the beginning (cold start). Motivated by the above, we propose a novel evaluation methodology, which we claim to be more realistic and indicative of the real performance of a rating elicitation strategy, i.e., combining AL with the natural acquisition of items and evaluating it throughout the evolution of RS. We show that especially in these settings, a single AL strategy cannot perform consistently well. To overcome this limitation, we propose active learning strategies that combine other AL strategies by (1) voting mechanism, or by (2) adaptively selecting seemingly best suited AL strategy. We believe that these results provide guidelines and conclusions that would help the deployment of AL rating elicitation in real RSs.

The rest of the paper is organized as follows. In section 2 we introduce the rating elicitation strategies that we have analyzed. In section 3 we present the simulation procedure that we designed to evaluate their effect on the system's recommendation performance (MAE and NDCG). The results of our experiments are shown in section 4, and in section 5 we summarize the outcome of this research.

2 Rating Elicitation Strategies

A rating dataset R is a $n \times m$ matrix of real values (ratings) with possible null entries. The variable r_{ui}, denotes the entry of the matrix in position (u, i), and contains the rating assigned by user u to item i, typically an integer between 1 and 5. r_{ui} could store a null value representing the fact that the system does not know the opinion of the user on that item.

A *rating elicitation strategy* S is a function $S(u, N, K, U_u) = L$ which returns a list of items $L = \{i_1, \ldots, i_M\}$ whose ratings should be elicited from the user u, where N is the maximum number of items that the strategy should return, K is the rating dataset of known ratings, i.e., the ratings (of all the users) that have been already acquired by the RS at a certain point. Finally, U_u is the set of items whose ratings have not yet been elicited from u, hence potentially interesting. The elicitation strategy enforces that $L \subset U_u$ and will not repeatedly ask a user to rate the same item; i.e. after the items in L are shown to a user they are removed from U_u.

Every elicitation strategy analyzes the dataset of known ratings K and scores the items in U_u. If the strategy can score at least N different items, then the N items with the highest score are returned. Otherwise a smaller number of items is returned. It is important to note that the user may have not experienced the items whose ratings are requested; in this case the system will not increase the number of known ratings. In practice, following a strategy may result in collecting a larger number of ratings, while following another one may results in fewer but more informative ratings. These two properties (rating quantity and quality) play a fundamental role in rating elicitation.

In a previous work we evaluated several individual strategies including: popularity, log(pop)$*$entropy, binary-prediction, highest-predicted, lowest-predicted, highest-lowest-predicted, voting, and random [3]. In this paper, since the observed behaviors of some strategies are similar, and to make our presentation clearer, we consider and illustrate just three individual strategies, which are actually representative of some others. We have chosen log(popularity)*entropy since it has been always indicated as an effective one [7], highest-predicted since it is a good representative for other prediction based strategies, and random. Moreover, we introduce here two novel strategies, voting and switching, which we call “combined”, since they aggregate and combine the previously mentioned individual strategies (or any other selection of strategies).

*log(popularity) * entropy:* the score for the item i is computed by multiplying the logarithm of the popularity of i, i.e., the number of known ratings for i, with the entropy of the ratings for i in K. This strategy tries to combine the effect of the popularity score, which is discussed above, with the heuristics that favors items with more diverse ratings (larger entropy) which provide more useful (discriminative) information about the user's preferences [2,7].

Highest Predicted: based on the ratings in K, predictions are computed for all the items in U_u and the scores are set equal to these predicted values. We use Matrix Factorization with gradient descent optimization to generate rating predictions [5]. The idea behind the highest-predicted strategy is that the best recommendations could also be more likely to have been experienced by the user and obtained ratings could also reveal important information about user's preferences. Moreover, this is the default strategy for RSs, i.e., enabling the user to rate the recommendations.

Random: the score for an item is a random integer. This is a baseline strategy, used for comparison.

Combined with Voting: the score for the item i is the number of votes given by the committee of the previously mentioned strategies: *log(pop)*entropy*, *highest predicted,* and *random.* Each of these strategies produces its top 100 candidates for rating elicitation, and then the items appearing more often in these lists are selected. This strategy obviously depends on the selected voting strategies.

Combined with Switching: every time this strategy is used, a certain percentage (40% in our experiments) of the users (exploration group) are randomly selected for choosing the best performing individual strategy and applying it on the remaining users (60%). Each individual strategy is applied to an equal number of random users in the exploration group: it selects items to be rated by these users, and acquires their ratings if they are present in a set of hold out ratings X, which represents the ratings that users could give but they have not provided yet to the system. How X is generated is described in the next section. Moreover, based on the ratings in K, a factor model is trained and its MAE and NDCG for these newly acquired ratings are computed. We also compute the probability for an individual strategy to acquire the ratings for the selected items (estimated as the ratio of the number of acquired ratings over the number of items requested to be rated). Finally, the score of each individual strategy is calculated by multiplying this probability either by the rating prediction error (MAE) on the acquired ratings, if MAE is the target metric to minimize, or by (1 - NDCG) in the other case. The strategy with the highest score is then selected. Hence, the combined switching strategy is selecting the individual strategy that was able to acquire from the exploration group the largest number of ratings, for items for which the system prediction is currently most erroneous (either for MAE or NDCG). Conjecturing that these are the most informative items, and that the selected strategy will have the same behavior on the remaining users (60% in our experiments), the winner strategy is then applied for eliciting ratings from the remaining users.

3 Evaluation Approach

We designed a procedure to simulate the evolution of the RS's performance by mixing the ratings acquired by an active learning strategy (individual or combined) with the ratings entered naturally by the users without being explicitly requested, just as it happens in actual settings. To accomplish this goal, we have used the larger version of the Movielens dataset (1,000,000 ratings) for which we considered only the ratings of users that were active and rated movies for at least 8 weeks (2 months).

We have split the available data into three matrices K, X and T. We inserted into K, the set of known ratings, those acquired by Movielens in the weeks 1-4 (first month) of the system usage (14,195 ratings). Then we randomly split the remaining ratings by inserting 70% of them into X (251,241) and 30% into T (111,867). X represents the ratings that are available to be elicited, while T are ratings that are never elicited and are used for testing the system's performance. We perform a simulated iteration every week, namely each day of the week (starting from the second week) an active learning strategy requests each user, who already has some non-null ratings in K, to rate 40 items. If these ratings are present in X, they are added to K. This step is repeated for 7 days (1 week). Then, all the ratings in the Movielens dataset that according to the timestamps were acquired in that week are also added to K. Finally, the system is trained using the ratings in K and then tested on the ratings in T that users actually entered during the following week (according to the timestamps). This procedure is repeated for $I = 48$ weeks (1 year). In order to precisely describe the evaluation procedure, we use the following notation, where n is the week index:

K_n: is the set of ratings known by the system at the end of the week n. These are the ratings that have been acquired up to week n. They are used to train the prediction model, compute the active learning rating elicitation strategies for week $n + 1$, and test the system's performance using the ratings contained in the test set of the next week $n + 1$, T_{n+1}.

T_{n+1}: is the set of ratings timestamped during the week $n+1$ that are used as a test set to measure the system's performance after the ratings in the previous weeks have been added to K_n.

AL_n: is the set of ratings elicited by a particular elicitation strategy, and is added to the known set (K_n) at week n. We note that these are ratings that are present in X but not in T. This is required for assuring that the active learning strategies are not modifying the test set and that the system's performance, under the application of the strategies, is consistently tested on the same set of ratings.

X_n: is the set of ratings in X, timestamped in week n that are not in the test set T_n. These ratings, together with the ratings in T_n, are all of the ratings acquired in Movielens during the week n, and therefore are considered to have been naturally provided by the (simulated) users without being asked by the system (natural acquisition). We note that it may happen that an elicitation strategy has already acquired some of these ratings, i.e., the intersection of AL_n and X_n may be not empty. In this case, only those not yet actively acquired are added to K_n. The testing of an active learning strategy S now proceeds in the following way:

- System initialization: week 1 to 4 (1 month)
  1 The entire set of ratings is partitioned randomly into the two matrices X, T.
  2 The non-null ratings in X_1 and T_1 are added to K_1 : $K_1 = X_1 \cup T_1$
  3 U_u, the unclear set of user u is initialized to all the items i with a null value k_{ui} in K_1.
  4 The rating prediction model is trained on K_1, and MAE, Precision, and NDCG are measured on T_2.
- For all the weeks n starting from $n = 5$
  5 Initialize K_n with all the ratings in K_{n-1}.
  6 For each user u with at least 1 rating in K_{n-1}:

- Using strategy S a set of items $L = S(u, N, K_{n-1}, U_u)$ is computed.
- The set L_e is created, containing only the items in L that have a non-null rating in X. The ratings for the items in L_e are added to AL_n
- Remove from U_u the items in L: $U_u = U_u \setminus L$.

7 Add to K_n the ratings timestamped in week n and those elicited by S: $K_n = AL_n \cup X_n \cup T_n$.

8 Train the factor model on K_n.

9 Compute MAE and NDCG on T_{n+1}.

4 Results

Figure 1 shows the evolution of the system MAE in the first 48 weeks under the application of the considered strategies. Here MAE is normalized with respect to the MAE of the baseline, i.e., when only the natural acquisition of the ratings is used: $(\frac{MAE_{Strategies}}{MAE_{Baseline}}) - 1$. Additionally, in figure 2 we plot the evolution of the MAE when the ratings are added only by the strategies themselves (i.e. at step 7 the ratings timestamped in week n are not added). Comparing these figures makes it clear that the natural acquisition of ratings can have a huge impact on the accuracy of the system. Without natural acquisition MAE fluctuates more, since the system's performance is only determined by the ratings acquired with active learning, which acquires fewer ratings. The traditional way of evaluating AL strategies in isolation is misleading, since ratings do get frequently added in a natural manner (besides being added by AL strategies), and do have a significant effect on the performance of AL strategies. Hence, evaluating active learning strategies in combination with the natural addition of ratings provides a more realistic and accurate evaluation.

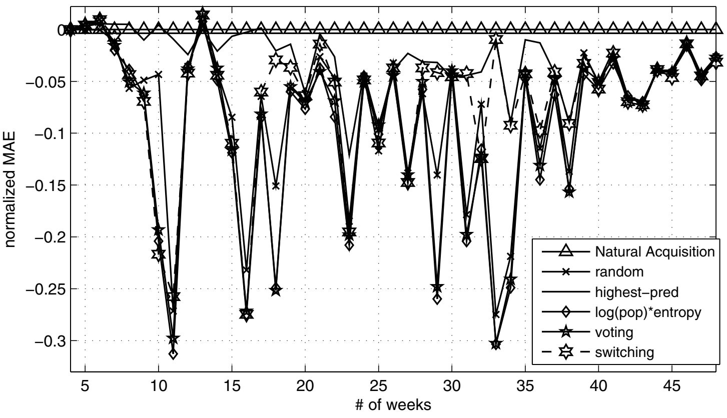

Fig. 1. Normalized system MAE under the effect of AL strategies and natural acquisition (48 weeks)

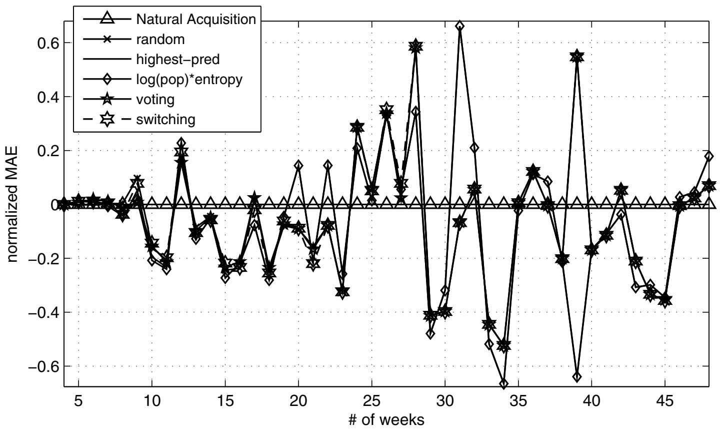

Fig. 2. Normalized system MAE under the effect of AL strategies without natural acquisition (48 weeks)

To better illustrate the MAE evolution, Figure 3 shows the system's performance, already presented in Figure 1, focusing only on the first 14 weeks (lower values are better). Here, it is clear that AL strategies can provide the potential benefit (most evident at week #11), but also that their efficiency may vary (e.g. week #11 vs. #13). It should be noted that there is a huge fluctuation of MAE, from week to week. This is caused by the fact that every week we test the system's performance on the next week's ratings. Hence, the difficulty of making good predictions and acquiring informative ratings may differ from week to week. The result shows that up to the seventh week, the performance of the strategies are similar. However, starting from week seven, the system MAE decreases except when the highest predicted strategy is used, which keeps it near to the baseline.

We have also evaluated the strategies with respect to Normalized Discounted Cumulative Gain (NDCG) that measures how close the ranking of the items predicted by the RS is to the optimal ranking. Figure 4 shows the NDCG of the considered strategies, when it is normalized by the NDCG of the natural acquisition baseline. In this case larger values are better. In the long run all the strategies can improve NDCG (over the baseline), however, their behavior varies over time. A possible explanation of the more stable behavior of NDCG, compared with MAE, is that NDCG is strongly influenced by a relatively small number of predicted top-rated items, and these are less likely to change week by week.

In order to better compare the strategies on a larger time window, in Table 1 we show the number of weeks that each strategy is the best or the second best among all the strategies. When the natural acquisition is considered, the three strategies that produce the lowest MAE are log(pop)*entropy, voting and switching. We can also observe that highest-predicted does not perform very

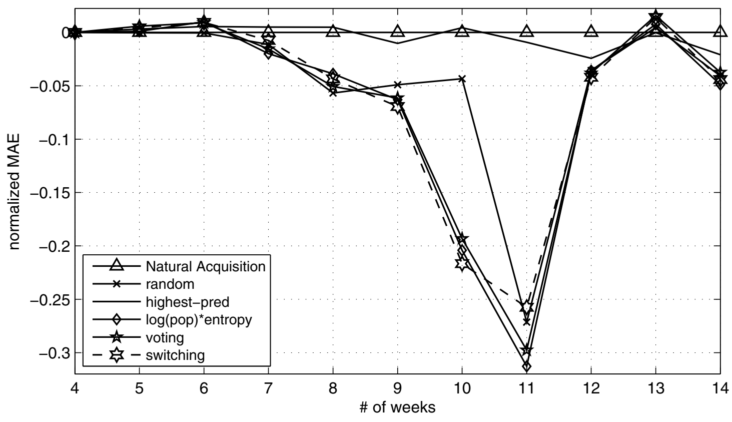

Fig. 3. Normalized system MAE under the effect of AL strategies and natural acquisition (14 weeks)

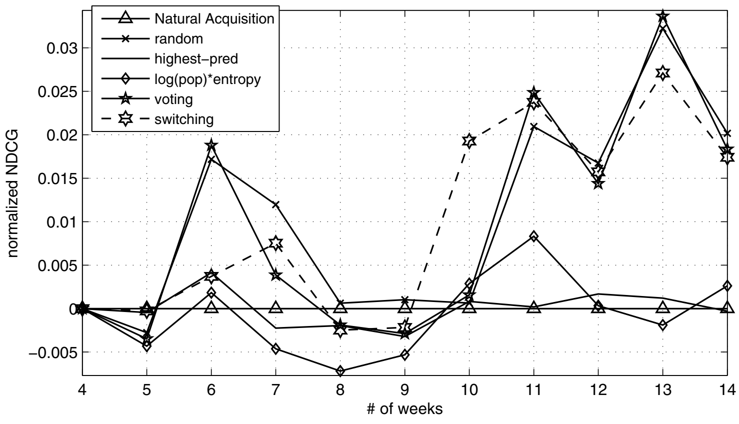

Fig. 4. Normalized system NDCG under the effect of AL strategies and natural acquisition (14 weeks)

differently from the baseline. The main reason is that this strategy is not acquiring additional ratings besides those already collected by the natural process, i.e., the user would rate these items on his own initiative. The other strategies are able to elicit more ratings, including those that the user will rate later on, i.e., in the successive weeks. When the natural acquisition is not considered, the best strategies are random, voting, switching and log(pop)*entropy. We stress again the different performance of the AL strategies when evaluated in isolation, and with the addition of naturally acquired ratings.

Considering NDCG, Table 1 shows that with natural acquisition the best strategies are voting, switching and log(pop)*entropy. However, without natural acquisition the best strategies are random, voting and switching. We should

Table 1. Number of weeks that a strategy performs as the best or second best (in 48 weeks)

	Without Nat. Acquisition		With Nat. Acquisition	
Strategy	**MAE**	**NDCG**	**MAE**	**NDCG**
Random	**34**	**42**	13	16
Highest predicted	20	28	8	10
log(popularity)*entropy	**22**	28	**29**	**20**
Voting	**26**	**42**	**18**	**23**
Switching (40% exploration)	**22**	**36**	**15**	**20**

mention that for several weeks, log(pop)*entropy performed worst among all the strategies which is quite different from results for MAE. Overall, considering both the MAE and NDCG metrics in the long run (48 weeks), switching and voting achieved remarkable results since both performed as good as the best strategies with respect to one of these metrics. For instance, as explained before, log(pop)*entropy can produce a very good MAE but not NDCG. While switching is comparable with log(pop)*entropy in terms of MAE as well as being equivalent to random in terms of NDCG.

5 Conclusions

In this work we have addressed the problem of selecting items to present to the users for acquiring their ratings; that is also defined as the ratings elicitation problem. We have proposed a more realistic active learning evaluation settings in which ratings are added not only by the AL strategies, but also by users that rate items on their own (without being prompted by the AL algorithms). Since AL strategies are no longer able to select all of the ratings, their behavior changes. For example, the default AL strategy (highest-predicted) becomes much less efficient, since many of the ratings that strategy tries to acquire are anyway added by the users. By evaluating the system's performance on a weekly basis, it becomes apparent that no single AL strategy performs consistently well due to changes in the rating set (caused by natural acquisition). We demonstrate that it is possible to adapt to the changes in the characteristics of the rating set, by proposing AL strategies that combine individual AL strategies, either by a voting mechanism, or by adaptively selecting them, from a pool of individual AL strategies, based on the estimation of how well each individual AL strategy is able to cope with the conditions at hand. For future work, we plan to fine-tune the switching strategy with a better method to minimize the exploration process. This can be done in a dynamic way so that in early weeks, the system concentrates more on exploration and later on exploitation. Moreover, we are implementing an online music recommender system where we will test the proposed methods, together with more recent strategies (e.g. [11,4]), in a live user experiment.

References

1. Burke, R.: Evaluating the dynamic properties of recommendation algorithms. In: Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys 2010, pp. 225–228. ACM, New York (2010)
2. Carenini, G., Smith, J., Poole, D.: Towards more conversational and collaborative recommender systems. In: Proceedings of the 2003 International Conference on Intelligent User Interfaces, Miami, FL, USA, January 12-15, pp. 12–18 (2003)
3. Elahi, M., Repsys, V., Ricci, F.: Rating Elicitation Strategies for Collaborative Filtering. In: Huemer, C., Setzer, T. (eds.) EC-Web 2011. LNBIP, vol. 85, pp. 160–171. Springer, Heidelberg (2011)
4. Golbandi, N., Koren, Y., Lempel, R.: Adaptive bootstrapping of recommender systems using decision trees. In: Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, pp. 595–604 (2011)
5. Koren, Y., Bell, R.: Advances in collaborative filtering. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P. (eds.) Recommender Systems Handbook, pp. 145–186. Springer (2011)
6. Lathia, N.K.: Evaluating Collaborative Filtering Over Time. PhD thesis, University College London (June 2010)
7. Rashid, A.M., Albert, I., Cosley, D., Lam, S.K., Mcnee, S.M., Konstan, J.A., Riedl, J.: Getting to know you: Learning new user preferences in recommender systems. In: Proceedings of the 2002 International Conference on Intelligent User Interfaces, IUI 2002, pp. 127–134. ACM Press (2002)
8. Rashid, A.M., Karypis, G., Riedl, J.: Learning preferences of new users in recommender systems: an information theoretic approach. SIGKDD Explor. Newsl. 10, 90–100 (2008)
9. Rubens, N., Kaplan, D., Sugiyama, M.: Active learning in recommender systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P. (eds.) Recommender Systems Handbook, pp. 735–767. Springer (2011)
10. Schein, A.I., Popescul, A., Ungar, L.H., Pennock, D.M.: Methods and metrics for cold-start recommendations. In: SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 253–260. ACM, New York (2002)
11. Zhou, K., Yang, S.-H., Zha, H.: Functional matrix factorizations for cold-start recommendation. In: Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, pp. 315–324 (2011)