# Active Learning in Collaborative Filtering Recommender Systems

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

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

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

**Abstract.** In Collaborative Filtering Recommender Systems user's preferences are expressed in terms of rated items and each rating allows to improve system prediction accuracy. However, not all of the ratings bring the same amount of information about the user's tastes. Active Learning aims at identifying rating data that better reflects users' preferences. Active learning *Strategies* are used to selectively choose the items to present to the user in order to acquire her ratings and ultimately improve the recommendation accuracy. In this survey article, we review recent active learning techniques for collaborative filtering along two dimensions: (a) whether the system requested ratings are personalised or not, and, (b) whether active learning is guided by one criterion (heuristic) or multiple criteria.

**Keywords:** recommender systems, cold start, active learning.

## 1 Introduction

This article concisely reviews the state-of-the-art in active learning for collaborative filtering recommender systems. *Active Learning* tackles the problem of obtaining high quality data that better represents the preferences of a user and largely improves the performance of the system. This is done by identifying, for each user, a set of items and requesting the user to rate them. The ultimate goal is to acquire ratings that will enable the system to generate effective recommendations. However, users are typically uninterested and reluctant to enter many ratings: this activity represents a cognitive cost for the user. This is why, it is important to carefully design a strategy to choose, the smallest in number, and the greatest in informativeness, of instances (ratings) to train the system. In other words, in collaborative filtering an active learning *Strategy* for rating elicitation is a precise procedure for selecting which items to present to the user for rating elicitation.

So far, several active learning strategies have been proposed and evaluated for collaborative filtering recommender systems. These strategies hold different

M. Hepp and Y. Hoffner (Eds.): EC-Web 2014, LNBIP 188, pp. 113–124, 2014.

features and implement various heuristics in selecting the useful items to be presented to the users to rate. Heuristics are strategies that instead of directly minimising the system prediction error try to improve other system properties that are indirectly influencing the error. For instance, acquiring more ratings for popular items (as the popularity-based strategy does) tends to reduce the system error but may also acquire too many high ratings that can erroneously bias the system towards high rating predictions [8,6].

The classification of these strategies with respect to descriptive and discriminative dimensions can give a useful overview and become a practical resource for practitioners and researchers in the field of recommender systems. For that aim, we have identified two important dimensions that have been addressed by several research works [17,18,11,19,12,10], and then, analyzed and classified the active learning strategies along these two dimensions, namely:

- **Personalisation:** that address the extent to which the personalisation is performed when selecting the list of candidate items for the users to rate. Hence, this dimension classifies the strategies into *Non-Personalzied*, and *Personalised*. Non-personalized strategies are those that ask all the users to rate the same list of items. Personalized strategies, on the other hand, ask different users to rate different items. Personalization is an important aspect in active learning since the users have different tastes, preferences and experiences, hence, the usefulness of the ratings for the same set of items could vary greatly from user to user. Moreover, selecting items to be rated with regards to preferences of each user, may provide a more pleasant experience for the user (e.g. by presenting them with items that they can actually rate), and at the same time may be more informative for the system.
- **Hybridization:** reflects whether the strategy takes into account a single heuristic (criterion) for selecting the items for the users to rate or combines several heuristics in order to create a more effective strategy. In this regard, the strategies can be classified into *Single-heuristic (or Individual)* and *Combined-heuristic (or Combined)* strategies. Single-heuristic strategies are those that implement a unique item selection rule and select items only based on that heuristic. Combined-heuristic strategies hybridize single-heuristic strategies by aggregating and combining a number of strategies and utilize multiple item selection rules in order to better estimate which items are more useful for improving the system performance and therefore should be presented to the user for rating.

We note that the main contributions of this article are: (1) novel dimensions for classification of active learning strategies in collaborative filtering, i.e., personalisation and hybridization in active learning, (2) comprehensive analysis and classification of more than 24 strategies (see Fig. 2) as well as the brief description of potential pros and cons of these strategies, (3) sub-classification of the strategies for an even better and more informative description and discrimination of them, e.g., static combined-heuristic or adaptive combined-heuristic strategies.

The remainder of the article is structured as follows: section 2 gives a summary on typical user-system interaction and an illustrative example of active learning in collaborative filtering; section 3 introduces several non-personalized active learning strategies that can also be either single-heuristic or combined-heuristics; section 4 presents an analogue analysis of personalized strategies; and finally, section 5 concludes this survey article.

## 2 Active Learning in Collaborative Filtering

The first research works in active learning for recommender systems have been motivated by the need to implement more effective sign up processes for the classical collaborative filtering systems [4]. In fact, these works assume the *Standard Interaction Model* [3] for user-system interaction in collaborative filtering, i.e., selecting and proposing to users a set of items to rate only during the sign up process, until the user rates a sufficient number of items. An example of the works dealing with this interaction model is [17] where the authors focus explicitly on the sign up process, i.e., when a new user starts using a collaborative filtering recommender system and must rate some items in order to provide the system with some initial information about her preferences.

A more recent alternative interaction model is the *Conversational and Collaborative* [3], i.e., the user is supposed to rate items during the sign-up process, but is also invited to rate additional items whenever she is motivated to provide more ratings. In [3] the authors propose a set of techniques to intelligently select items to rate when the user is particularly motivated to provide such information.

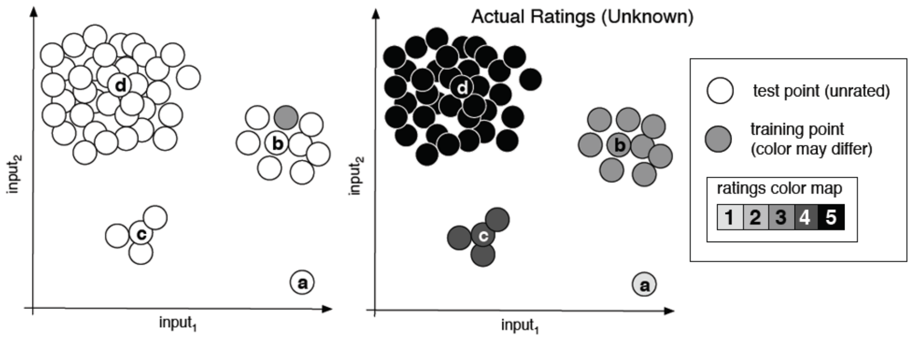


**Fig. 1.** Active learning, an illustrative example [19]

Figure 1 illustrates an example of active learning in recommender systems [19]. The left chart in the figure, represents the starting state, in which the system has requested the user to rate just a movie within the upper right group, say the Sci-Fi genre. The right chart shows the actual ratings of the user (color coded) but the system is not aware of these preferences. The chart is also showing four possibilities for selecting the next movie to be rated, i.e., movie *a, b, c* or *d*. If the system selects the movie $a$, say a very peculiar movie, it may not influence prediction accuracy, since, no any other movie is located nearby. However, if the movie $b$ is selected, it may let the system to make prediction for the movies within the same area. However, it has already some information for predicting movies within this area. If the movie $c$ is selected, the system is able to make new predictions, but only for the other three movies in this area, say Zombie movies. Ultimately, by selecting movie $d$, the system will be able to make predictions for a large number of movies that are located nearby, in the same area (say Comedy movies). Therefore, selecting movie $d$ is likely to be the ideal choice as it allows the system to improve the accuracy of the predictions for the largest number of movies [19].

# 3 Non-Personalised Active Learning

Simpler active learning strategies do not take into account users previously expressed preferences and request users to rate the same set of items. We refer to these strategies as non-personalized. In this case, the heuristic for the item selection process does not depend on the profiles of individual users. For example, a non-personalized strategy may select items based on their popularity. Note that popularity criterion is not affected by preferences of users who haven't yet experienced the item; hence all of the users would be presented with the same item to rate (regardless of their preferences). We have identified two sub-categories within this group: single and combined heuristic.

## 3.1 Single-Heuristic Strategies

**Uncertainty-Reduction.** The strategies within this group, favour items that have received controversial feedback from the users (i.e., more diverse ratings). Hence, the system is more uncertain about user's opinion on them. Asking the user to rate such items can bring useful (discriminative) information about the user's preferences.

*Variance* [20,19,9,10]. It considers the ratings' variance as the indication of the uncertainty of the system about the rating prediction of an item. This strategy asks the users to rate the items with the highest variance of the ratings in the dataset. Hence, it favours the items that have been rated diversely.

*Entropy of Ratings* [17,1,20,9,12]. Entropy measures the dispersion of the ratings of the users for an item [18]. Entropy of ratings is computed by using the relative frequency of each of the five possible rating values (1-5) [17]. A variant of this strategy, *Entropy0* [18,9,10], tries to solve a limitation of the entropy strategy, i.e., its tendency to select unpopular items. Entropy0 tackle this problem by considering the unknown ratings as a new rating value, equal to 0, and hence considering a rating scales between 0 to 5. In such a way, a high frequency of the 0 rating (i.e., many unknown ratings) tends to decrease Entropy0 [18].

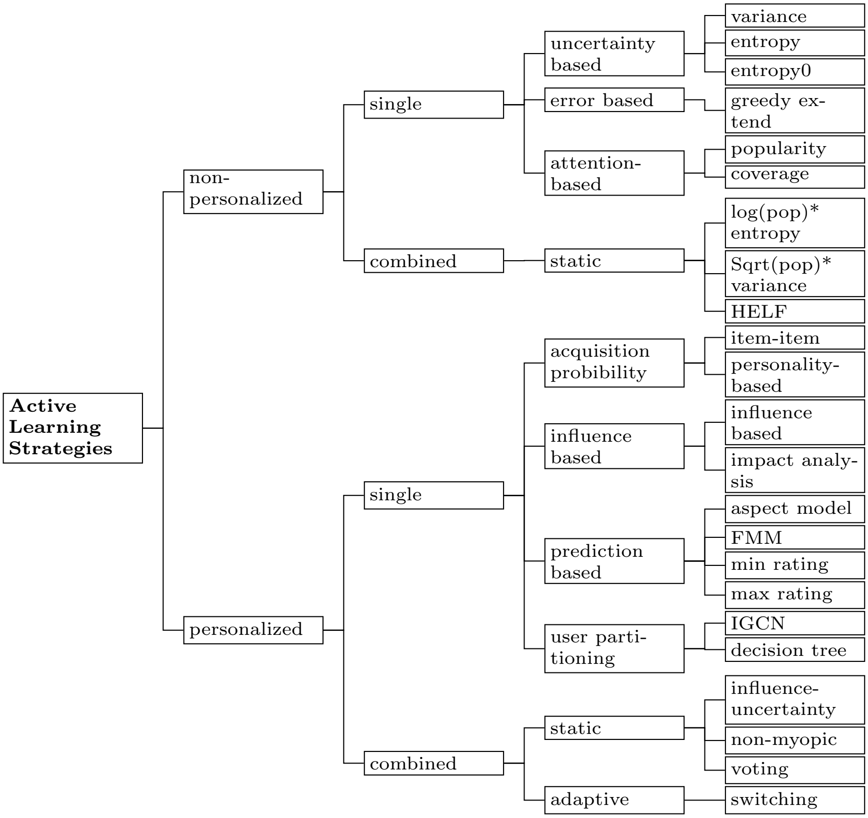


**Fig. 2.** Classification of active learning strategies in collaborative filtering

**Error Reduction.** In this group of strategies items are considered useful if they enhance the predictive accuracy of the system. While items with diverse rating may seem informative, they may not be necessarily the best items to request to rate if the rating prediction error is the primary concern. Indeed, error reduction is the main goal of any active learning strategy [19] since it is known to be strictly (negatively) correlated to the user satisfaction.

*Greedy Extend* [9,10]. Lets us denote with $A$ the prediction algorithm (e.g. Factor Model), and with $F(A)$ the performance measure of $A$, i.e., RMSE (estimated on the training set). The goal of greedy extend is to obtain an item list $L$, whose ratings should be elicited from the users, such that RMSE is minimised: $L = argmin_{L \subset I_u,\, |L|=N}\{F(A_L)\}$ where $A_L$ is the prediction algorithm $A$ trained only on the ratings given for the items in the list $L$, $I_u$ is the subset of the all items whose ratings are not known to the system, and may therefore be selected by the strategy to acquire ratings, and $N$ is the maximum number of items that the strategy should return.

**Attention-Based.** The attention-based strategies are simple and easy to implement, and they were used as the initial attempts to solve the cold start problem in collaborative filtering [17,3]. They are considered as baseline strategies in several analyses of active learning in recommender systems [16,12,9,15].

*Popularity* [17,3,20,18,9,12]. This strategy selects the items that have received the highest number of ratings. Hence, it is more likely that the user is able to rate these items and consequently the size of the rating dataset can be increased [3]. However, popular items are typically widely liked and already rated high by the users. Therefore, their ratings may bring a little information to the system.

*Coverage* [9]. This strategy selects the items that are highly co-rated by the users, and hence, eliciting their ratings may improve the prediction accuracy for the other items [9]. Here, $Coverage(i) = \sum_j n_{ij}$, where $n_{ij}$ is the number of users who co-rated the items $i$ and $j$. The heuristic used by this strategy is that collaborative filtering recognizes patterns across the items that are co-rated and hence correlated. These items are supposed to be the most useful ones in the sense that they help the system to better learn the users' preferences.

### 3.2 Combined-Heuristic Strategies

Combined-heuristic strategies hybridize single strategies in order to achieve a range of objectives, e.g., accuracy improvement, coverage widening, or user satisfaction. Such hybridization can be done in various ways: a certain number of strategies can vote for items and the most voted ones are selected; a certain number of strategies assign selection scores to each item and the total score is the product of these scores; a number of strategies select sets of items and the union or intersection of the selected items is chosen.

*Log(popularity)*Entropy* [17,3,20,16]. This strategy scores an item by computing the logarithm of the popularity multiplied by the entropy of the ratings given to the item. Hence, this strategy tries to combine the effect of the popularity score, which is discussed above, with the heuristics that favours items with more diverse ratings (larger entropy), which may provide more useful (discriminative) information about the user's preferences [3]. There is also a variation of this strategy proposed in [9] which uses $\sqrt{popularity}$ instead of $log(popularity)$, and *variance* instead of *entropy*.

*HELF* [18,9,10]. HELF stands for Harmonic mean of Entropy and Logarithm of Frequency. HELF aims at combining popularity with informativeness. As mentioned before (see uncertainty based strategies) the entropy strategy tends to select obscure items that are rarely rated. Hence, HELF attempts to solve this problem by selecting informative items that are also rated frequently by the users. Here entropy and popularity scores are combined to select the items that are familiar for a larger number of users and at the same time the users widely disagreed on them [18]: $HELF(i) = \frac{2 \times LF_i \times H(i)}{LF_i + H(i)}$ where, $LF_i$ denotes for the normalized logarithm of the rating frequency of item $i$, and $H(i)$ denotes for the normalized entropy of item $i$.

## 4 Personalised Active Learning

The second major category of active learning strategies for collaborative filtering comprises strategies that ask different users to rate different items (personalisation). The strategies in this group can also be either single or combined heuristic.

### 4.1 Single-Heuristic Strategies

**Acquisition Probability Based.** Strategies in this group focus on improving the performance of the system by maximizing the probability that the selected items are familiar to the user, hence are ratable. By proposing more familiar items to the users, these strategies tend to collect more ratings. For example if the user has not watched a movie, read a book, listened to a music track, or visited a touristic place, it is better for the system not to ask the user to rate them. In fact, it is important that the active learning strategy estimates and considers the probability that a user is able to rate an item before proposing it for rating.

*Item-Item Strategy* [17,18]. This strategy selects the items with the highest similarity to the previously rated items. Hence, after acquiring at least one rating from a user (e.g., by requesting her to rate in the sign-up process), the similarity values between her rated item(s) and the other unrated items are computed, and the items most similar to the rated ones are presented to the user to rate. Since this strategy does not care about the informativeness of the items and it can acquire the ratings of less useful items, hence may fail to improve the system in terms of prediction accuracy [17].

*Personality-Based Binary Prediction* [5,2]. This strategy first transforms the rating matrix to a matrix with the same number of rows and columns, by mapping null entries to 0 and not null entries to 1. Hence, the new matrix models only whether a user rated an item or not, regardless of its value. Then, this new matrix is used to train an extended version of the matrix factorization algorithm. This model can also profile users not only in terms of the binary ratings, but also using known user attributes, for instance, the Big Five personality traits scored on a scale from 1 to 5. Given a user $u$, an item $i$ and the set of user attributes $A(u)$, it predicts which items have been experienced by the user by using the following rule: $\hat{pr}_{ui} = \bar{i} + b_u + q_i^\top \cdot (p_u + \sum_{a \in A(u)} y_a)$ where $\hat{pr}_{ui}$ is the computed estimation of the probability that user $u$ has experienced item $i$, and, $p_u$, $q_i$ and $y_a$ are the latent factor vectors associated with the user $u$, the item $i$ and the user attribute $a$ respectively [5]. The model parameters are then learnt, as it is common in matrix factorization, by minimizing the associated regularized squared error function through stochastic gradient descent.

**Influence Based.** Eliciting the ratings for the items that the system is more uncertain about the users' evaluation, can be beneficial for the system to better predict the ratings of these particular items. However, that will not necessarily end up with the reduction of the uncertainty of the rating prediction for other items. Influence based strategies select items, in the attempt to minimize the rating prediction uncertainty for all the items.

*Influence Based* [20]. This strategy estimates the influence of item's ratings on the rating prediction of other items and selects the items with the largest influence. First the rating prediction $\hat{r}_{u,i}$ for user $u$ and an unrated item $i$, is computed. Then, the rating prediction $\hat{r}_{u,i}$ is decreased by 1 unit: $\hat{r}'_{u,i} = \hat{r}_{u,i} - 1$. Finally two prediction models are generated, one adding $\hat{r}_{u,i}$ and another adding $\hat{r}'_{u,i}$ to the traing set, and the absolute value of the differences of theirs predictions for the ratings of all the items different from $i$ are computed. The influence of $i$ is estimated by summing up all these differences. Finally, the items with the highest influence are selected for active learning [20].

*Impact Analysis* [16]. This strategy selects the items whose ratings have the highest impact on the prediction of the other ratings. In order to get the gist of this strategy, let us consider a graph-based representation of the rating dataset, where users and items are nodes of the graph. A rating can be seen as a link between a user and an item. In order to make a recommendation for a user using a user-based nearest neighbour model there should be at least a four-node path being created. A four-node path between the users $u_1$, $u_2$ and items $p_1$, $p_2$ is found when the user $u_1$ has rated item $p_1$ and $p_2$, and user $u_2$ has co-rated $p_2$. In this case, the nearest neighbour model can generate a prediction of the rating of $u_2$ for $p_1$. Hence, the more four-node paths are created the better the prediction may become. For that reason, this strategy attempts to discover and elicit the ratings of items that produces more new four-node paths.

**Prediction Based.** These are strategies that use a rating prediction model in order to identify which items are better to propose to the users to rate. Items are ranked according to their predicted ratings and the top items are selected for rating elicitation. The prediction models used by these strategies may differ. Hence, we can partition these strategies according to the prediction model they use. One of the advantages of these strategies is that they are typically less bothersome to the user since they request to rate items that are predicted as relevant for the user. Hence, the user may even enjoy seeing and rating them. Moreover, since they are relevant for the user, they are also likely to have been experienced by her.

*Aspect Model* [13,11,19,14]. This strategy determines a latent space model for each user, i.e., models the user as a mixture of multiple interests (aspects). Every user $u \in U$ has a probabilistic membership in multiple aspects $z \in Z$, and users in the same group are assumed to have similar rating patterns. The probability that the user $u$ rates $r$ the item $i$ is estimated by: $p(r|u,i) = \sum_{z \in Z} p(r|z,i)p(z|u)$ where $p(z|u)$ models how likely the user $u$ will belong to the group $z$ and $p(r|z,i)$ models how likely the users in group $z$ will rate $r$ the item $i$. An extension of this strategy uses *Flexible Mixture Model (FMM)* [13,11] and determines two sets of latent aspects $z_u$ and $z_i$: every user $u$ is considered to be a mixture of multiple interests (user aspects) but also every item $i$ is considered to be a mixture of patterns (item aspects).

*MinRating* [15]. This strategy selects, for the user to rate, the items with the smallest predicted ratings. It is argued that when a new user is registered, since the system has no or few ratings of that user, the model parameters computed for this user can be inaccurate and considerably different from the optimal parameters. Therefore, it could be better to choose the items whose ratings are more erroneously predicted. On the other hand, since most of the ratings given by the users are large (users tend to rate the items they like) the prediction error for the items with low predicted ratings is expected to be high. Hence, eliciting low ratings should reveal large prediction errors and may impose significant changes into the model parameters.

*MinNorm* [15]. This strategy uses a prediction model (i.e., matrix factorization) to compute the latent factors of each item and then selects the items whose latent factor vectors have the smallest Euclidean norm. The reason is that when active learning has acquired a number of ratings, in matrix factorization, it could be better to stabilise the prediction model and avoid large changes in the latent factors. This is because the system has already achieved a certain level of accuracy that may be better to keep it. In order to do that, the change of the factors (gradient) should be minimized. While the system has less control on the prediction error, minimising the item factors may result in the minimum gradient and more stable prediction model.

**User Partitioning.** The heuristic guiding the strategies within this group is to first partition the existing users into a number of clusters, where the users of each cluster posses similar tastes and affinities. Then it selects the items whose rating will better reveal to which cluster the user belongs.

*Information Gain through Clustered Neighbours (IGCN)* [18]. This strategy constructs a decision tree where each leaf node represents a cluster of users and each internal node stores a test for a specific item that is proposed to a user to rate. Users are clustered according to their similarity values that are measured using a neighbourhood based collaborative filtering approach. Starting from the root node, a new user is proposed to rate a sequence of items, according to the built decision tree, until she reaches one of the leaf nodes. Hence, the most informative items, which belong to the tree, are those whose ratings will enable to better classify the user in her representative cluster.

*Decision Tree* [10,21]. Also this strategy uses a decision tree whose nodes, either internal or leaf, represents groups of users. Given an internal node, a candidate partitioning movie, divides the users into three groups based on the ratings of the users: *Lovers* (who rated the item high), *Haters* (who rated the item low), and *Unknowns* (who did not rate the item). When the tree is built, for each of these groups, the rating predictions of their unrated items are estimated. The estimated RMSE is computed as the squared root of the deviation of the predicted ratings from the true ratings. Finally, the total prediction error is computed by summing up the $RMSE$ in the three groups and the movie with minimal total error is selected for active learning. This process is iterated within each of the three previously built groups, to select the next movies and corresponding users partitions.

### 4.2 Combined-Heuristic Strategies

Personalized Combined-heuristic strategies hybridise personalized single-heuristic strategies by combining them in order to leverage the advantages that each method provides.

*Influence and Uncertainty Based* [20]. This strategy combines influence-based (see section 4.1) and variance strategy (see section 3.1). It selects items based on the influence that the rating of an item may have on the rating prediction of the other items and the variance of the item's ratings (as a measure of system prediction uncertainty): $argmax_i\{Var(i)\ I(i)\}$ where $I(i)$ is the influence of item $i$ (see section 4.1).

*Non-Myopic* [15]. This strategy combines two prediction based strategies: MinRating and MinNorm. At the beginning, when the first requests are made by the system, the items are selected mainly by MinRating strategy, which is supposed to work better in the early stages of the system, when the users have not rated many items (see the description of MinRating strategy). As more requests are made by the system, and more ratings are elicited, the system tends to use more the MinNorm strategy since MinNorm is supposed to work better in the late stage of the system evolution, e.g., when the users have already rated many items (see the description of MinNorm) [15].

*Voting* [8,6]. This strategy scores an item $i$ with the number of votes given by a committee of strategies. Each strategy produces its top candidate items for rating elicitation (strategy votes) and then the items appearing more often in these lists are selected. We stress that the voting strategy depends on the selected voting strategies. For example, including random strategy may impose an exploratory behaviour that could improve the system coverage [8].

*Combined with Switching* [7]. Every time this strategy is applied, a certain percentage of the users (called exploration group) are randomly selected for choosing the best performing individual strategy, then this winning strategy is applied on the remaining users. Each individual strategy is tested 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 available). Based on the ratings acquired by the system, a factor model is trained and its MAE and NDCG for these newly acquired ratings are computed. Then the observed probability for each individual strategy to acquire ratings for the selected items is also computed by estimating 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 is able to acquire from the exploration group the largest number of ratings for items whose system rating prediction is currently most erroneous [7].

## 5 Conclusion

In this article we provided a concise review of the state-of-the-art on active learning in collaborative filtering recommender systems. We have performed a comprehensive analysis and classified a wide range of active learning techniques, called *Strategies*, along the two dimensions: how personalised these techniques are, and how many different item selection criteria (heuristics) are considered by these strategies in their rating elicitation process.

It is worth noting that active learning for collaborative filtering is a multi disciplinary field, overlapping with a broad range of topics such as machine learning, data mining, information retrieval and filtering, recommender systems, human computer interaction and cognitive science. Such a broad range of disciplines, makes it difficult to cover all the works related to them. However, in this paper, we have conducted a comprehensive analysis of the literature by covering almost the main research works in the field. To our knowledge no paper before has performed such an analysis.

However, there are a number of interesting topics that have not been addressed so far and can be considered for future works. Firstly, it is important to survey works that have been done in active learning for other types of recommender systems, such as content-based and context-aware. Secondly, it is important to analyze active learning techniques based on their applicability to specific application domains. Indeed, the majority of the works in this field have tested active learning techniques in the movie domain (in offline settings). Finally, we must stress the importance of conducting more live user studies where active learning benefit can be better assessed.

## References

1. Boutilier, C., Zemel, R.S., Marlin, B.: Active collaborative filtering. In: Proceedings of the Nineteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI 2003), Acapulco (2003)
2. Braunhofer, M., Elahi, M., Ge, M., Ricci, F.: Context dependent preference acquisition with personality-based active learning in mobile recommender systems. In: Zaphiris, P., Ioannou, A. (eds.) LCT 2014, Part II. LNCS, vol. 8524, pp. 105–116. Springer, Heidelberg (2014)
3. Carenini, G., Smith, J., Poole, D.: Towards more conversational and collaborative recommender systems. In: Proceedings of the 8th International Conference on Intelligent User Interfaces, IUI 2003, pp. 12–18. ACM, New York (2003)
4. Desrosiers, C., Karypis, G.: A comprehensive survey of neighborhood-based recommendation methods. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 107–144. Springer (2011)
5. Elahi, M., Braunhofer, M., Ricci, F., Tkalcic, M.: Personality-based active learning for collaborative filtering recommender systems. In: Baldoni, M., Baroglio, C., Boella, G., Micalizio, R. (eds.) AI*IA 2013. LNCS, vol. 8249, pp. 360–371. Springer, Heidelberg (2013)
6. 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)

7. Elahi, M., Ricci, F., Rubens, N.: Adapting to natural rating acquisition with combined active learning strategies. In: Chen, L., Felfernig, A., Liu, J., Raś, Z.W. (eds.) ISMIS 2012. LNCS, vol. 7661, pp. 254–263. Springer, Heidelberg (2012)
8. Elahi, M., Ricci, F., Rubens, N.: Active learning strategies for rating elicitation in collaborative filtering: a system-wide perspective. ACM Transactions on Intelligent Systems and Technology 5(1) (2014)
9. Golbandi, N., Koren, Y., Lempel, R.: On bootstrapping recommender systems. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM 2010, pp. 1805–1808. ACM, New York (2010)
10. Golbandi, N., Koren, Y., Lempel, R.: Adaptive bootstrapping of recommender systems using decision trees. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, WSDM 2011, pp. 595–604. ACM, New York (2011)
11. Harpale, A.S., Yang, Y.: Personalized active learning for collaborative filtering. In: SIGIR 2008: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 91–98. ACM, New York (2008)
12. He, L., Liu, N.N., Yang, Q.: Active dual collaborative filtering with both item and attribute feedback. In: AAAI (2011)
13. Jin, R., Si, L.: A Bayesian approach toward active learning for collaborative filtering. In: Proceedings of the 20th Conference in Uncertainty in Artificial Intelligence, UAI 2004, Banff, Canada, July 7-11, pp. 278–285 (2004)
14. Karimi, R., Freudenthaler, C., Nanopoulos, A., Schmidt-Thieme, L.: Active learning for aspect model in recommender systems. In: CIDM, pp. 162–167. IEEE (2011)
15. Karimi, R., Freudenthaler, C., Nanopoulos, A., Schmidt-Thieme, L.: Non-myopic active learning for recommender systems based on matrix factorization. In: IRI, pp. 299–303. IEEE Systems, Man, and Cybernetics Society (2011)
16. Mello, C.E., Aufaure, M.-A., Zimbrao, G.: Active learning driven by rating impact analysis. In: Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys 2010, pp. 341–344. ACM, New York (2010)
17. 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)
18. 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)
19. 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)
20. Rubens, N., Sugiyama, M.: Influence-based collaborative active learning. In: Proceedings of the 2007 ACM Conference on Recommender Systems, RecSys 2007, pp. 145–148. ACM, New York (2007)
21. Zhou, K., Yang, S.-H., Zha, H.: Functional matrix factorizations for cold-start recommendation. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, pp. 315–324. ACM, New York (2011)