Active Dual Collaborative Filtering with both Item and Attribute Feedback

Luheng He and **Nathan N. Liu** and **Qiang Yang**
Hong Kong University of Science and Technology,
Clear Water Bay, Kowloon, Hong Kong
{luhenghe,nliu,qyang}@cse.ust.hk

Abstract

The new user problem (aka user cold start) is very common in online recommender systems. Active collaborative filtering (active CF) tries to solve this problem by intelligently soliciting user feedback in order to build an initial user profile with minimal costs. Existing methods only query the user for feedback on items, while users can have preferences over items as well as certain item attributes. In this paper, we extend active CF via user feedback on both items and attributes. For example, when making movie recommendations, the system can ask users for not only their favorite movies, but also attributes such as genres, actors, etc. We design a unified active CF framework for incorporating both item and attribute feedback based on the random walk model. We test the active CF algorithm on real-world movie recommendation data sets to demonstrate that appropriately querying for both item and feature feedback can significantly reduce the overall user effort measured in terms of number of queries. We show that we can achieve much better recommendation quality as compared to traditional active CF methods that support only item feedback.

Introduction

Collaborative filtering based recommender systems predict a user's preference based on what he liked before (i.e. user feedback) and what other similar-minded users had liked. Albeit effective, collaborative filtering (CF) inevitably suffer from the user cold-start problem, where no existing feedback is available for a new user. In order to have a usable profile for a new user, most recommender systems solicit some initial user feedback during registration. Most often, this is achieved by either asking the user to voluntarily enter a set of favorite items initially, or asking him a series of questions to answer. For example, the MovieLens Web site requires a user to provide a list of at least 15 favorite movies during registration, which can be very tedious and unpleasant. As a result, it would be highly desirable if the user feedback could be solicited in a manner which involved a minimal amount of user effort in a natural way.

Active collaborative filtering (active CF) provides a remedy to streamline the preference elicitation process by trying to identify the most *informative* questions to query the user in an attempt to maximize the amount of benefit gained from these questions. Traditional active CF methods have focused on soliciting *item feedback* by asking a user for his preferences over selected informative items; e.g., a movie recommender system may ask a user "How much you like the movie Lord of Rings?" However, users' interest may be better expressed in alternative forms of feedback. Consider, for example, in the domain of movie recommendation, where one can obtain item feedback by asking a user if he favors particular movies (e.g.: Lord of the Ring). Alternatively, the users can be asked to provide *attribute feedback*, such as whether he favors particular genres (e.g.: Adventure) or directors (e.g.: "Peter Jackson"). Attribute feedback an complement item feedback effectively to more precisely profile a user's interests. For example, when we only know that a user likes the popular movie "Lord of the Rings", we still could not determine if he is more interested in the movie's genre "fantasy" or in its director "Peter Jackson". However, if he also tell us "Peter Jackson" is his favorite director, then the system can more confidently infer that he would be more interested in other Peter Jackson movies such as "King Kong", rather than other "fantasy" movies such as "Harry Porter".

Attribute feedback can be adopted more broadly in various types of recommender systems where the users tend to be familiar with the meta data associated with the recommended items. For music services where the items to be recommended are songs, attribute feedback may be expressed in the form of favorite singers, musical styles, record labels, etc. For news and other information services, users may be interested in content features such as keywords or non-content features like publishers, etc. In general, attribute feedback is a novel form of user inputs that can nicely complement item feedback for profiling user interests. In text classification, several recent works have demonstrated that acquiring word labels can significantly reduce the number of document labels required to build accurate text classifiers (Druck, Settles, and McCallum 2009; Liu et al. 2004). However, to the best of our knowledge, little attention has been paid to profiling new users in collaborative filtering based recommender systems by actively acquiring attribute feedbacks.

In this paper, we explore the problem of *Active Dual Collaborative Filtering*, which concerns how to best query a

new user on both items and attribute feedbacks simultaneously. Our goal is to build a most effective recommendation model with a minimal amount of effort demanded from the user. We propose a unified framework for active dual collaborative filtering based on random walk with restart model (RWR). We design various query selection and fusion schemes for effectively acquiring both item and attribute feedbacks from a user. Experimental results on a movie recommendation data set demonstrate that the proposed algorithm significantly improves the recommendation quality over traditional item feedback based active collaborative filtering methods.

Active Dual Collaborative Filtering

In active dual collaborative filtering, the input data consists of three types of objects: users, items(e.g.: movies) and attributes(e.g.: genres). The data also involve three types of relations among the objects: item feedbacks, attribute feedbacks and item-attribute associations. The item and attribute feedback tell us which users like which items and attributes, respectively, while item-attribute associations tell us which attributes are present in which items. Given a new user, our goal is to interactively probe him for what items and attributes he likes so that we can recommend items to him based on the collected item and attribute feedback.

In the following, we first describe a tripartite graph based representation for encoding the heterogeneous input data and a Markov random walk model on this tripartite graph for making personalized recommendations. We then propose several ways to measure the "informativeness" of items and attributes based on the Markov random-walk model. Finally, we discuss how to effectively combine the two types of queries when probing a user during active learning.

Random Walk Model

Let $\mathcal{U} = \{u_1, u_2, ..., u_m\}$ denote the set of m users, $\mathcal{I} = \{i_1, i_2, ..., i_n\}$ denote the set of n items and $\mathcal{A} = \{a_1, a_2, ..., a_k\}$ denote the set of k attributes. We construct an undirected tripartite graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\} = \{\mathcal{U} \cup \mathcal{I} \cup \mathcal{A}, \mathcal{E}_{UI} \cup \mathcal{E}_{UA} \cup \mathcal{E}_{IA}\}$ with three types of bipartite edges derived from item feedback, attribute feedback and item-attribute associations respectively:

- **User-Item Edges ($\mathcal{E}_{UI}$):** From the item feedback, we create two edges (u_i, i_j) and (i_j, u_i) for a pair of user u_i and item i_j if u_i likes i_j. Let $\mathcal{I}(u_i)$ and $\mathcal{U}(i_j)$ denote the set of items liked by u_i and and set of users who like i_j respectively.
- **User-Attribute Edges ($\mathcal{E}_{UA}$):** From the attribute feedback, we create two edges (u_i, a_k) and (a_k, u_i) for a pair of user u_i and attribute a_k if u_i likes a_k. Let $\mathcal{A}(u_i)$ and $\mathcal{U}(a_k)$ denote the set of attributes liked by u_i and and set of users who like a_k respectively.
- **Item-Attribute Edges ($\mathcal{E}_{IA}$):** From the item attribute associations, we create two edges (i_j, a_k) and (a_k, i_j) for every pair of item i_j and attribute a_k if a_k is present in i_j. Let $\mathcal{A}(i_j)$ and $\mathcal{I}(a_k)$ denote the set of attributes present in i_j and and set of items which contain a_k respectively.

Furthermore, we assume the three types of edges are associated with weights $\omega_{UI}, \omega_{UA}, \omega_{IA}$, respectively, which control the relative importance of the three types of information. Given the graph structure and the edge weights, we can define transition probabilities by normalizing the edge weights out of each node:

$$P(v|u) = \begin{cases} \frac{\omega_{u,v}}{\sum_{v' \in \mathcal{V}(u)} \omega_{u,v'}} & \text{if } (u,v) \in \mathcal{E} \\ 0 & \text{if } (u,v) \notin \mathcal{E} \end{cases}$$

where $\mathcal{V}(u)$ denotes the set of nodes connected to u in the tripartite graph. All the transition probabilities can be organized as a $m+n+k$ by $m+n+k$ matrix $\mathbf{P}$ whose i,j-th entry corresponds to $P(v_i|v_j)$. The matrix $\mathbf{P}$ is column stochastic, so that the all columns sum to 1.

Personalized Recommendation via Random Walk with Restart

Given a few items and attributes that are relevant to a target user, we can determine the relevance of the remaining nodes in the graph based on the probabilities of reaching these nodes after performing t steps of random walk with restart from the known relevant nodes. At each step of random walk with restart (RWR), with probability $1-\alpha$, the random surfer would choose to follow edges out of the current node according to the transition probabilities defined in $\mathbf{P}$. With probability α, the random surfer would teleport to a node uniformly chosen from a set of *start nodes*, which corresponds to the set of items and attributes known to be relevant to the target user. Restart allows the random walk algorithm to stay in place and reinforces the importance of known relevant nodes by slowing diffusion to far away nodes.

Let $\mathcal{V}_0$ denote the set of start nodes that are known to be relevant to the target user. We define an initial distribution $P_0(\cdot|\mathcal{V}_0)$ as follows:

$$P_0(v_i|\mathcal{V}_0) = \begin{cases} \frac{1}{|\mathcal{V}_0|} & \text{if } v_i \in \mathcal{V}_0 \\ 0 & \text{if } v_i \notin \mathcal{V}_0 \end{cases} \quad (1)$$

Then, to obtain $P_t(v_i|\mathcal{V}_0)$, the probabilities of arriving at node v_i after t steps, we can repeatedly apply the following equations, which compute $P_t(\cdot|\mathcal{V}_0)$ from $P_{t-1}(\cdot|\mathcal{V}_0)$:

$$P_t(v_i|\mathcal{V}_0) = \alpha \cdot P_0(v_i|\mathcal{V}_0) + (1-\alpha) \cdot \sum_j P(v_i|v_j) P_{t-1}(v_j|\mathcal{V}_0) \quad (2)$$

Conducting RWR on the tripartite graph formed by users, items and attributes provide a natural way to diffuse relevance from the set $\mathcal{V}_0$ along the graph edges. By following the user-item and user-attribute edges during the random walk, the model is able to identify relevant items based on the preferences of other similar users, whereas if we follow item-attribute links, we can identify new relevant items with similar attributes. RWR fuses all these heterogeneous information to determine the relevance of a node based on the global structure of the tripartite graph rather than considering local structure only (e.g., comparing items only based on observed ratings or attributes).

Item and Attribute Query Selection Strategies

Active dual collaborative filtering is an iterative, human-in-the-loop learning paradigm. In each iteration, the recommender picks a set of items and/or attributes to query the user, such that knowing if the user likes these items and/or attributes would provide the maximum benefit to the recommender. Once additional user feedback are obtained, the learner updates its underlying model, which can then be used to select queries in the next iteration.

Various general query-selection principles, such as model entropy (Jin and Si 2004) and model change (Roy and Mccallum 2001), have been proposed in the past for classification and regression models. It is not clear how such principles can be applied to collaborative filtering models for the selection of both item and attributes to query. In this section, we design several query selection strategies based on the random walk model proposed in the previous section. The proposed random walk framework provides a unified representation of both items and attributes as graph nodes. A general strategy to characterize the value of node labels can then be easily applied to select both item and attribute queries.

Model Entropy Minimization One strategy is to select queries whose answers can minimize the expected entropy of the user model (Jin and Si 2004), which is equivalent to maximizing the confidence in the predicted relevance. In the random walk with restart model, a very high entropy of the probability distribution $P_t(\cdot|\mathcal{V}_0)$ defined over the graph nodes would indicate that the current RWR model can be equally likely to go anywhere on the graph. In contrast, a low entropy of this distribution would imply that the current RWR model can focus on a particular subset of nodes. This leads to the following *model entropy minimization* based query selection criterion:

$$v^* = \arg\min_{v \in \mathcal{I} \cup \mathcal{A}} \sum_{u \in \mathcal{U}} P_t(u|\mathcal{V}_0 \cup \{v\}) \log(P_t(u|\mathcal{V}_0 \cup \{v\})) \tag{3}$$

where $P_t(\cdot|\mathcal{V}_0 \cup \{v\})$ denotes the distribution over nodes resulted from a t step RWR if v is added as an additional start nodes. This lets us select the node such that if it is judged as relevant by the user, the distribution over the other nodes resulted from the RWR would have the minimal entropy.

A major computational bottleneck in computing model entropy lies in obtaining the probability distribution $P_t(\cdot|\mathcal{V}_0 \cup \{v\})$, which involves running power iterations based on equation 2 for every candidate node v. Given the large number of items and attributes that are potential queries, it would be prohibitive to compute this for each of them. Fortunately, in active learning, one is often only interested in finding a set of top k queries with the highest scores. Moreover, a query will only become useful if it is judged relevant by the user. These two conditions allow us to approximately find the top k queries by focusing on the set of items and attributes that are most likely to be relevant. More specifically, we try to approximately find the top k queries by first rank the candidate queries based on the predicted relevance (i.e. $P_t(\cdot|\mathcal{V}_0)$), and then re-rank only the top $d \times k$ most relevant queries in order to obtain the top k queries, where d controls the re-rank range. Empirically we find that setting $d = 3$ could lead to satisfactory performances.

Model Change Maximization Another general active learning principle is to select the queries that would influence the model the most, once its label is obtained. For parametric models for where the gradients with respect to the model parameters can be easily computed, query selection can be done by maximizing the "expected gradient length" (Settles and Craven 2008). However, the RWR model is nonparametric, and thus it is not possible to quantify the potential influence on an example in terms of gradients with respect to the model parameters. As a result, we propose to measure the model change indirectly by comparing the predicted relevance over the graph nodes before and after acquiring the query's label. In particular, for a candidate query node v, we can quantity its impact given the original distribution $P_t(\cdot|\mathcal{V}_0)$ and the updated distribution $P_t(\cdot|\mathcal{V}_0 \cup \{v\})$,

$$v^* = \arg\max_{v \in \mathcal{I} \cup \mathcal{A}} \sum_{u \in \mathcal{U}} (P_t(u|\mathcal{V}_0 \cup \{v\}) - P_t(u|\mathcal{V}_0))^2 \tag{4}$$

This criteria effectively identifies those nodes that are most informative in the sense that, after incorporating the feedback on these nodes, the model's predicted preferences on the remaining items would be significantly different. The extent of such changes in terms of L2-distance effectively quantifies the amount of novel knowledge carried by this additional user feedback. As the expected model change also requires the distribution $P_t(u|\mathcal{V}_0 \cup \{v\})$ for evaluating every candidate query v, we thus follow the same trick proposed in the previous section by restrict the computation to the set of nodes predicted to be most relevant based on the current model (i.e., $P_t(\cdot|\mathcal{V}_0)$).

Item and Attribute Query Fusion Schemes

The active dual collaborative filtering framework makes it possible to learn a user's preferences from both relevant items and relevant attributes simultaneously. With the capability to raise two different types of queries, we are also faced with two new challenges: (1) When is the best time to acquire which type of feedback? (2) How much of each type of feedbacks should be acquired? Following typical active learning evaluation methodology, we assume that the learner can ask the user a total of n queries in k iterations, with n/k queries in each iteration. Thus, the learner is able to update itself based on new user feedback in each iteration before selecting new queries for the next iteration. In this section, we explore several schemes for controlling the amount and ordering of item and attribute queries in order to understand how these two different types of feedback can be best combined to improve the effectiveness of active learning.

Attribute Before Item Using this scheme, we can first issue n_a queries to solicit only feature feedback, after which we will then issue n_i queries to solicit only item feedback. This is based on the assumption that recommendation based on attribute feedback are easier to generalize and can be more effective in cold start stages to initialize a user model more efficiently.

Attribute After Item Contrary to the previous scheme, we can also try to obtain item feedback first by issuing n_i item queries and then solicit attribute feedback in the end by issuing a sequence of n_a attribute queries. Under this scheme, the attribute queries would be determined based on a set of already obtained relevant items. This is based on the assumption that attribute feedback are best used for refining an existing model rather than initializing it.

Model Gauged Interleaving The previous two schemes are based on different heuristic assumptions of the effect of attribute feedback. Our third scheme relies on the model itself to determine the timing and number of item and attribute feedbacks to acquire. We refer to this as *model gauged interleaving*. As we discussed in Section , the random walk with restart model defined over the tripartite graph of user, item and attributes allow all our proposed query score measures model entropy and model change to be computed in a unified fashion and on the same scale. Therefore we can simply use the utilities computed over both types of queries to rank them altogether. Then, at any point during the active learning process, the model can be used to automatically determine how many queries of each type to ask.

Related Work

Collaborative filtering is a popular technology for building large scale recommender systems. By modeling correlation between users' observed feedback in the past, it is able to recommend novel items to a user based on the feedback from other similar minded users. This is achieved by either directly computing user-user similarities based on their observed feedback (Herlocker, Konstan, and Riedl 2002) or via more compact statistical models such as matrix factorization (Koren, Bell, and Volinsky 2009) and latent variable models (Hofmann 2004; Pennock et al. 2000). The random walk model was first applied to collaborative filtering in (Gori and Pucci 2007) but that model did not consider attribute information. (Melville, Mooney, and Nagarajan 2002) presented a collaborative filtering model that can also utilize item attributes. None of the existing works considered the use of attribute feedback, nor did they study the active learning aspect.

User cold-start is a common problem for recommender systems. Many researchers have studied the problem of eliciting preferences from users to initialize their profile. Rashid et al (Rashid et al. 2002) studied the problem of eliciting preferences from new users, in which they compared various heuristics include popularity, rating variance, etc. Expert Clerk (Shimazu 2001) is an agent system that simulates a human sales clerk by asking a user a series of questions to narrow down matching products. It is effective for the situation when the user is looking for specific products but not suitable for initializing a new user's profile for a recommender system, as the user's preferences would be much less well defined. Most of these existing works focused on the user interface or system design aspect of preference elicitation whereas our focus on designing active learning strategies for such systems.

Active learning in the context of regression and classification problems has been studied extensively in past, which resulted in many successful techniques such as uncertainty sampling (Lewis and Catlett 1994), expected error reduction (Roy and Mccallum 2001) and expected model change (Settles and Craven 2008) have been proposed. There is little work on active learning for collaborative filtering in the past (Jin and Si 2004; Harpale and Yang 2008), which have only considered the use of item feedback. Attribute feedback is a very novel concept but so far has only been exploited in the form of word labeling for building text classifiers in both passive(Liu et al. 2004) and active fashion(Raghavan, Madani, and Jones 2006; Sindhwani, Melville, and Lawrence 2009). Our work is the first to formalize the notion of attribute feedback.

Experiments

Data Sets

To empirically study the effectiveness of the active dual collaborative filtering framework, we use the Movielens[1] data set. To focus on the setting when user feedback information is very sparse in a recommender system, we randomly selected a set of 1000 users to form the set of existing users and extracted another set of 500 users with more than 100 ratings as active users for evaluation. The number of distinct items is 10,681. To obtain movie attributes, we crawled the IMDB pages of all the movies and extracted the actors, directors, genres and plot keywords information for each movie. We then filtered out all attributes that appeared in less than 3 movies and obtained a total of 5,866 distinct attributes in total.

Evaluation Protocol

We treat the set of movies rated by each active user as being relevant to him and the set of unrated movies as being irrelevant. 70% of each user's relevant movies are used as item feedback that may be queried during the active learning process whereas the remaining 30% are used for testing. Therefore, during the active learning process, the item queries will be selected from the pool consisting of all the relevant movies in the training set plus all the irrelevant movies. To evaluate the performance of a model, we use it to rank the pool of movies consisting of the set of relevant movies in the test set along with all the irrelevant movies. The ranking quality is then measured by *Average Precision*, which averages the precisions at the positions of each relevant items:

$$AP = \frac{\sum_{i=1}^{N} \delta^+(i) Pre@i}{N^+} \tag{5}$$

where $Pre@i$ is the proportion of relevant items in the top i ranking results, $\delta^+(i)$ is an indicator function, which equals to 1 if the i-th result is relevant and 0 otherwise. N^+ is the total number of relevant results.

Simulating Attribute Feedback

Attribute feedback is novel concept that has not be well studied in the context of recommender systems. To the best of

[1] http://www.grouplens.org

our knowledge, there is no existing public data sets containing both item and attribute feedback, nor is there any established procedure for the evaluation of such schemes. We therefore resort to the following designed procedure to derive attribute feedback. For each attribute a, let n_a denote the number of items containing this attribute. Based on the users movie ratings, we can also count the number of movies containing attribute a among the set of rated by each user u, denoted by $n_{u,a}$. Based on these two quantities, we propose to use the value $n_{u,a}/n_a$ to measure the degree to which user u likes an attribute a. Intuitively, the more movies with a particular attribute watched by a user, the more likely is the attribute relevant to the user. As $0 \leq n_{u,a}/n_a \leq 1.0$, we used a threshold of 0.6 to determine the set of attributes that are relevant to each user. This resulted in around 70 relevant attributes for each active user.

Comparing Query Fusion Schemes

In this set of experiments, we try to study how to most effectively allocate queries for acquiring item and attribute feedbacks respectively. We chose entropy minimization as the query selection strategy and compared the performances of three query fusion schemes: (1) attribute before item; (2) attribute after item; (3) model gauged interleaving. We seed each algorithm with one randomly chosen relevant item. For attribute after item and attribute before item, we let the learner issue 50 item and 50 attribute queries respectively. The MAP growth curves of the all schemes are plotted in Figure 1.

We can see that attribute after item performed significantly better than attribute before item. The result confirms the hypothesis that attribute feedback is more effective for model refinement rather than model initialization. Moreover, we can also see that the model-gauged interleaving method works very well and outperformed the two heuristic fusion schemes. This demonstrates that our random walk based active learning framework is very effective for soliciting item and attribute feedback in a unified fashion.

To better understand the behavior of the different query fusion schemes, we also counted the total number of acquired feedbacks as well as acquired item and attribute feedbacks under each scheme. These statistics are shown in Figure 2. Comparing attributes before with attributes after querying items, we can see while they acquired a similar number of item feedbacks, 'attribute after item' acquired much more attribute feedbacks. This shows that attribute feedback can be more effectively acquired based on some amount of existing item feedback. We can also see that the total number of feedbacks acquired by the model-gauged interleaving scheme is less than the other two schemes, but it acquired the most attribute feedback. This proves that acquiring attribute feedback plays a key role in the success of active dual collaborative filtering.

Comparing Query Selection Strategies

In this set of experiments, we try to see whether the proposed query selections strategies are effective for the active dual collaborative filtering framework. In addition to the proposed strategies, we also consider two naive baselines: a

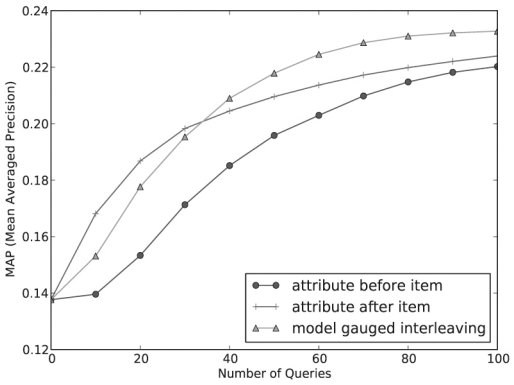

Figure 1: Comparing Different Fusion Schemes of Item and Attribute Queries

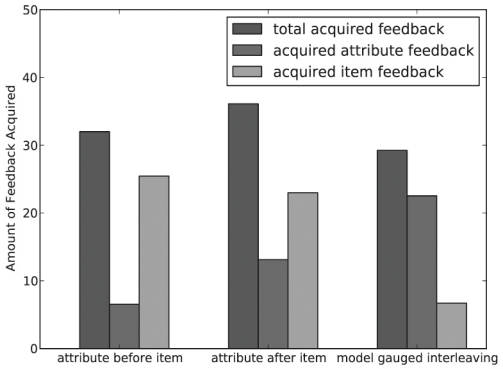

Figure 2: Comparing the Amount of Acquired Feedback with Different Fusion Schemes

random strategy which pick items or attributes randomly to query the user, and a popularity based strategy which prioritizes items and attributes based on the number of users. Finally, we also compare with a recently proposed algorithm personalized active learning (Harpale and Yang 2008), which only acquires item feedback. For all algorithms considered, we restrict the total number of queries to be 100, which are divided into 10 batches of 10 queries each. The results are shown in Figure 3, the x-axis of which is the number of queries asked and the y-axis shows the MAP-growth as more and more user feedback are obtained. From the results, we can make the following observations:

- The proposed entropy minimization and model change maximization methods can significantly outperform popularity and random strategies. Entropy minimization appeared to be the most effective query selection strategy. However, the two more advanced query strategies tend to have slow growth in the early stages of active learning process, for which the popularity based query selection appear to be more effective. This led us to wonder if a hybrid strategy which uses different types of query selectors at different stages would be more effective, which is an issue we plan to investigate in our future work.
- The item feedback based personalized active learning

method appeared to be very effective in the early iterations but its performance would quickly stop to grow as more feedback are collected. Whereas our method with model entropy and model-change-based query selectors can constantly improve the model's performance and significantly outperform the personalized active learning method in the end with more than 10% higher in MAP.

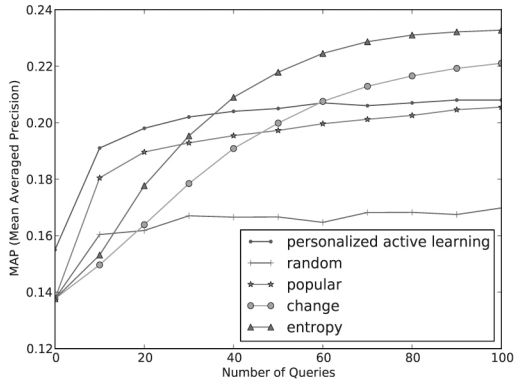

Figure 3: Comparing ADCF using Different Query Strategies with PAL

Conclusion and Future Work

In this paper, we proposed a novel *active dual collaborative filtering* framework for soliciting a new user's preference by asking the user both what items he likes (i.e., item feedback) as well as what attributes he likes (i.e., attribute feedback). We proposed a unified framework for actively acquiring item and attribute feedbacks based on the random walk with restart (RWR) model. We designed multiple query selection strategies and query fusion schemes for active learning with the two heterogeneous forms of user feedback. We have conducted extensive experiments to show that incorporating attribute feedbacks can lead to significant improvement over traditional methods that only solicit item feedbacks.

In the future, we would like to focus on improving the current model along three directions. First, we will consider computational methods for speeding up the RWR model by replacing power iterations with faster approximate algorithm. Second, we will design batch-mode query-selection strategies to avoid possible redundancy and encourage diversity within a batch of queries. Finally, we will design hybrid query-selection strategies based on the learning progress, so that the item and attribute selection strategies can be made more adaptive.

Acknowledgements

The authors would like to thank the support from NEC China Lab, Hong Kong RGC project 621010, and Hong Kong RGC/NSFC joint project N_HKUST 624/09.

References

Druck, G.; Settles, B.; and McCallum, A. 2009. Active learning by labeling features. In *Proc. of the EMNLP'09*.

Gori, M., and Pucci, A. 2007. Itemrank: A random-walk based scoring algorithm for recommender engines. In *Proc. of the IJCAI'07*. Morgan Kaufmann Publishers Inc.

Harpale, A. S., and Yang, Y. 2008. Personalized active learning for collaborative filtering. In *Proc. of the SIGIR'08*. ACM.

Herlocker, J.; Konstan, J. A.; and Riedl, J. 2002. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information retrieval* 5(4):287–310.

Hofmann, T. 2004. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst.* 22(1):89–115.

Jin, R., and Si, L. 2004. A bayesian approach toward active learning for collaborative filtering. In *Proc. of the UAI'04*, 278–285. AUAI Press.

Koren, Y.; Bell, R. M.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *IEEE Computer* 42(8):30–37.

Lewis, D. D., and Catlett, J. 1994. Heterogeneous uncertainty sampling for supervised learning. In *Proc. of the ICML'94*, 148–156. Morgan Kaufmann.

Liu, B.; Li, X.; Lee, W. S.; and Yu, P. S. 2004. Text classification by labeling words. In *Proc. of the AAAI'04*, 425–430. AAAI Press.

Melville, P.; Mooney, R. J.; and Nagarajan, R. 2002. Content-boosted collaborative filtering for improved recommendations. In *Proc. of the AAAI'02*, 187–192. American Association for Artificial Intelligence.

Pennock, D. M.; Horvitz, E.; Lawrence, S.; and Giles, C. L. 2000. Collaborative filtering by personality diagnosis: A hybrid memory and model-based approach. In *Proc. of UAI'00*, 473–480.

Raghavan, H.; Madani, O.; and Jones, R. 2006. Active learning with feedback on features and instances. *J. Mach. Learn. Res.* 7:1655–1686.

Rashid, A. M.; Albert, I.; Cosley, D.; Lam, S. K.; McNee, S. M.; Konstan, J. A.; and Riedl, J. 2002. Getting to know you: learning new user preferences in recommender systems. In *Proc. of the IUI'02*, 127–134.

Roy, N., and Mccallum, A. 2001. Toward optimal active learning through sampling estimation of error reduction. In *In Proc. of the ICML'01*, 441–448. Morgan Kaufmann.

Settles, B., and Craven, M. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proc. of the EMNLP'08*.

Shimazu, H. 2001. Expertclerk: Navigating shoppers buying process with the combination of asking and proposing. In *Proc. of the IJCAI'01*, 1443–1450.

Sindhwani, V.; Melville, P.; and Lawrence, R. D. 2009. Uncertainty sampling and transductive experimental design for active dual supervision. In *Proc. of the ICML'09*, 953–960. ACM.