Selectively Acquiring Ratings for Product Recommendation

Zan Huang
Pennsylvania State University
419 Business Building
University Park, PA 16802
1-814-863-1940
zanhuang@psu.edu

ABSTRACT
Accurate prediction of customer preferences on products is the key to any recommender systems to realize its promised strategic values such as improved customer satisfaction and therefore enhanced loyalty. In this paper, we propose proactively acquiring ratings from customers for a newly introduced product to quickly improve the accuracy of the predicted ratings generated by a collaborative filtering recommendation algorithm for the entire customer population. We formally introduce the problem of identifying the most informative ratings to acquire and termed it as the *product rating acquisition problem*. We proposed an active learning sampling method for this problem that is generic to any recommendation algorithms. Using the Netflix Prize dataset, we experimented with our proposed method, a uniform random sampling method, and a degree-based sampling method that is biased toward customers with large numbers of ratings for the user-based and item-based neighborhood recommendation algorithms. The experimental results showed that even with the random sampling method, acquiring 10% of all ratings in addition to a randomly selected 10% initial ratings achieved 4.5% improvement on overall rating prediction accuracy of the movie. In addition, our proposed active learning sampling method consistently outperformed the random and degree-based sampling for the better-performing item-based algorithm and achieved more than 8% improvement by acquiring 10% of the ratings.

Categories and Subject Descriptors
H.1.2 [**Information Systems**]: User/Machine systems -*Human information processing*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval - *Information filtering; Relevance feedback; Retrieval models.*

ICEC'07, August 19–22, 2007, Minneapolis, Minnesota, USA.

General Terms
Algorithms, Design, Experimentation.

Keywords
Recommender system, collaborative filtering, active learning, sampling.

1. INTRODUCTION
Recommender systems are being widely adopted in many application settings to suggest products, services, and information items to potential customers. A wide range of companies, such as *Amazon.com*, *Netflix.com*, *Half.com*, *CDNOW*, *J.C. Penney*, and *Procter & Gamble*, have successfully deployed recommendation technologies to increase Web and catalog sales and improve customer loyalty [33]. For some of these online businesses, the recommendation service plays a central role in the business strategy, with *Amazon.com* and *Netflix.com* as the prominent examples.

Accurate prediction of customer preferences on products is the key to any recommender systems. There has been a substantial literature on recommendation algorithm design and evaluation. There is also ever increasing industry interest and efforts in improving the performance of real-world recommender systems, highlighted by the Netflix Prize launched in October 2006.

The key input for a recommendation algorithm is the previously observed customer-product interactions, implicit (such as the purchase of a product by an Amazon customer) or explicit (such as ratings of a movie by a Netflix customer). In this paper we limit our focus on recommendation derived from rating-based explicit interactions. Previous efforts on improving recommendation performance have primarily focused on designing a better recommendation algorithm given a set of customer-product interactions and characteristics of the customers and products. As is true for any data mining task, a set of competitive algorithms may deliver slightly varying levels of performances within a certain range largely dictated by the nature of the input data. In the context of an operational recommender system, it is possible to influence the nature of the available customer-product interactions by proactively acquiring customers' ratings. The carefully acquired ratings have the potential to significantly improve the accuracy of the predicted ratings for the population of customers, especially at the stage when a new product just enters the system.

Consider the example of Netflix's recommendation service. When a new movie becomes available on DVD but few customers have provided ratings for the movie, the accuracy of the predicted ratings on this movie is expected to be low. Netflix has the incentive to quickly improve the prediction accuracy to avoid customer dissatisfaction. In this situation, sending the movie free of charge to a set of selected customers to get their ratings could bring significant return on customer satisfaction that overweighs the cost associated with it.

In this paper, we formalize the problem of selectively acquiring customer ratings to improve product recommendation quality, i.e., for a given number of ratings to acquire identifying which customers should be reached for their ratings in order to achieve the largest improvement on overall rating prediction accuracy for a particular product. We term this problem as the *product rating acquisition* problem.

There is substantial literature on selective sampling for building classifiers in a cost-sensitive context. Many algorithms have been proposed under the *active learning* scheme, in which labeled data are acquired incrementally, using the model learned using the available data so far to select particularly helpful additional training examples. In this paper, we propose an active learning sampling algorithm for the product rating acquisition problem and compare with two benchmark sampling methods, a random sampling method and a *degree-based* sampling method that selects customers with the greatest number of ratings. We use the data from Netflix Prize to evaluate the proposed active learning method and assess the potential gain from proactive acquisition of customer ratings.

The remainder of the paper is organized as follows. Section 2 reviews relevant literature on recommendation algorithms, data acquisition and active learning research, and the limited previous studies on user-oriented active learning for recommendation. In Section 3 we formalize the product rating acquisition problem and describe the proposed active learning method and the random and degree-based sampling methods. Section 4 provides the details of the experimental study using the Netflix dataset. Section 5 concludes the paper by summarizing the contributions and pointing out the future directions of this research.

2. RELATED WORK

In this section, we first review the literature on recommendation algorithm research and describe the two commonly used algorithms that will be used in this study, the user-based and item-based neighborhood algorithms. We then summarize the literature on data acquisition and active learning research. Lastly, we review few previous studies that apply the idea of active learning for recommendation.

2.1 Recommendation Algorithms

At the heart of recommender systems are the algorithms for making recommendations. Models based on the item or user attributes attempt to explain user-item interactions using these intrinsic attributes (e.g., [23, 25]). Intuitively these models learn either explicitly or implicitly rules such as "Joe likes science fiction books" and "well-educated users like *Harry Potter*." Techniques such as regression and classification algorithms have been used to derive such models. The performances of these approaches, however, rely heavily on high-quality user and item attributes that are often difficult or expensive to obtain.

Collaborative filtering-based recommendation takes a different approach by utilizing only the customer-product interaction data and ignoring the customer and product attributes [26]. Based solely on the interaction data, customers and prodcuts are characterized implicitly by their previous interactions. Collaborative filtering has been reported to be the most widely-adopted and successful recommendation approach. The literature on recommendation algorithm design has largely focused on collaborative filtering algorithms. A wide range of collaborative filtering algorithms have been proposed, including standard user-based and item-based neighborhood algorithms [26, 32], rule-based approaches [1, 21], cluster and generative models [12, 35], advanced matrix analysis approaches [11, 31], and graph-based algorithms [14, 17].

In this paper, we focus on collaborative filtering recommendation based on the rating data. We focus on the standard user-based and item-based neighborhood algorithms in this paper. The user-based algorithm was the first collaborative filtering algorithm proposed in the literature and often serves as a comparison benchmark for later proposed algorithms. A number of evaluation studies (e.g., [6]) have shown that this algorithm can achieve competitive performance with many other algorithms. The item-based algorithm is adopted by real-world systems such as Amazon's [22] because of its relative computational efficiency when customers substantially outnumber the products in the system. It has also been shown to outperform the user-based algorithm for many datasets [9, 32].

The input for collaborative filtering algorithms is a set of ratings $R = \{r_{c,p}\}$ given by a set of customers $C = \{c_1, c_2, \ldots, c_M\}$ on a set of products $P = \{p_1, p_2, \ldots, p_N\}$.

The user-based neighborhood algorithm constructs a customer similarity matrix based on the customers' co-rated products. For a target product, the predicted rating for a customer is derived from aggregating the observed ratings given by the customers similar to this customer. The item-based algorithm operates in a similar manner but constructs the product similarity matrix instead. The similarity for a product pair is based on the ratings from customers who rated both products. The predicted rating for a customer is derived from aggregating her ratings on products that are similar to the target product.

There are a variety of design choices on customer/product similarity matrix construction and prediction generation. We have adopted the designs in [6, 32], which have been demonstrated to have competitive performance with other designs.

For the user-based algorithm, the similarity between customers c_i and c_j is given by the Pearson-r correlation

$$w_c(i,j) = \frac{\sum_{p \in P_{i,j}} (r_{c_i,p} - \bar{r}_{c_i})(r_{c_j,p} - \bar{r}_{c_j})}{\sqrt{\sum_{p \in P_{i,j}} (r_{c_i,p} - \bar{r}_{c_i})^2 \sum_{p \in P_{i,j}} (r_{c_j,p} - \bar{r}_{c_j})^2}} \quad (1)$$

where $P_{i,j}$ denotes the set of products both customers c_i and c_j have rated and $\bar{r}_c$ denotes customer c's overall average rating.

Similarly, for the item-based algorithm, the similarity between products p_i and p_j is given by

$$w_p(i,j) = \frac{\sum_{c \in C_{i,j}} (r_{c,p_i} - \bar{r}_c)(r_{c,p_j} - \bar{r}_c)}{\sqrt{\sum_{c \in C_{i,j}} (r_{c,p_i} - \bar{r}_c)^2 \sum_{c \in C_{i,j}} (r_{c,p_j} - \bar{r}_c)^2}} \quad (2)$$

where $C_{i,j}$ denotes the set of customers who have rated both products p_i and p_j. Note that we do not follow exactly the Pearson correlation formula but adopted the formulation used in [32]. This adjusted version captures differences in rating scale between different customers and was reported to deliver superior performance over the Pearson correlation formulation.

We adopted the weighted sum approach for rating prediction generation. For the user-based algorithm, the predicted rating on product p for customer c is given by

$$p_{c,p} = \bar{r}_c + \frac{\sum_{c' \in C} w(c,c')(r_{c',p} - \bar{r}_{c'})}{\sum_{c' \in C} | w(c,c') |} . \quad (3)$$

Similarly, the item-based algorithm generates the rating prediction by

$$p_{c,p} = \bar{r}_c + \frac{\sum_{p' \in P} w(p,p')(r_{c,p} - \bar{r}_c)}{\sum_{p' \in P} | w(p,p') |} \quad (4)$$

The performances of the large set of recommendation algorithms proposed in the literature have been compared using real-world recommendation data in many algorithm and evaluation studies. Many of these studies (e.g., [6, 16]) confirmed the finding that across different recommendation datasets no single algorithm outperforms all the others and that the general level of recommendation quality is related to certain inherent characteristics of the customer-product interaction data other than the number of ratings available (e.g., [15]). It is often found that the performance differences among a set of competitive algorithms are smaller than the performance differences across datasets. These findings indicate that selecting a proper structure of the available rating data is likely to significantly improve the overall rating prediction accuracy given the same number of ratings.

2.2 The Data Acquisition Problem and Active Learning

There is a substantial literature on information acquisition for predictive modeling, targeting at building the most accurate model with a specified number of labeled training instances selected from a pool of examples. This idea of selecting the most 'useful' subset of examples to build a model comparable to model derived from the entire pool of examples, as backed by the theoretical results in [2], is intriguing for domains where training examples are costly to acquire.

Information acquisition has been studied in a variety of settings including the multi-armed bandit problem originally proposed by Robbins [27] and the optimal experiment design (OED) [3, 19] in statistics and active learning [7] in machine learning research. The general objective is to select the most informative data to improve the predictive power of the model, where the predictive models range from simple univariate distribution estimations of a single random variables to parametric statistical function relating distribution of the dependent variable to independent variables to non-parametric machine learning models such as categorical classifiers. The major difference between OED and active learning is on the parametric or non-parametric nature of the model at study [30]. OED studies optimal data acquisition for parametric statistical models, for which close-form objective functions that relates to certain notion of the utility of the acquired data can be specified [37]. In this context, the data acquisition problem is eventually formulated as an optimization problem. Active learning, on the other hand, typically deals with non-parametric machine learning models for which the close-form utility function of a candidate data observation cannot be derived. Active learning acquires information sequentially based on the model learned so far. Most active learning research has focused on categorical classification. A major class of active learning methods relies on certain measures of uncertainty of the currently held model with respect to individual data instances to guide acquisition. The general notion is that the data instances with the greatest model uncertainty should be acquired such that the model can be improved the most. A wide variety of uncertainty measures have been proposed in the literature such as voting-based measure in the Query by Committee algorithm [10], probability of binary class membership [20], and local class-probability-estimation error [29]. The second class of active learning methods directly optimizes the expected error on future test examples (e.g., [8, 28]). Both classes of methods rely on analytically or numerically derived probabilistic estimates (regarding the model uncertainty or the prediction error).

2.3 Active Learning for Recommendation

There have been only a few previous studies that explore the application of the active learning idea in the recommendation context. These studies follow the tradition in active learning research for categorical classification to formulate uncertainty measures of the current collaborative filtering model with respect to individual ratings to guide acquisition. The proposed active learning methods in these studies are designed specifically for a particular type of probabilistic collaborative filtering models. Boutilier et al. [5] proposed acquiring rating based on the expected value of information for the multiple-cause vector quantification model. Yu et al. [36] used a measure of the entropy of the likemindedness for rating acquisition for a probabilistic memory-based collaborative filtering model, in which a rating on a product by a customer is related probabilistically to a set of hidden customer profile prototypes. Jin and Si [18] proposed using entropy of the model parameters to guide rating acquisition which is in principle applicable to the probabilistic latent class model [13], personality diagnosis model [24], and mixture models [34].

To the best of our knowledge, no previous study has investigated the generic active learning method that is applicable to any recommendation algorithms, especially the heuristic memory-based algorithms such as the popular user-based and item-based neighborhood algorithms. These previous work is also limited to the "new user problem," trying to find informative additional products for a new customer to rate in order to quickly improve the recommendation quality for this particular customer. In this study, we focus on the "new product problem," trying to quickly improve the accuracy of the predicted ratings of all customers for a newly introduced product. In operational commercial recommender systems, such as Netflix, such a rating acquisition strategy would have a much wider impact on the customer base

and the overall customer satisfaction out of the recommendation service.

3. THE PRODUCT RATING ACQUISITION PROBLEM

3.1 Problem Setup

We formalize the generic product rating acquisition problem as follows. For a new product q, a small set of ratings may become available naturally provided by the early raters C^0. We denote this set of rating as the initial rating set $R^0 = \{r_{c,q}\}$, where $c \in C^0$. A set of target customers $C = \{c_1, c_2, \ldots, c_M\}$ is identified for whom the rating prediction is needed. Denote the actual (unobserved) rating of q by the target customers as $R_q = \{r_{c,q}\}$, where $c \in C$. In practice, the target customer set could be the entire customer population or a subset depending on to which customers the product p is of *potential* interest. We assume $C^0 \subset C$. Based on the target customer set the relevant product set $P = \{p_1, p_2, \ldots, p_N\}$ is determined. Products in P have been rated by more than a specified minimum number k of customers in C. The set of ratings given by customers in C on products in P is denoted as $R = \{r_{c,p}\}$, where $c \in C$ and $p \in P$.

In order to improve the accuracy of rating predictions of the product q for the target customers, we identify s customers $C^1 = \{c \mid c \in C, c \notin C_0\}$, where $|C^1| = s$, to acquire their ratings $R_q(C^1) = \{r_{c,q}\}$ $(c \in C^1)$ on q to improve the overall rating prediction accuracy on the entire target customer set. With a particular product recommendation function, f: $R \times q \rightarrow R_q'$, where $R_q' = \{r_{c,q}'\}$, $c \in C$, we can obtain the predicted ratings of the target customers on q. Our objective is to find the set C^1 such that the error of the predicted ratings of customers in C on q, $Error(R_q, R_q')$, is minimized. A number of forms of rating prediction error have been used in the literature, such as mean absolute error (MAE), mean square error (MSE) and root mean square error (RMSE).

The input and output of the product rating acquisition problem formulation is summarized in Figure 1.

Input: target product q, target customer set C, relevant product set P, early raters C^0 and their ratings on q R^0, number of additional ratings to acquire s, a recommendation algorithm f that takes as input the rating set R and a target product q and produces predicts q's predicted ratings by the customers R_q'

Output: acquisition set C^1, where $c \in C$, $c \notin C_0$, and $|C^1| = s$, such that $Error(R_q, f(R \cup R^0 \cup R_q(C^1), q))$ is minimized

Figure 1. Product rating acquisition Problem

3.2 Rating Acquisition Methods

In this section we describe the proposed active learning sampling method for rating acquisition. We also describe two straightforward benchmark sampling methods, the uniform random sampling and degree-based sampling.

3.2.1 Active Learning Sampling

Our proposed method adopts the fundamental notion of active learning schemes for learning classifiers. The candidate customers to acquire ratings are selected based on the prediction error of the recommendation function derived from the currently available ratings.

Most existing active learning methods identify examples for acquisition based on certain notion of the uncertainty of the current model. Take the *Uncertainty Sampling* [20] for binary classification as an example, the most informative examples are identified as the ones for which the current classifier estimates a class membership probability closest to 0.5. The idea is that the current classification model is most uncertain about the class membership of these examples and their class labels should be acquired to achieve the largest improvement on the model. Such an approach relies on the classifier's ability to provide predictions in a probabilistic form. In order to follow exactly the uncertainty sampling approach to perform rating acquisition, we have to be limited to the recommendation algorithms that can provide some uncertainty measure of the predicted rating, such as the latent class model and mixture models. For a large number of other recommendation algorithms, such measures of rating prediction uncertainty have not been well-studied, making such a sampling method rarely applicable.

In order to develop a generic active learning sampling method for any recommendation algorithm, we rely on the observed prediction errors on the customers whose actual ratings are available, C^0. Based on the recommendation function derived from the available ratings we can obtain predicted ratings of the new product for the target customers, $R_q' = f(R \cup R^0, q)$. Among these target customers, for $c \in C^0$ the observed prediction error is available $(r_{c,q} - r_{c,q}')$ but for the rest of the customers (candidates to acquire ratings) actual rating is not available yet to derive prediction error. Here the challenge for active learning sampling is that the uncertainty of the prediction with respect to a particular customer only becomes known after his/her rating has already been acquired. As the rating-based similarity is the fundamental data pattern any recommendation algorithm relies on to generated predictions, we propose to sample the close neighbors of the customers for which the current large rating prediction error is observed. Under the fundamental assumption of collaborative filtering, similar customers are expected to continue to show similarity in rating new products. Building upon this assumption, it is reasonable to expect that the current recommendation quality is poor for the close neighbors of the customers whose observed prediction error is large. These customers' ratings should be acquired to improve the overall recommendation quality.

Figure 2 presents this active learning sampling algorithm.

The algorithm starts with computing the customer similarities using Eq (1) introduced in Section 2.1. Predicted ratings on the new product q are then generated by a given recommendation algorithm using the ratings on relevant products (R) and available ratings on product q (R^0). The absolute prediction error is obtained for customers in C^0 and for each of these customers a selection probability proportional to the absolute prediction error is computed. We then perform s random draws from the customers in C^0 based on their selection probability. Each time a customer c_i is drawn, her closest unselected neighbor is added to

the selection set. Intuitively we bias the sampling towards the areas with observed large prediction errors in the customer similarity space. The probabilistic selection process is used based on the success of such procedures in evolutionary computing such as genetic algorithms.

Active Learning Sampling Algorithm

Input: target product q, target customer set C, relevant product set P, early raters C^0 and their ratings on q R^0, number of additional ratings to acquire s, a recommendation algorithm f that takes as input the rating set R and a target product q and produces predicts q's predicted ratings by the customers R_q'

1. Compute the customer similarities $\{w(i,j)\}$ based on $R \cup R^0$ following Eq (1)

2. Apply the recommendation algorithm f to obtain predicted ratings: $R_q' = f(R \cup R^0, q)$

3. Compute the absolute prediction errors for customers in C^0: $E = \{e_c = | r_{c,q} - r_{c,q}' | \}$

4. Compute selection probability based on E: $PR = \{pr_c\}$, $pr_c = | r_{c,q} - r_{c,q}' | / \sum_{c \in C^0} | r_{c,q} - r_{c,q}' |$

5. Set the selection set S to be empty

6. While $| S | < s$

7. Randomly select a customer c_i from C^0 according to selection probability PR

8. Find the closest unselected neighbor c_n of c_i to add to S: $n = \arg\max_{c_i \notin C^0, c_i \notin S} (| w_c(i,n) |)$

Output: acquisition set C^1, where $c \in C$, $c \notin C_0$, and $| C^1 | = s$, such that $Error(R_q, f(R \cup R^0 \cup R_q(C^1), q))$ is minimized

Figure 2. Active learning sampling algorithm

We next examine in detail the impact of the new ratings acquired following the active learning sampling for both recommendation algorithms. Under the user-based algorithm, a large observed rating prediction error for customer c could be due to lack of information on the customer neighborhood (either none or very few of c's neighbors have rated product q therefore there is no information to infer c's rating). Acquiring the rating by a close neighbor of c enhances the information available about the match between q and this customer neighborhood and improves the accuracy of predicted ratings for the customers in this neighborhood. Under the item-based algorithm, a large observed rating prediction error for customer c could be related to lack of information on q's relationship to the products previously rated by c (either none or very few products rated by c appear within the neighborhood of product q). Acquiring the rating of c's close neighbor helps bring the maximum amount of information on the relationship between q and these products, as c' by definition has strong correlation with c on co-rated products. With the additional rating $r_{c',q}$, each pair of highly correlated ratings by c and c' on the co-rated products will now contribute to form the similarity between q and these co-rated products.

3.2.2 Sequential Active Learning Sampling

One major advantage of active learning sampling is that it can be performed sequentially to further improve the sampling performance. In the recommendation context, a relatively small set of ratings can be acquired at each step. The algorithm in Figure 2 can be repeatedly applied to perform the sequential active learning sampling. Measures can be taken to alleviate the heavy computing demand for this approach. Note that when there are a large number of relevant products, additional ratings acquired on the new product along the active learning sampling process have negligible effect on the customer similarities. Therefore, we can use the customer similarities computed using $R \cup R^0$ throughout the entire sampling process. Depending on the specific recommendation algorithm used, the efficiency for rating prediction generation process can also be improved based on the model previously built.

3.2.3 Benchmark Sampling Methods

The most straightforward sampling method is the uniform *random sampling* method. Under this method, each unselected customer is chosen with equal probability.

Another intuitive method is to prefer the active customers who have previously rated large numbers of products. One may conjecture that these customers might be experts and plays a role of opinion leader in the customer community. Following the terminology in networks, we refer these customers as *high-degree* customers (who have large numbers of links in a customer-product rating graph) and such a sampling method as *degree-based sampling*. Under this method, an unselected customer is chosen with a probability proportional to his/her degree.

4. AN EXPERIMENTAL STUDY

4.1 Data

We use the rating data from Netflix Prize competition [4] to empirically test the performance of our proposed active learning sampling algorithm in comparison with the benchmark sampling methods. The Netflix Prize competition was launched in October 2006, for which a dataset containing 100 million anonymous movie ratings was released.

We sampled eight movies of varying level of popularity released in 2005 from the Netflix Prize training dataset. We did not include the extremely popular movies as the proactive rating acquisition might not be necessary for these movies. For each movie, we identified all customers who gave ratings and treated this set of customers as the target customers C. Their ratings on the movie constitute the set R_q. The relevant movie set P is formed by the movies released prior to 2005 rated by more than 20 customers in C. The customers in C without any rating on the relevant movies were removed from C. The rating set R consists of ratings given by customers in C on movies in P. Table 1 shows the sample movies used in this study.

Table 1. Sample movies used in the experiment

Movie Id	Title	$\|C\|$	$\|P\|$	$\|R\|$
1	Thomas & Friends: Calling All Engines	59	176	4917
2	Johnny Cash: Ridin' the Rails: The Great American Train Story	64	179	4834
3	Kelly Clarkson: Behind Hazel Eyes	100	421	14858
4	The Work of Director Jonathan Glazer	100	436	15589
5	Fraggle Rock: Live by the Rule of the Rock	510	2393	199978
6	The Life and Times of Frida Kahlo	514	1917	140083
7	Barbie and the Magic of Pegasus	999	1898	209161
8	Mondovino	1038	2057	210801

4.2 Experiment Procedure

For each target movie q, we first randomly select about 10% of the target customer set C to acquire their ratings. These ratings are intended to simulate the ratings that are naturally available in the system and form the initial rating set R^0. In order to get a complete picture of the impact of the acquired ratings on overall recommendation quality we vary the acquisition size by deciles from 10% to 90% of the target customer set. The last acquisition set always exhausts the entire target customer set as 10% of the customers were already used for initial rating set. Presumably, as more ratings are acquired the recommendation quality should improve. As the acquisition size approaches 90% of the target set the marginal improvement should diminish as the information value of the additional ratings decreases when a large number of ratings are already available.

We perform sequential sampling as the acquisition size increases. For example, to reach the acquisition size of 50% we use the 10% initial ratings and the 30% additional ratings acquired in the previous step to form the current initial ratings. An additional 10% of the ratings are acquired following random, degree-based, and active learning sampling methods.

After each step of acquisition, predicted ratings of all customers in the target set R_q' are computed using the user-based and item-based algorithms. Accuracy measures of these predicted ratings are computed comparing to their actual ratings R_q. In this study we adopt the commonly used MAE, MSE, and RMSE as rating prediction accuracy measures. These measures are computed as follows.

$$\text{MAE} = \sum_{c \in C} | r_{c,q} - r_{c,q}' |$$

$$\text{MSE} = \sum_{c \in C} (r_{c,q} - r_{c,q}')^2$$

$$\text{RMSE} = \sqrt{\sum_{c \in C} (r_{c,q} - r_{c,q}')^2}$$

Note that the active learning sampling method relies on the predicted ratings computed in the previous step. Therefore substantially different samples may be selected depending on the user-based or item-based algorithm is incorporated in the sampling process if the two algorithms generate substantially different rating predictions. In our experiments, when the user-based (item-based) algorithm is used to generate the recommendation after acquisition, the user-based (item-based) algorithm is also used during the acquisition process. It is possible though, for example, to perform active learning sampling using item-based algorithm and use the resulting rating set to perform user-based recommendation. We leave such a setup for future research.

4.3 Results

Figure 3 shows the results of our experiments on the 8 sample movies. We only report the MAE measure in this paper due to space limitation. The MSE and RMSE measures we obtained showed similar patterns.

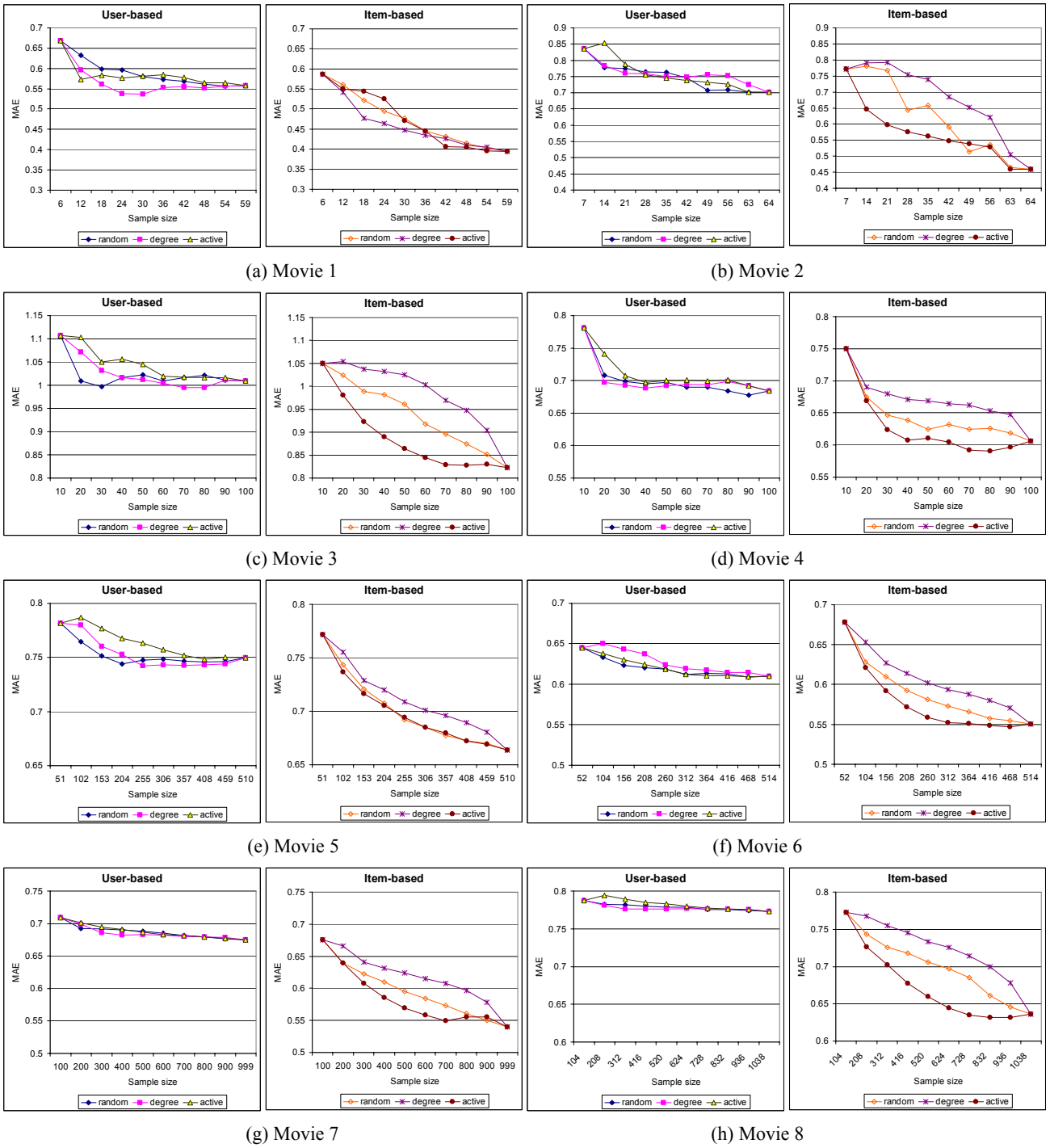
Overall, we did observe that as more ratings were acquired to be used for recommendation, the prediction error of both algorithms generally decreased as expected, for all three sampling methods. There were several exceptions, though, typically at the early stages of the rating acquisition process. This general decreasing trend of the prediction error confirmed that the fundamental assumption for collaborative filtering is consistent with the reality to certain extent and the recommendation algorithms we used have some predictive power. We also observed that for large-size target customer sets, the additional ratings were less valuable in improving the recommendation performance. This is likely to be the result of our experiment procedure. Because we always used 10% randomly selected ratings as the initial rating set, for a movie with a large target customer set the initial rating set was also large. Had we used a fixed batch size to acquire additional ratings, we would likely to observe the same steep decreasing curve at the beginning for the movies with large target customer set as well. However, in practical systems the popular movies are also likely to naturally get larger numbers of ratings from early raters. Therefore, our experiment may capture the actual potential gain for Netflix from adopting the proactive rating acquisition strategy for different types of movies.

Table 2. MAE differences between using initial 10% ratings and all ratings

Movie Id	User-based	Item-based	Item-based/ User-based
1	0.11118	0.19159	1.72321
2	0.13434	0.31285	2.32887
3	0.09793	0.22663	2.31425
4	0.09701	0.14361	1.48034
5	0.03171	0.10816	3.41036
6	0.03517	0.12754	3.62608
7	0.03451	0.13593	3.93924
8	0.01432	0.13643	9.52868
Avg	0.06952	0.17284	3.54388

We found that for the sample movies we studied the item-based algorithm consistently outperformed the user-based algorithm as more ratings were acquired for recommendation. Across movies, the predicted ratings generated by the two algorithms using the 10% initial rating set had similar MAE measures with differences

less than 0.1. Of the eight movies, item-based algorithm had lower initial MAE than the user-based algorithm for six movies. However, as we acquired additional ratings to be used for generating recommendations, we observed that the prediction error of the item-based algorithm experienced much greater decrease than the user-based algorithm. Table 2 shows the MAE differences between the recommendations generated from the initial rating set and the entire rating set for user-based and item-based algorithms. The reduction in MAE achieved by the item-based algorithm was on average 3.54 (ranging from 1.48 to. 9.53) times of that achieved by the user-based algorithm.



(a) Movie 1 (b) Movie 2

(c) Movie 3 (d) Movie 4

(e) Movie 5 (f) Movie 6

(g) Movie 7 (h) Movie 8

Figure 3. MAE measures of predicted ratings by user-based and item-based algorithms with different training sample sizes using three rating acquisition methods.

Table 3 shows the percentage improvement in MAE by acquiring additional ratings to use 20%, 40%, and 60% of all ratings over using the 10% initial ratings for the item-based algorithm. We reported the random sampling and active learning sampling methods here. On average, randomly selecting additional ratings to acquire achieved 4.51%, 11.443%, and 15.751% improvements over the initial recommendation when ratings used for recommendation reached 20%, 40%, and 60%. The corresponding improvements using the active learning sampling methods were 8.023%, 15.007%, and 19.484%. These are substantial improvements considering that Netflix offered 1 million dollars for a 10% improvement.

Table 3. Percentage improvements of using 20%, 40%, and 60% ratings over using 10% initial ratings for the item-based algorithm

% of ratings used	Movie Id	Random	Active	Difference
20%	1	4.417%	6.236%	1.818%
	2	-1.146%	16.155%	17.301%
	3	2.454%	6.606%	4.152%
	4	10.007%	10.877%	0.870%
	5	3.678%	4.550%	0.872%
	6	7.422%	8.388%	0.966%
	7	5.402%	5.350%	-0.051%
	8	3.843%	6.021%	2.178%
	Avg	4.510%	8.023%	3.513%
40%	1	15.678%	10.421%	-5.257%
	2	16.650%	25.376%	8.727%
	3	6.512%	15.234%	8.721%
	4	14.862%	19.001%	4.139%
	5	8.306%	8.592%	0.286%
	6	12.670%	15.706%	3.036%
	7	9.745%	13.391%	3.645%
	8	7.120%	12.335%	5.215%
	Avg	11.443%	15.007%	3.564%
60%	1	24.158%	24.109%	-0.049%
	2	23.395%	28.952%	5.557%
	3	12.531%	19.528%	6.997%
	4	15.790%	19.404%	3.614%
	5	11.249%	11.232%	-0.017%
	6	15.490%	18.550%	3.060%
	7	13.595%	17.437%	3.842%
	8	9.799%	16.660%	6.861%
	Avg	15.751%	19.484%	3.733%

These experimental results show that improving rating prediction accuracy by rating acquisition could be a viable strategy. Even with random sampling, acquiring ratings of an additional 10 percent of the customers can result in 4.51% in improvement on rating prediction accuracy. Our results also show that the proposed active learning sampling methods substantially outperformed the random sampling method. On average, the active learning sampling achieved over 3.5% improvement more than the random sampling (3.513%, 3.564%, and 3.733% when 10%, 30%, and 50% additional ratings were acquired).

We also present the MAE measures of the item-based algorithm under random and active learning sampling in Table 4.

Table 4. MAE measures of using 20%, 40%, and 60% ratings for the item-based algorithm

% of ratings used	Movie Id	Random	Active	Improvement
20%	1	0.56053	0.54987	0.01066
	2	0.78032	0.64684	0.13347
	3	1.02388	0.98030	0.04358
	4	0.67496	0.66843	0.00652
	5	0.74331	0.73658	0.00673
	6	0.62789	0.62134	0.00655
	7	0.63937	0.63972	-0.00035
	8	0.74321	0.72638	0.01683
	Avg	0.72418	0.69618	0.02800
40%	1	0.49449	0.52532	-0.03083
	2	0.64302	0.57570	0.06732
	3	0.98129	0.88974	0.09154
	4	0.63854	0.60750	0.03104
	5	0.70759	0.70539	0.00221
	6	0.59230	0.57171	0.02059
	7	0.61002	0.58538	0.02464
	8	0.71788	0.67758	0.04030
	Avg	0.67314	0.64229	0.03085
60%	1	0.44476	0.44505	-0.00029
	2	0.59099	0.54812	0.04287
	3	0.91811	0.84467	0.07344
	4	0.63159	0.60448	0.02711
	5	0.68489	0.68502	-0.00013
	6	0.57317	0.55242	0.02075
	7	0.58400	0.55803	0.02597
	8	0.69717	0.64414	0.05303
	Avg	0.64058	0.61024	0.03034

The comparison among the three sampling methods was less clear for the user-based algorithm. There was no single sampling method consistently outperformed the other two. Surprisingly, the active learning sampling had generally the worst performance for Movies 3, 4, 5, and 8. It is not exactly clearly yet why the sampling methods had different performance on the user-based versus the item-based algorithms. One possible explanation is that the customer similarity is not as reliable as movie similarity in the Netflix data. That is to say, if two movies have been rated similarly by a large number of customers, another customer is most likely to rate the two movies similarly as well. However, if two customers have rated

a large number of movies similarly in the past, they are still likely to disagree on a new movie. That explains the generally superior performance of the item-based algorithm in our experiments. Given this understanding, the relative performance of active learning sampling for user-based and item-based algorithms can be explained by the fact that active learning sampling only relies on the static (not so reliable) customer similarity for user-based algorithm but helps quickly improving the reliability of movie *q*'s similarity with other movies all the time for the item-based algorithm.

5. CONCLUSIONS AND FUTURE DIRECTSION

In this paper, we proposed an alternative strategy to improve rating prediction accuracy on a new product in recommender systems by selectively acquiring informative ratings from customers. We formalized this product rating acquisition problem and proposed an active learning sampling method that is generic to any recommendation algorithms. Our approach relies on the construction of customer neighborhoods based on their similarities in past ratings and sequentially selects close neighbors of the customers who have provided ratings deviating most from the prediction from the current model. Using the Netflix Prize dataset, we evaluated our proposed sampling method and two benchmark methods, uniform random sampling and degree-based sampling that prefers the customers who have previously rated large number of products. Two versions of neighborhood collaborative filtering algorithms that have been commonly used in research and practice, the user-based and item-based algorithms were used in our experiment. The experimental results showed that proactively acquiring additional ratings from the customers (even randomly) can quickly improve the overall rating prediction accuracy of a new product substantially, especially for the item-based algorithm. This finding provides empirical support for our proposed rating acquisition strategy to improving recommendation quality on newly introduced products. Our proposed active learning sampling method also substantially outperformed the benchmark sampling methods for the item-based algorithm, which had significantly better performance than the user-based algorithm in our experiments. This finding confirms that additional ratings should be acquired *selectively* to achieve the maximum improvement on recommendation quality.

To the best of our knowledge, this is the first study to investigate the application of the active learning notion to the recommendation of new products. Our proposed active learning sampling method also departs from the existing active learning methods in the sense that it relies on the observed prediction errors on the acquired examples to perform selection and is applicable to any recommendation algorithms.

We believe the rating acquisition problem introduced in this paper is an intellectually intriguing problem that has large potential impact in practice of e-commerce recommender systems. A wide variety of ideas are yet to be explored on designing effective sampling methods for product rating acquisition, which is a major future direction of this research. Our current formulation of the product rating acquisition problem is limited to a single new product. It is interesting to explore whether the correlations among the multiple products can be exploited to improve the acquisition effectiveness. Meanwhile, we are also extending the current work by experimenting on additional movie rating data and exploring the detailed analysis targeted at explaining the different behavior with the user-based and item-based algorithms.

6. ACKNOWLEDGMENTS

This research is partly supported by a grant from Smeal School of Business, Pennsylvania State University. We also want thank Netflix for making Netflix Prize dataset available for research use.

7. REFERENCES

[1] Adomavicius, G. and Tuzhilin, A. Using data mining methods to build customer profiles. *IEEE Computer*, *34* (2). 74-82, 2001.

[2] Angluin, D. Queries and concept learning. *Machine Learning*, *2*. 319-342, 1988.

[3] Atkinson, A. The usefulness of optimum experimental designs. *Journal of the Royal Statistical Society*, *58*. 59-76, 1996.

[4] Bennett, J. and Lanning, S. The Netflix Prize, Netflix, 2007.

[5] Boutilier, C., Zemel, R.S. and Marlin, B., Active collaborative filtering. in *Nineteenth Annual Conference on Uncertainty in Artificial Intelligence*, (2003), 98-106.

[6] Breese, J.S., Heckerman, D. and Kadie, C., Empirical analysis of predictive algorithms for collaborative filtering. in *Fourteenth Conference on Uncertainty in Artificial Intelligence*, (Madison, WI, 1998), Morgan Kaufmann, 43-52.

[7] Cohn, D., Atlas, L. and Ladner, R. Improved generalization with active learning. *Machine Learning*, *15*. 201-221, 1994.

[8] Cohn, D.A., Ghahramani, Z. and Jordan, M.I. Active learning with statistical models. *Journal of Artificial Intelligence Research*, *4*. 129-145, 1996.

[9] Deshpande, M. and Karypis, G. Item-based top-N recommendation algorithms. *ACM Transactions on Information Systems*, *22* (1). 143-177, 2004.

[10] Freund, Y., Seung, H.S., Shamir, E. and Tishby, N. Selective sampling using the query by committee algorithm. *Machine Learning*, *28* (2-3). 133-168, 1997.

[11] Goldberg, K., Roeder, T., Gupta, D. and Perkins, C. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, *4* (2). 133-151, 2001.

[12] Hofmann, T. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, *22* (1). 89-115, 2004.

[13] Hofmann, T. and Puzicha, J., Latent class models for collaborative filtering. in *International Joint Conference in Artificial Intelligence*, (Stockholm, 1999), Morgan Kaufmann, 688-693.

[14] Huang, Z., Chen, H. and Zeng, D. Applying associative retrieval techniques to alleviate the sparsity problem in

collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, *22* (1). 116-142, 2004.

[15] Huang, Z. and Zeng, D., Why does collaborative filtering work? -- Recommendation model validation and selection by analyzing random bipartite graphs. in *Fifteenth Annual Workshop on Information Technologies and Systems (WITS 2005)*, (Las Vegas, NV, 2005).

[16] Huang, Z., Zeng, D. and Chen, H. A comparative study of recommendation algorithms for e-commerce applications. *IEEE Intelligent Systems*, *forthcoming*, 2007.

[17] Huang, Z., Zeng, D. and Chen, H., A link analysis approach to recommendation with sparse data. in *Americas Conference on Information Systems*, (New York, NY, 2004), 1997-2005.

[18] Jin, R. and Si, L., A Bayesian approach toward active learning for collaborative filtering. in *Twentieth Conference on Uncertainty in Artificial Intelligence*, (Banff, Canada, 2004), 278-285.

[19] Kiefer, J. Optimal experimental designs. *Journal of the Royal Statistical Society, series B*, *21*. 272-304, 1959.

[20] Lewis, D. and Gale, W., Training text classifiers by uncertainty sampilng. in *International ACM Conference on Research and Development in Information Retrieval (SIGIR-94)*, (1994), 3-12.

[21] Lin, W., Alvarez, S.A. and Ruiz, C. Efficient adaptive-support association rule mining for recommender systems. *Data Mining and Knowledge Discovery*, *6*. 83-105, 2002.

[22] Linden, G., Smith, B. and York, J. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, *7* (1). 76-80, 2003.

[23] Pazzani, M. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, *13* (5). 393-408, 1999.

[24] Pennock, D.M., Horvitz, E., Lawrence, S. and Giles, C.L. Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach, *16th Conference on Uncertainty in Artificial Intelligence (UAI-2000)*, 2000, 473-480.

[25] Popescul, A., Ungar, L.H., Pennock, D.M. and Lavrence, S., Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. in *17'th Conference on Uncertainty in Artificial Intelligence (UAI 2001).* (2001), 437-444.

[26] Resnick, P., Iacovou, N., Suchak, M., Bergstorm, P. and Riedl, J., GroupLens: An open architecture for collaborative filtering of netnews. in *ACM Conference on Computer-Supported Cooperative Work*, (1994), 175-186.

[27] Robbins, H. Some aspects of the sequential design of experiments. *Bulletin American Mathematical Society*, *55*. 527-535, 1952.

[28] Roy, N. and McCallum, A., Toward optimal active learning through sampling estimation of error reduction. in *18th International Conference on Machine Learning*, (2001), Morgan Kaufmann, 441-448.

[29] Saar-Tsechansky, M. and Provost, F. Active sampling for class probability estimation and ranking. *Machine Learning*, *54*. 153-178, 2004.

[30] Saar-Tsechansky, M. and Provost, F. Decision-centric Active Learning of Binary-Outcome Models. *Information Systems Research*, *forthcoming*, 2007.

[31] Sarwar, B., Karypis, G., Konstan, J. and Riedl, J., Application of dimensionality reduction in recommender systems: a case study. in *WebKDD Workshop at the ACM SIGKKD*, (Boston, MA, 2000).

[32] Sarwar, B.M., Karypis, G., Konstan, J.A. and Riedl, J.T., Item-based collaborative filtering recommendation algorithms. in *Tenth International World Wide Web Conference*, (2001), 285-295.

[33] Schafer, J., Konstan, J. and Riedl, J. E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, *5* (1-2). 115-153, 2001.

[34] Si, L. and Jin, R., Flexible mixture model for collaborative filtering. in *Twentieth International Conference on Machine Learning*, (2003).

[35] Ungar, L.H. and Foster, D.P., A formal statistical approach to collaborative filtering. in *Conference on Automated Learning and Discovery (CONALD)*, (Pittsburgh, PA, 1998).

[36] Yu, K., Schwaighofer, A., Tresp, V., Xu, X. and Kriegel, H.-P. Probabilistic memory-based collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering*, *16* (1). 56-69, 2004.

[37] Zheng, Z. and Padmanabhan, B. Selectively acquiring Customer Information: A new data acquisition problem and an active learning-based solution. *Management Science*, *52* (5). 697-712, 2006.