Relevance meets Coverage: A Unified Framework to Generate Diversified Recommendations

LE WU[1], QI LIU[1], ENHONG CHEN [1], NICHOLAS JING YUAN[2], GUANGMING GUO[1], XING XIE[2], University of Science and Technology of China [1], Microsoft Research [2]

Collaborative Filtering (CF) models offer users personalized recommendations by measuring the *relevance* between the active user and each individual candidate item. Following this idea, user based collaborative filtering (UCF) usually selects the local popular items from the like-minded neighbor users. However, these traditional relevance based models only consider the individuals (i.e., each neighbor user and candidate item) separately during neighbor set selection and recommendation set generation, thus usually incurs highly similar recommendations that lack diversity. While many researchers have recognized the importance of diversified recommendations, the proposed solutions either needed additional semantic information of items or decreased accuracy in this process. In this paper, we describe how to generate both accurate and diversified recommendations from a new perspective. Along this line, we first introduce a simple measure of *coverage* that quantifies the usefulness of the whole set, i.e., the neighbor userset and the recommended itemset as a complete entity. Then, we propose a recommendation framework named *REC* that considers both traditional *RE*levance based scores and the new *C*overage measure based on UCF. Under REC, we further prove that the goals of maximizing relevance and coverage measures simultaneously in both the neighbor set selection step and the recommendation set generation step are NP-hard. Luckily, we can solve them effectively and efficiently by exploiting the inherent submodular property. Furthermore, we generalize the coverage notion and the REC framework from both a data perspective and an algorithm perspective. Finally, extensive experimental results on three real world datasets show that the REC based recommendation models can naturally generate more diversified recommendations without decreasing accuracy compared to some state-of-the-art models.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information Filtering*

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Collaborative Filtering, Coverage, Diversity, Personalized Recommendation

1. INTRODUCTION

Collaborative Filtering (CF) is a technique to offer users personalized recommendations based on the wisdom of crowds [Adomavicius and Tuzhilin 2005]. Specifically,

This research was partially supported by grants from the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61325010), the National High Technology Research and Development Program of China (Grant No. 2014AA015203), the Natural Science Foundation of China (Grant No. 61403358), the Fundamental Research Funds for the Central Universities of China (Grant No. WK0110000042) and the Anhui Provincial Natural Science Foundation (Grant No. 1408085QF110). Qi Liu gratefully acknowledges the support of the Youth Innovation Promotion Association, CAS.
Authors' addresses: L. Wu's email: wule@mail.ustc.edu.cn, Q. Liu's (corresponding author) email: qiliuql@ustc.edu.cn, E. Chen's email: cheneh@ustc.edu.cn, N. Yuan's email: nicholas.yuan@microsoft.com, G. Guo's email: guogg@mail.ustc.edu.cn, X. Xie's email: xing.xie@microsoft.com.

DOI:http://dx.doi.org/10.1145/0000000.0000000

given the historical behavior data (e.g., browsing history, click streams or item satisfaction expressed in ratings), CF usually recommends a top-N list of items that are most *relevant* to the target user's interests [Su and Khoshgoftaar 2009]. The devised models in this research area could generally be classified into two categories– matrix factorization models and neighborhood based models [Koren 2008]. In matrix factorization models, users and items are projected into the same low latent space and the relevance between each pair of them is compared directly in this space. For neighborhood based methods, e.g., User based Collaborative Filtering (UCF), relevance between the user and an unknown item is measured by the like-minded (relevant) neighbor users' opinions on this item.

In this way, these traditional CF models are successful at providing accurate recommendations that match some of the user's dominant interests. However, the recommendation set/list maybe monotonous (i.e., the recommended items are highly similar to each other) and it is hard to cover all of this user's interests [Zhang and Hurly 2009; Said et al. 2013]. For instance, suppose *Alice* is a fan of Harry Potter book series and she is also interested in the topic of Machine Learning. She read many books written by *J.K. Rowling* and another book of *Pattern Recognition and Machine Learning*. Using traditional UCF, *Alice*'s neighbors are all fans of *J.K. Rowling* as these neighbor users need to have a lot of overlapped liked items with *Alice*. However, her interests for machine learning would be neglected in this process. Similarly, the matrix factorization models are also good at preserving the principal components of *Alice*'s dominant interests while neglecting the minor ones [Koren et al. 2009]. Correspondingly, all the books in the recommendation set are those written by *J.K. Rowling* without any machine learning related reference. *Alice* will soon get frustrated by such homogeneous recommendations. Thus, we have reasons to argue, a practical recommender system should not only make *relevant* but also *diversified* recommendations to improve overall user satisfaction.

Indeed, the importance of diversified recommendations has been well recognized in the literature in the past few years. Some argued that accuracy related metrics are far from enough for measuring recommendation quality and diversity should be considered as an important ingredient to user satisfaction [McNee et al. 2006; Boim et al. 2011; Said et al. 2013]. Others devised various models to improve recommendation diversity. These models usually followed two steps. They first retrieved a larger candidate recommendation set by traditional CF algorithms and then re-ranked this candidate set by diversity-enhancing techniques. E.g., Ziegler et al. introduced item category information to access intra-list similarity of the recommendation set and presented a topic diversification approach to improve recommendation diversity [Ziegler et al. 2005]. A similar approach was proposed in [Hurley and Zhang 2011], where the final recommendation results for each user was formulated as a binary optimization problem with the trade-off between diversity and accuracy. In both approaches, the diversity was measured by the distance based metrics in this itemset. Nevertheless, these models relied on additional category information to access item diversity or resulted in accuracy reduction during this process, which limits the generality of these models when only the user-item preference data is available. Thus we ask, is it possible to generate both accurate and diversified recommendations that are applicable to common scenarios even when no semantic data of items is available?

To tackle this problem, in this paper, we look at recommendation diversification from a new perspective. We start from the widely used UCF model and analyze why it fails to generate diversified recommendations empirically and experimentally. In fact, during the two steps of UCF, i.e., the neighbor set selection and the recommendation set generation, the relevance based measures only consider individuals in each set separately without a holistic view to treat elements in this set as a complete entity. To

this end, we first introduce a simple notion of *coverage* that measures the usefulness of the whole set. We further design several variants of this measure to consider different factors when measuring the utility of the whole set. Specifically, we apply this notion in both stages of UCF and define what it means for neighbor users to cover users' interests and the recommended items to cover diversified neighbors. Then we propose a recommendation framework named *REC* that not only encourages traditional *RE*levance based score but also rewards the new *C*overage measure. Since we reward coverage of each user's interests by selecting a diversified neighbor set and each diversified neighbor better contributes to the final recommendations, it is natural that the REC framework brings more diversified recommendations. Furthermore, we generalize this coverage notion and introduce how to adapt it to various CF models, including item-based collaborative filtering and matrix factorization models. In summary, we make the following contributions:

— We introduce a notion of coverage that measures the utility of the whole set in the two steps of UCF, i.e., neighbor set selection and recommendation set generation. We also design several variants of this measure to consider some important factors when measuring the utility of the whole set.
— We propose a recommendation framework that encourages both traditional relevance measure and the new coverage measure based on UCF. We prove that the problem of maximizing both relevance and coverage is NP-hard and provide efficient algorithms by exploiting the submodular property.
— We generalize the proposed coverage measure and the whole recommendation framework from both a data perspective and an algorithm perspective. Thus our proposed measure and the overall framework can be easily adapted to different kinds of data and various traditional CF models.
— We evaluate our proposed models on three real world datasets. Experimental results show that all the proposed models that incorporate the coverage measure increase diversity results when compared to the traditional CF models. We also provide alternatives to further improve diversity with a small loss in accuracy. E.g., in Douban dataset, the improvement of diversity is 12.7% with only 3.3% loss with accuracy.

Organization: We discuss the motivations and problem definition in Section 2. In Section 3 we present the proposed REC model. Then the generalization from both the data perspective and the algorithm perspective is discussed in Section 4. We conduct extensive experiments in Section 5, followed by related work in Section 6. Finally, we offer conclusions and future work in the last section.

Table I. Mathematical Notations

Notations	Description
U, V	userset, itemset in the recommender system
u, u'	users in the userset, u (u') $\in U$
v, v'	items in the itemset, v (v') $\in V$
L_u	items that user u likes, $L_u \subseteq V$
E_v	users that expressed likeness for item v, $E_v \subseteq U$
N_u	items that user u hasn't shown likeness
S_u	the selected neighbor set for user u
T_u	top-N recommendations for u ($\|T_u\|$=N, $T_u \subseteq N_u$)

2. PROBLEM DEFINITION AND PRELIMINARIES

In this section, we start from the UCF model, which assumes that an active user prefers the items that are locally popular among the like-minded neighbor users. This model has been widely studied in CF, and shows simplicity, robustness

as well as superiority in explaining the *Word of Mouth* phenomenon for decision making [Herlocker et al. 1999; Sarwar et al. 2000a; Su and Khoshgoftaar 2009; Said et al. 2013; tak]. Then we present a preliminary empirical and experimental analysis to show the insufficiency of UCF in top-N recommendation, which motivates our research. Specifically, for each user u, given a set of items L_u liked by her, top-N recommendation aims at retrieving a set of items T_u to u such that $|T_u| = N$ and $L_u \cap T_u = \emptyset$. For ease of explanation, Table I lists some notations used in this paper.

In UCF, there are usually two steps for generating recommendations for each user u: the neighbor set selection and the recommendation set generation. That is, it first selects a set of most similar (relevant) users as neighbors for u, and then measures the relevance between her and candidate items based on the neighbors' past choices. The commonly used assumptions for these two steps are [Herlocker et al. 1999; Sarwar et al. 2000a]:

— `Neighbor Set Selection`: If u' and u liked many items in common in the past, u' could be a neighbor of u;
— `Recommendation Set Generation`: If most neighbors in the neighbor set liked item v, then v is probably in the top-N recommendation set.

In the neighbor set selection step, various relevance based measures have been proposed. Since the focus of this paper is not to devise more sophisticated means to calculate user similarity, we choose the Jaccard measure, which performs well for the binary preference data [Das et al. 2007]:

$$sim(u, u') = |L_u \cap L_{u'}|/|L_u \cup L_{u'}|. \tag{1}$$

After that, the neighbor set S_u is generated by selecting top K users with the largest similarities. Then in the recommendation set generation step, the predicted likeness of user u to item v, denoted as $\hat{r}(u, v)$, is calculated by:

$$\hat{r}(u, v) = \sum_{v \in N_u, u' \in S_u} I(u', v)/|S_u|, \tag{2}$$

where N_u are the items that user u hasn't shown likeness ($L_u \cup N_u = V$ and $L_u \cap N_u = \emptyset$). $I(u', v)$ equals 1 if $v \in L_{u'}$ and 0 otherwise. Here, $\hat{r}(u, v)$ can be seen as measuring the relevance between user u and item v indirectly based on neighbors' past choices. Typically, the recommendation set is generated by sorting $\hat{r}(u, v)$s in a descending order. That is, the more popular an item is among the neighbor users, the more likely it would be recommended to the target user.

As can be seen, in both steps of UCF, we need to generate a set of elements, i.e., neighbors and recommendations respectively. However, using relevance based measures in traditional UCF, individuals in each set are considered separately without a holistic view to be treated as a whole [McNee et al. 2006]. Specifically, in the neighbor set selection period, only considering the relevance between the active user and each candidate neighbor separately would lead to incomplete coverage of her interests as some of her minority interests may not be covered by any neighbor. The incomplete coverage of users' interests in the neighbor set selection would further influence the recommendation period, making the final recommendation set contains no items that are similar to the uncovered interests of the user; Likewise, in the recommendation set generation step, only recommending the most popular items among neighbors would lead to incomplete coverage of neighbors. That is, even if we have selected a diversified neighbor set, each of which covered different interests of u, the final recommendations would be monotonous if some neighbors contribute nothing.

For further validation, we conduct preliminary experiments on three real-world datasets. These datasets are from different domains and all contain user-item preference data (see Table III in Section 5 for detailed statistics). We select neighbors of each

user based on Eq. (1) and calculate the percentage of her liked items that have been covered by the associated neighbor set (Eq. (12)). Ideally, the coverage score equals 1, implying each user's interests are perfectly covered by her neighbors. However, this score is far from satisfactory as shown in Table II, with 82.13% for MovieLens and 61.35% for Douban, thus many of users' interests are neglected since no neighbor user has preferences for these uncovered items. What's worse, the uncovered items are less popular than the covered ones, implying these unpopular items are harder to be covered in the traditional neighbor set selection period. Notice that these unpopular items are more valuable in distinguishing the active user's interests from other users than those items that are liked by the crowds [Yin et al. 2012]. In this table, the popularity of an item is proportional to users that liked it: $pop(v) = \frac{log|E_v|}{log|U|}$.

Table II. Average Coverage Score of UCF ($|S_u| = 10, |T_u| = 5$)

Dataset	Interest coverage score	Popularity of covered items	Popularity of uncovered items	Neighbor coverage score
MovieLens	82.13%	0.6911	0.5216	93.94%
Douban	61.35%	0.5451	0.4419	83.10%
Ihou	91.90%	0.6792	0.4876	65.17%

We further calculate the neighbor coverage score (i.e., the percentage of neighbors that contribute to the final recommendations, Eq. (14)) of the top-5 recommendation results for UCF, which are shown in the last column of Table II. We observe that nearly 20% of the neighbors contribute nothing for the final recommendations in Douban. For the Ihou data, even though 91.9% of users' interests have been covered in the neighbor set selection, there are still one third of the selected neighbors that do not contribute to recommendation results. This phenomenon implies that even if we selected a diversified neighbor set and covered most interests of u, we also need to modify the recommendation set generation step in UCF to get diversified recommendations. As only recommending the local popular items would lead to the incomplete coverage of neighbors, which also results in the monotonous recommendations for the active user.

In summary, there are some inherent insufficiency of traditional UCF. The fundamental reason is that: The relevance based metrics (similarity and predicted likeness) in the two UCF steps are designed to treat individuals separately, and this may not perform the best with respect to the whole set. Please note that similar problems also exist in other traditional CF models as shown in the Introduction section. In this paper we first focus on UCF for better illustration and then would discuss how to solve this problem for other traditional CF models. Thus, we argue that for both selecting the set of neighbors and the set of recommendations, the measurements should also consider the usefulness of the set as a whole rather than a collection of individuals. In this way, we set up the new goals in these two steps for top-N recommendation:

— `Neighbor Set Selection`: It is ideal that each neighbor is similar with u and that the whole neighbor set covers as many of u's interests as possible.
— `Recommendation Set Generation`: It is ideal that each recommendation is locally popular among neighbors and that the whole set of recommendations covers as many neighbors as possible.

3. THE PROPOSED FRAMEWORK

Based on the new goals mentioned above, we first introduce a metric of *coverage*, which naturally considers the utility of the whole set by encouraging diversity of individuals. This metric utilizes the historical preference data and it is applicable to common scenarios when no item content is available. We define what it means for neighbors

to cover the user's interests and for recommendations to cover the selected neighbors. Accordingly, we propose a two-stage framework named *REC* for recommendation. The commonality of these two stages is that, for both the neighbor set selection and the recommendation set generation, we formulate the new goals mentioned above as optimization problems with the dual objectives of maximizing relevance and coverage. Fig. 1 illustrates the overall framework of the REC model. Next we introduce these two stages in detail and show how the new goals can be achieved.

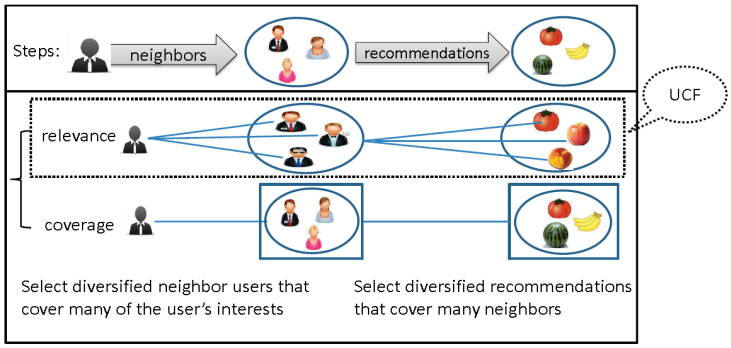


Fig. 1. The overall REC framework.

3.1. Neighbor Set Selection by Interest Coverage

For each active user, we wish to select a representative neighbor set that covers as many of her interests as possible. Here, the interests can be a collection of items that are liked by the active user or some higher level representations, e.g., the categories of the items or the hidden topics learned from the item descriptions. Since it is time and labor consuming to get the exact categories of each item, to improve the generality, we simply refer to the collection of items that the user likes as her interests.

As shown previously, when selecting the neighbor set for user u, we should not only ensure each neighbor has a relatively large relevance value with u. Also, this neighbor set should cover as many of u's interests as possible. In summary, two important measures are *Interest Relevance* and *Interest Coverage*. For each user u, we model the neighbor set selection quality as follows:

$$F(S_u, u) = \alpha \cdot IRel(S_u, u) + (1 - \alpha) \cdot ICov(S_u, u), \tag{3}$$

where $IRel(S_u, u)$ measures the relevance (similarity) between user u and the selected neighbor set S_u. $ICov(S_u, u)$ rewards the interest coverage score by S_u, and α $(0 \leq \alpha \leq 1)$ is a trade-off coefficient balancing these two measures.

The relevance score can be interpreted as the group similarity of the neighbor set S_u to user u. Since the similarity value of each candidate to the active user does not interfere with other candidates, a simple approach is to sum each candidate's similarity with u (Eq. (1)) in this group, thus:

$$IRel(S_u, u) = \sum_{u' \in S} sim(u', u). \tag{4}$$

As discussed before, this score is widely used as the only measure for the neighbor set selection in traditional CF models [Herlocker et al. 1999; Sarwar et al. 2000a].

For the interest coverage score, it measures the percentage of the user's interests that have been covered by the neighbors. A simple form of this measure is defined as:

$$ICov(S_u, u) = \frac{\sum_{v \in L_u} cov(S_u, v)}{\sum_{v \in L_u} 1} = \frac{\sum_{v \in L_u} \mathbf{1}[\exists u', u' \in S_u \& v \in L_{u'}]}{|L_u|}, \tag{5}$$

where $cov(S_u, v)$ measures the coverage score of item v by neighbor set S_u. A simple idea is to define it as an indicator function that equals 1 if v appears in one of the neighbors' interests and 0 otherwise. And the interest coverage score of u is calculated by averaging all coverage scores of her interests. Though intuitive, this naive definition suffers from two drawbacks:

— *Interest Importance*: The active user's interests for all items are treated equally above, thus we can not distinguish the importance of certain items. E.g., the less popular items are more important to reflect the user's interests than popular ones.
— *Incremental Interest Coverage*: The above interest coverage function is too strict, since if one neighbor has covered an interest v of user u ($v \in L_u$), the coverage score for $cov(S_u, v)$ would never gain even if other neighbors would cover v later. This strong condition would result in the imbalance of covered times between different interests, causing some of u's interests being covered only once or twice. Consider an extreme case, there exists one virtual user *Bob* who liked all items in this system. After *Bob* has been chosen in the neighbor set for user u, the coverage score is now maximized and we only need to select other neighbors from candidate users who have the largest similarities as traditional UCF does. Thus this naive coverage measure is frail and we have to find an incremental interest coverage function as an alternative.

We now address each of these issues in detail. For better measuring the importance of each interest, we can simply add a weight to each item:

$$ICov(S_u, u) = \frac{\sum_{v \in L_u} w_v \times cov(S_u, v)}{\sum_{v \in L_u} w_v} = \frac{\sum_{v \in L_u} w_v \times \mathbf{1}[\exists u', u' \in S_u \& v \in L_{u'}]}{\sum_{v \in L_u} w_v}. \quad (6)$$

As to the incremental interest coverage problem, a better idea is to assign a soft function $cov(S_u, v)$ to replace the above strong condition. There are some intuitions in designing this soft function. First, when no neighbor is selected yet, $cov(\emptyset, v)$ equals 0. Second, the interest coverage score is non-decreasing as we add more neighbors to the neighbor set: $cov(S_u \cup u', v) \geq cov(S_u, v)$. That is, when more neighbors are added, there are higher chances for item v to be covered. Last but not least, the marginal gain of the coverage score decreases as we add more neighbors to u: $\forall A \subseteq B \subseteq U \setminus u', cov(A + u', v) - cov(A, v) \geq cov(B + u', v) - cov(B, v)$. This intuition refers to that each additional time we see a new neighbor covered item v, the reward is decreased with the number of neighbors that have already covered this item. Since for a larger neighbor set B, the covered times of item v must be no less than those of a subset set A. Fig. 2 draws an ideal shape of the coverage score function of a particular item, where the left figure depicts the strong coverage function and the right the desired incremental function. With the increasing of the occurrences in the neighbor set, the incremental coverage function value would gain slowly while the marginal gain decreases at the same time (the dashed line). In contrast ,the strong coverage function reaches 1 when this item has been covered by the first time and would never gain any more, even if it would be covered by other neighbors later. In fact, the phenomenon of decreased marginal gain is a well-understood fundamental principle of economics, which is termed as "Law of Diminishing Returns" [Baumol and Blinder 2011]. This phenomenon happens everywhere in our everyday life. For example, *Alice* will be thrilled if her parents buy an iPhone 5 as her first smartphone. However, when given another iPhone 5 later, her satisfaction would decrease compared to the satisfaction from owing the first iPhone. Similarly, in the interest coverage function, the marginal gain of an item would decrease as it has been covered by more neighbors.

Actually, there are various functions satisfying the incremental coverage requirements (right part of Fig. 2). E.g., it is easy to truncate part of the sigmoid functions or use simple linear transformations to meet the requirements. Here we use a simple

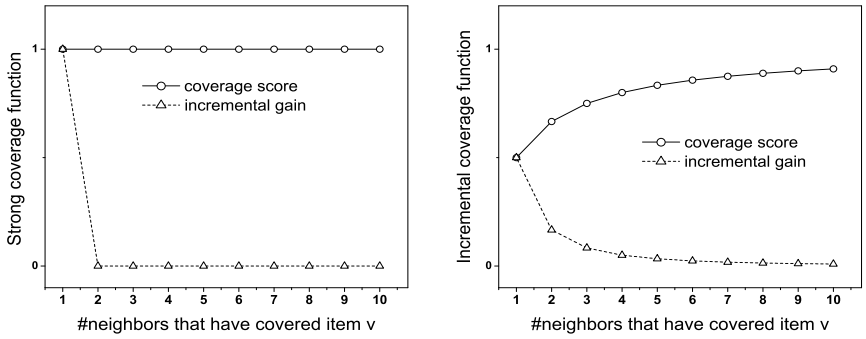


Fig. 2. Interest coverage function comparison, where the left figure shows the strong coverage score and the right depicts the desired incremental coverage score.

incremental coverage function, which is defined as:

$$cov(S_u, v) = \frac{cnt(S_u, v)}{cnt(S_u, v) + 1}, \tag{7}$$

where $cnt(S_u, v) = \sum_{u' \in S_u} \mathbf{1}[v \in L_{u'}]$ denotes how many times item v has been covered by the neighbors. In this coverage function, it reaches 0.5 as only one neighbor covers this item and 0.9 when it has been covered by 10 neighbors. After that, the coverage score increases slowly to the limit of 1.

Finally, we propose the following interest coverage function that combines importance (Eq. (6)) and incremental coverage (Eq. (7)):

$$ICov(S_u, u) = \frac{\sum_{v \in L_u} w_v \times cov(S_u, v)}{\sum_{v \in L_u} w_v} = \frac{\sum_{v \in L_u} w_v \times \frac{cnt(S_u, v)}{cnt(S_u, v)+1}}{\sum_{v \in L_u} w_v}. \tag{8}$$

Since in the real world, it is harder to cover the unpopular items for a traditional CF as shown in Table II, we simply set $w_v = 1 - pop(v) = 1 - \frac{log|E_v|}{log|U|}$. That is, the more popular an item v, the smaller the weight of it. This definition is intuitive, e.g., suppose u liked an item v that is very popular among the crowds, then v is not informative compared to those unpopular items liked by u.

According to Eq.(8), the interest coverage score ranges in $[0, 1)$. To make the relevance measure and coverage measure comparable,we normalize the relevance score as follows: $Rel(S_u, u) = \frac{Rel(S_u, u)}{C}$,where C is a constant that sums up the top K users that have the largest similarity with u. In this paper, the similarity between a pair of users is calculated by Eq.(1). Then the value of the relevance score function also ranges from 0 to 1. It reaches 1 when $\alpha = 1$, meaning only considering the relevance measure in THE neighbor set selection. At this time, our model degrades to the traditional neighbor set generation method in UCF.

Note that these two measures characterize different aspects when selecting the neighbor set. Using the relevance score, each neighbor in a set S should have a large similarity with user u. Meanwhile, the neighbors are encouraged to be diversified, such that they can cover most of the user's interests. In summary, these two measures can be seen as judging the individual element and the whole group of the neighbor set respectively. Then the neighbor set selection can be reformulated as the Interest Coverage Maximization Problem:

Interest Coverage (IntCov) Maximization Problem: Given a set of items L_u liked by user u, identify a K neighbor set S_u that maximizes the quality function:

$$\max_{S_u \in U, |S_u| = K} F(S_u, u). \tag{9}$$

3.1.1. Properties. The IntCov Problem has several interesting properties. First, it is NP-hard as proven in the following theorem.

THEOREM 3.1. *The IntCov problem is NP-hard.*

PROOF. Consider an instance of the NP-hard *Weighted Maximum Coverage* problem, defined by a collection of sets $A = A_1, A_2, ..., A_m$ over a domain of elements $E = e_1, e_2, ..., e_n$ and each element is associated with a weight w_v. Given a number K, the task is to find a subset $A' \in A$ such that the total weights of covered elements is maximized [Hochba 1997]. Next, we show that our IntCov can be reduced to this problem under a special situation. Consider when $\alpha = 0$, i.e., we only need to maximize the interest coverage score of the neighbor set. For user u, we define a corresponding collection of sets $A = L_u \cap L_{u_1}, L_u \cap L_{u_2}, ..., L_u \cap L_{u_{|U|-1}}$. The weight of each item corresponds to $w_v \times \frac{cnt(v)}{cnt(v)+1}$, where $cnt(v)$ denotes how many times item v has been covered by the selected K sets. Under this situation, maximizing the weighted coverage score is equal to finding a K element set in A and our problem is reduced to the NP-hard Weighted Maximum Coverage Problem. □

Meanwhile, $F(S_u, u)$ has some other important properties. First, $F(\emptyset, u)=0$, i.e., the quality score is zero when we do not select any neighbor for u. Second, $F(B, u) \geq F(A, u)$ if $A \subseteq B$, as both the relevance and the coverage scores are non-decreasing. Third, it is submodular which satisfies the property of diminishing returns: $\forall A \subseteq B \subseteq U \setminus u', F(A + u') - F(A) \geq F(B + u') - F(B)$. In fact, we have already designed the coverage score of a particular interest of u, i.e., $cov(S_u, v)$ as a submodular function. Next, we will prove the submodularity of the neighbor set quality function F.

THEOREM 3.2. *The neighbor set quality function F is submodular.*

PROOF. First we claim that both $IRel(S_u, u)$ and $ICov(S_u, u)$ are submodular functions. For any $\forall A \subseteq B \subseteq U \setminus u'$, $IRel(A \cup u', u) - IRel(A, u) = \frac{sim(u,u')}{C} = IRel(B \cup u', u) - IRel(B, u)$, where C is a constant that sums up the top K users with largest similarity scores with u. Thus $IRel(A \cup u', u) - IRel(A, u) \geq IRel(B \cup u', u) - IRel(B, u)$ holds true all the time. We conclude that $IRel(S, u)$ is submodular.

For each $v \in L_u$, we have $cov(A + u', v) - cov(A, v) = \frac{cnt(A+u',v)-cnt(A,v)}{[cnt(A+u',v)+1] \times [cnt(A,v)+1]}$ and $cov(B + u', v) - cov(B, v)) = \frac{cnt(B+u',v)-cnt(B,v)}{[cnt(B+u',v)+1] \star [cnt(B,v)+1]}$. As $A \subseteq B, cnt(A + u', v) - cnt(A, v) \geq cnt(B + u', v) - cnt(B, v)$ and $[cnt(A + u', v) + 1] \times [cnt(A, v) + 1] \leq [cnt(B + u', v) + 1] \times [cnt(B, v) + 1]$, we have $cov(A + u', v) - cov(A, v) \geq cov(B + u', v) - cov(B, v)$, which is the defining quality of submodularity. Finally, as submodularity is closed under non-negative linear combinations, the interest coverage function $ICov$ (defined in Eq. (8)) is submodular. Thus, function F that linearly combines $IRel$ and $ICov$ satisfies submodularity. □

In conclusion, though maximizing the objective function $F(S_u, u)$ is NP-hard, it is a *non-negative monotone submodular function*. Researchers have already shown that a simple heuristic greedy algorithm can guarantee high performance for maximizing this kind of functions [Nemhauser et al. 1978]. Furthermore, there are no other efficient algorithms which can generate better performance guarantee unless P=NP [Feige 1998]. Specifically, this greedy algorithm starts with an empty set and incrementally constructs the required set S in K steps. Each time it adds a new element u' to this set S_u which maximizes the marginal gain: $u' = argmax_{u' \subseteq U \setminus S_u} F(S_u \cup u', u) - F(S_u, u)$. Following is the well-known performance guarantee.

THEOREM 3.3. *[Nemhauser et al. 1978; Cornuejols et al. 1977] For any* non-negative monotone submodular function *F, let $S^\star$ be the K element set with the best*

performance, and S the same size set obtained by greedy algorithm, which selects an element with maximum marginal gain each time, then $F(S) \geq (1 - \frac{1}{e})F(S^\star)$.

3.1.2. Scaling Up IntCov Algorithm. For each user u, the time complexity of IntCov is $O(KM)$ with naive greedy evaluations, where K denotes the neighbor set size S and M the number of total users. Inspired by [Leskovec et al. 2007], we can further exploit the submodular property to reduce time complexity for finding candidate neighbors. In submodular functions, the marginal gain of a new node shrinks as the set becomes larger, i.e., for each candidate neighbor u', the marginal gain in the current iteration can not be larger than that in the previous iteration. Thus, instead of recomputing the marginal gain of each candidate in every iteration, we do *lazy evaluations*. We keep a sorted list recording the marginal gain for each candidate neighbor of u. In every iteration, we just need to look at the list from the top node, if it is valid then move it to the neighbor set and go directly to the next step, otherwise recompute the gain and insert it into the list. This lazy forward method only needs to scan all candidate neighbors in the first step. In our experiments, from step 2 to K, the recomputation of the top element in the sorted list will lead to a new value that is not much smaller than the previous one. Usually simply calculating the top several elements will help us find a newly added neighbor, and thus save computation. The whole algorithm for IntCov is described in Algorithm 1.

Algorithm 1 Overall flowchart of IntCov Algorithm

Input: user u, neighbor size K
Output: u's neighbor set S_u

Initialize $S_u^0 = \emptyset, C = U \setminus u, SL = \emptyset$;
for each $u' \in C$ **do**
  compute $\delta_{S_k}(u') = F(u', u)$;
  $SL.insert([u', \delta_{S_k}(u')])$ with the descending order of $\delta_{S_k}(u')$
end for
$[u', \delta(u')]$=SL.pop(); $S_u^1 = S_u^0 \cup u'$
for $k = 2$; $k <= K$; $k++$ **do**
  $[u', \delta(u')]$= SL.pop()
  recompute the marginal gain of u': $\delta_{S_k}(u') = F(S_u^{(k-1)} \cup u', u) - F(S_u^{(k-1)}, u)$
  while $\delta_{S_k}(u') < SL.top.values()$ **do**
    SL.insert($[u', \delta_{S_k}(u')]$)
    $[u', \delta(u')]$= SL.pop()
    recompute the marginal gain of u': $\delta_{S_k}(u')$
  end while
  $S_u^k = S_u^{(k-1)} \cup u'$
end for
Return S_u^k.

3.2. Recommendation Set Generation by Neighbor Coverage

Following the proposed neighbor set selection criteria, a natural idea for recommendation set generation is that, we should not only recommend items with large predicted likeness scores by neighbors (*Neighbor Relevance*), but also ensure each recommended item better covers different neighbors (*Neighbor Coverage*), such that each diversified neighbor better contributes to the final recommendations. In a word, for each user u, we model the recommendation quality $G(T_u, u)$ as follows:

$$G(T_u, u) = \beta \cdot NRel(T_u, u) + (1 - \beta) \cdot NCov(T_u, u), \tag{10}$$

where the relevance score $NRel(T_u, u)$ measures the predicted likeness of the whole recommendation set T_u to user u, which is the sum of each single recommendation's predicted likeness (i.e., Eq. (2)):

$$NRel(T_u, u) = \sum_{v \in T_u} \hat{r}(u, v). \tag{11}$$

The neighbor coverage score $NCov$ measures the percentage of the neighbors that have contributed to the final recommendations T_u. It can be simply defined as follows:

$$NCov(T_u, u) = \frac{\sum_{u' \in S_u} cov(T_u, u')}{\sum_{u' \in S_u} 1} = \frac{\sum_{U' \in S_u} \mathbf{1}[\exists v, v \in T_u \& v \in L_{u'}]}{|S_u|}, \tag{12}$$

where $cov(T_u, u')$ is a binary valued coverage function that equals 1 if neighbor u' has contributed to the final recommendation and 0 otherwise. This simple method would not differentiate neighbors that contribute many recommendations and that contribute only once or twice. Since if a neighbor has contributed a recommendation v ($v \in T_u$), the coverage score of the neighbor $cov(T_u, u')$ turns from 0 to 1 and that score would never gain even if u' has covered other recommendations later. Inspired by the idea of "Incremental Interest Coverage", we can similarly adapt the strong coverage function to the *Incremental Neighbor Coverage*:

$$cov(T_u, u') = \frac{cnt(T_u, u')}{cnt(T_u, u') + 1}, \tag{13}$$

where $cnt(T_u, u')$ denotes how many recommended items have been covered by the neighbor u'. Combining Eq. (12) and Eq. (13), we get the final incremental neighbor coverage function:

$$NCov(T_u, u) = \frac{\sum_{u' \in S_u} cov(T_u, u')}{\sum_{u' \in S_u} 1} = \frac{\sum_{u' \in S_u} \frac{cnt(T_u, u')}{cnt(T_u, u')+1}}{\sum_{u' \in S_u} 1}. \tag{14}$$

Thus the neighbor coverage score ranges from $[0, 1)$, and it gets larger values as the neighbors cover more items of the final recommendations. β is a coefficient balancing $NRel$ and $NCov$. To make the neighbor relevance score and the neighbor coverage score comparable, we also normalize the predicted likeness score as follows: $NRel(T_u, u) = \frac{NRel(T_u, u)}{C}$, where C is a constant that sums up the top N items with the highest predicted ratings of u. Then the value of the likeness score function also ranges from 0 to 1.

Similarly, these two measures characterize different aspects of selecting the top-N recommendation set. Using $NRel$, each item in set T_u should have a relatively high predicted ranking. Using $NCov$, the recommended items should be diversified for covering different neighbors. In summary, they can be seen as judging the individual element and the whole group of the recommendation set respectively. Then the recommendation set generation step, can be reformulated as the neighbor set coverage problem:

Neighbor set Coverage (NeiCov) Problem: Given the neighbor set S_u for each user, identify a set of N items that maximize the score:

$$\max_{T_u, |T_u|=N} G(T_u, u). \tag{15}$$

The NeiCov problem shares the same property as the IntCov problem, which is a non-decreasing submodular function. Users could refer to Section 3.1.1 to get similar results. Thus we can use the same lazy forward greedy algorithm and scaling up technique (Algorithm 1) to solve NeiCov. In summary, by pushing the coverage measure into the two steps of IntCov and NeiCov of the REC framework, the diversity of recommendations is reached naturally.

4. GENERALIZATION AND DISCUSSION

In the previous section, we detailed the way of incorporating the coverage measure into the two UCF steps to generate diversified recommendations. The proposed REC framework is described mainly on a binary user-item preference dataset without any item content. However, when more kinds of data are available, is it easy to incorporate the content information, e.g., the categories of these items, into the proposed

framework? In addition, though UCF is used by earlier recommender systems, the data sparsity and scalability issues have limited the widespread usage of this algorithm in today's real world recommender systems. With this in mind, can the coverage measure and the REC framework also be applied to more advanced CF algorithms, such as item based collaborative filtering [Deshpande and Karypis 2004] and matrix factorization models [Koren et al. 2009]? In this section, we would like to answer these questions. We would explore the possibility of generalizing the proposed coverage measure and the whole recommendation framework from both a data perspective and an algorithm perspective.

4.1. Generalization of non-binary data and content information

4.1.1. Generalization of non-binary data. Recall that in the two stages of the REC framework, we formulated the goals in both steps as optimization problems with the dual objectives of maximizing relevance and coverage. The input of the REC framework is the binary preference values of users, with 1 standing for likeness and 0 for dislike. For some websites, users' preferences for items are expressed as explicit ratings, where the larger values denote higher appreciation. For ease of explanation, we use r_{ij} to denote user i's explicit rating for item j. In fact, these explicit ratings can be incorporated into both the relevance measure and the coverage measure in the REC framework.

Specifically, in the neighbor set selection period, instead of relying on Jaccard similarity to calculate the *Interest Relevance* (Eq. (4))for a neighbor userset, we can devise more measures to calculate the similarity between a pair of users, e.g., the cosine similarity measure:

$$sim(u,u') = cos(\vec{r(u,)}, \vec{r(u',)}) = \frac{< \vec{r(u,)}, \vec{r(u',)} >}{||\vec{r(u,)}||_F^2 * ||\vec{r(u',)}||_F^2}, \tag{16}$$

where $<,>$ is the inner product of two vectors. $\vec{r(u,)}$ denotes a V dimensional vector of user u's rating records and the j's element of this vector is r_{uj}. Also, we can incorporate the detailed rating values into the *Interest Coverage* (Eq. (8)) measure in the neighbor set selection. With the interest coverage measure, interest importance is a necessary factor. One possible solution is to embed the detailed rating values into the interest importance factor of the interest coverage measure. For each user u, if item v has a larger rating value than v' $(r(u,v) > r(u,v'))$, then interest v must weigh more than v' $(w_v > w_{v'})$ in u's opinion. E.g., for user u, a simple intuition is to set $w_v \propto r(u,v)$. Note that under this circumstance, the interest importance w_v is personalized and varies among people. In summary, the detailed rating values can be incorporated into both measures during the neighbor set selection period.

Similarly, in the recommendation set generation period, we can extend the detailed rating values into the *Neighbor Relevance* measure (Eq.(11)). Specifically, the predicted likeness $\hat{r}_{u,v}$ of user u to item v can be calculated as:

$$\hat{r}(u,v) = \sum_{u' \in S_u} r(u',v)/|S_u|, \tag{17}$$

where $\hat{r}(u,v)$ equals 0 if user u does not rate item v and the real rating value if u rates v. Thus, the REC framework can be easily adapted to non-binary data in both the neighbor set selection and recommendation set generation period.

4.1.2. Generalization of content information. In the previous section, we assumed that no item information was available and each user's interests were expressed as the set of items that she likes. In fact, we can also characterize each user's interests with some higher representations, e.g., the categories of the items. In the real world, when the categorization of items is available (such as genres and tags), we can group items into categories and each user's interests are mapped to the categories based on the items

that she likes. Then in the interest coverage measure of the neighbor set selection period, instead of calculating how many liked items of u are covered by the neighbor set, we measure how many categories liked by user u are covered by the neighbor users. Under this circumstance, the interest coverage score would increase if more of the user's liked categories are covered by neighbors. After the neighbor set selection, since the interest coverage measure does not interfere with the neighbor coverage measure, the proposed NeiCov algorithm for recommendation set generation in the REC framework can be applied directly to generate more diversified recommendations. Thus it is easy to generalize the coverage measure and the REC framework by characterizing users' interests with the available item information.

4.2. Generalization of other CF models

In traditional CF, there are usually two kinds of approaches: neighborhood based models and *M*atrix *F*actorization models (MF) [Koren 2008]. Neighborhood based models can be further divided into user-based (UCF) and item-based (ICF) collaborative filtering. We previously detailed how to apply the coverage notion and the REC framework to UCF. However, the scalability and data sparsity issues limit the applicability of UCF in today's real world recommender systems. Next we would like to discuss several alternatives to generalize this coverage measure and the whole REC framework to ICF and MF models.

4.2.1. Generalization of ICF. ICF assumes that a user would probably prefer items that were liked by her in the past. Thus, instead of finding similar users in UCF, ICF first selects the neighbor itemset of each item and then makes recommendations based on these item neighbors. Here, we present a simple idea to extend the coverage measure into ICF. In the first step, similar to ICF, we calculate the neighbors of each item based on the user-item preference matrix. That is, for each item v, the Jaccard similarity between it and v' is calculated as:

$$sim(v, v') = \frac{|E_v \cap E_{v'}|}{|E_v \cup E_{v'}|}, \tag{18}$$

where E_v denotes the users that like v. Then the neighbor itemset of v, denoted as S_v, is generated by selecting top K items that have the largest similarities to v. After the neighbor item selection period, for each user u, the predicted likeness of user u to item v, denoted as $\hat{r}(u, v)$, is calculated by averaging the target user's preferences over the item neighbors:

$$\hat{r}(u, v) = \sum_{v \in N_u, v' \in S_v} I(u, v')/|S_v|, \tag{19}$$

where $I(u, v')$ is equal to 1 if $v' \in L_u$ and 0 otherwise.

However, in contrast to ICF, for each user u, instead of selecting the top ranked candidate items that have the largest predicted ratings for recommendation, we consider the union of items that are neighbors of each item liked by u as the candidate recommendation set. That is, the candidate recommendation set C_u for each user u can be denoted as $C_u = \cup_{v \in L_u} S_v$, where L_u are the items that u likes and S_v is the selected neighbor itemset of item v by Eq.(18). Thus, u's interests are fully covered by the candidate recommendation set after the first step of neighbor itemset selection. During the second step of recommendation set generation, for each user u, our goal is to generate a recommendation set T_u from the candidate recommendation set C_u. Instead of recommending the most relevant items that have the largest predicted ratings (Eq.(19)), we incorporate the neighbor coverage measure to ensure that the recommendations cover more of those items that are neighbors of u's liked items. A simple form to define the neighbor coverage score for recommendation set generation is:

$$NCov(T_u, u) = \frac{\sum_{v \in L_u} cov(T_u, v)}{|L_u|} = \frac{\sum_{v \in L_u} \mathbf{1}[\exists v', v' \in T_u \& v' \in S_v]}{|L_u|}, \tag{20}$$

where $T_u \subseteq C_u$ is the final recommendation set for u, $cov(T_u, v)$ is an indicator function that is equal to 1 if at least one recommended item in the recommendation set T_u is a neighbor item of v and 0 otherwise. This neighbor coverage measure encourages more of the items that are similar to a user's interests would contribute to the final recommendations. By combing both the neighbor relevance (Eq.(19)) measure and the neighbor coverage (Eq.(20)) measure, we formulate a similar recommendation quality function that balances maximizing these two measures, which is introduced in Eq.(10) of Section 3.2.

In summary, to incorporate the coverage measure and the REC framework into the ICF, we could enrich the candidate recommendation set C_u for each user u after the neighbor selection step, ensuring that the user's interests to be fully covered by the candidate recommendation set. Then in the recommendation generation step, the final recommendation set T_u for each target user u better covers more of the user's interests, encouraging diversified recommendations compared to ICF.

4.2.2. Generalization of MF. To solve the data sparseness problem and improve scalability, various MF models have been proposed. A basic idea of MF is that attributes or preferences of users are determined by a small number of factors, thus both users and items can be projected into a low latent space from the historical records [Mnih and Salakhutdinov 2007; Koren et al. 2009]. That is, given a user-item preference matrix $R^{|U|*|V|}$, MF models try to infer the low latent D dimensional representations of users $U^{|U|*D}$ and items $V^{|V|*D}$ as accurate as possible. Various models have been proposed to solve this problem based on the data type of the user-item preference matrix R. E.g., probabilistic matrix factorization performs well if the preference matrix consists of ratings [Mnih and Salakhutdinov 2007] and Bayesian personalized ranking model is suitable for ranking evaluations with a binary preference matrix input [Rendle et al. 2009]. Since we focus on generalizing the REC framework to MF models, we assume that we have already got the low latent representations of users and items, i.e., U and V, from some state-of-the-art MF models (e.g., probabilistic matrix factorization). We'll introduce how to adapt the results to the REC framework. The basic idea is that, in the neighbor set selection period, instead of calculating the relevance scores by using the traditional measure that considers how many overlapped items are liked by a pair of users in UCF (Eq.(1)), we determine the similarity between a pair of users from the low latent representations of them as:

$$sim(u, u') = cos(U_u, U_{u'}) = \frac{< U_u, U_{u'} >}{||U_u||_F^2 * ||U_{u'}||_F^2}, \tag{21}$$

where U_u is the D dimensional low latent representation of user u from $U^{|U|*D}$. We adopt the same interest coverage measure as introduced in Eq.(8). Then we follow the similar neighbor set selection step as proposed in IntCov for diversified neighbor selection and further generate recommendations with NeiCov.

In all, the whole recommendation framework for the generalization of MF is still based on the two-stage REC framework that consists of neighbor set selection by IntCov and recommendation set generation by NeiCov. The only difference is that, instead of calculating the interest relevance measure based on the sparse preference data of users, we apply the low latent representations of users for better interest relevance measure calculation. Thus we can take advantage of the reduced latent space representations of users for better neighbor selection. Indeed, the dimension reduction technique showed good performance by some previous works when the data is extremely sparse [Liu et al. 2011; Zhang and Hurley 2009].

Actually, we should notice that another possible way to generalize the REC framework to MF is to extending the coverage measure directly to the low latent representations of users and items. However, as the coverage measure is mainly based on

binary inputs in this paper, we leave the problem of how to adapt it to any value type efficiently and effectively in our future research plan.

5. EXPERIMENTAL RESULTS

We evaluate the proposed REC framework on three real-world datasets. Specifically, we demonstrate: (1) the effectiveness of our models in covering users' interests and neighbors, (2) how the accuracy and diversity values change with the controlling parameters, (3) the efficiency of the proposed models and (4) the overall performance of our models compared with baselines. Though we mainly focus on the performance of the REC related models in this paper, we would also give the experimental results of generalizing the coverage measure to other CF models as introduced in Section 4 when performing overall comparison.

5.1. Data Description

We use the following three datasets for our experiments:

MovieLens-1M dataset: MovieLens-1M is a public available dataset collected by the GroupLens research[1]. It contains 1 million movie rating records with ratings ranging from 1 to 5. There are 18 movie genres and on average each movie belongs to 1.65 genres. Since we focus on recommending a set of items that the target user probably likes, similar as other works [Liu et al. 2012; Adomavicius and Kwon 2012], we select ratings that are larger than threshold 3 as the records that are liked by users.

Douban-Book dataset: Douban[2] is a popular social-based website in China that allows users to rate movies, books and music. In this experiment, we crawled a large part of users' rating history of books through its public APIs. This initial dataset contains 23,000 users' book ratings ranging from 1 to 5 star scale. We first take similar filtering steps as described above and then filter out the users that have less than 5 rating records. There are 7 categories of these books, namely literature, popularity, culture, life, business&economics, technology and others. Each book belongs to one of these categories.

Ihou dataset: Ihou[3] is a popular online karaoke website owned by iFlytek Co., Ltd, a leading provider of speech and language technology in China. Users are encouraged to sing songs in this platform and others can listen to these songs and express their feelings about the user's performance. This initial dataset is provided by Ihou and contains users' singing records during July 2011 to Sep 2012. To reduce noise, we only keep users and songs that appeared more than 5 times. This initial dataset has no category information and nearly all of these songs belong to the popular genre, we turn to the original singer as the category of each song [4]. We further filter out the singers that have less than 10 songs in this dataset and finally get 15 singers (categories) and 555 songs. Table III shows the details of each dataset.

Table III. General statistics of the three datasets

DataSet	Domain	Users	Items	Records	Avg items per user	Sparsity	Categories
MovieLens	Movie	6,038	3,533	575,281	95.28	97.30%	18
Douban	Book	17,840	36,226	1,000,039	56.05	*99.845%*	7
Ihou	Music	14,793	555	139,307	*9.42*	98.30%	15

[1]http://www.grouplens.org/node/73

[2]http://www.douban.com/

[3]http://ihou.com

[4]Usually music has quite a broad range of genres, e.g., classical, jazz. However, it requires professional training to sing these and most Chinese prefer to sing popular songs for leisure.

5.2. Evaluation Metrics and Baselines

5.2.1. Evaluations Metrics. We evaluate the proposed models from two aspects: accuracy and diversity. The detailed metrics are listed as follows:

Accuracy Metric: We use the widely adopted *Precision* for measuring the ranking accuracy of recommendation results in collaborative filtering. Specifically, given the recommendation set T_u for u, $Prec_u$ measures the fraction of recommendations that are really liked by the user:

$$Pre_u = \frac{|T_u \cap TT_u|}{|T_u|} = \frac{|T_u \cap TT_u|}{N},$$

where TT_u is the itemset that are liked by u in the test data.

Diversity Metric: As discussed, each user has unique ranges of interests, with some focus on a few topics while others may encompass a wide range of interest, thus the diversification level in the recommendation set should be adapted to u's own interest level. As there is no universal standard on how to measure the diversity level of each recommendation set, we borrow the category information of items for evaluation. For each user, the diversity level Div_u is measured as follows:

$$Div_u = \frac{|C(T_u) \cap C(L_u)|}{|C(L_u)|},$$

where $C(L_u)$ denotes the categories that u likes in the training data while $C(T_u) \cap C(L_u)$ are the categories that also appear in the recommendation set. Please note that to improve the generality of our model and make it applicable to common recommendation scenarios even when no item information is available, the category information is only used to measure the diversity level of each model in the evaluation process.

These two kinds of metrics characterize different aspects of recommendation quality. The accuracy metric tries to measure the *individual level* of recommendation results, assuming that the recommendation set is good if each element is accurate. However, even though each recommendation is accurate, the user would feel frustrated if the recommendations are too similar to each other and lack diversity [McNee et al. 2006]. Thus a more diversified recommendation set that covers different ranges of users' interests is preferred [Said et al. 2013]. The diversity measure defined above reaches this goal since it focuses on the *whole utility* of the recommendation set, regarding it as a complete entity rather than simple aggregations of individuals under accuracy metrics.

5.2.2. Baselines. We compare our model with both traditional collaborative filtering models and some recent representative techniques for diversified recommendations. For traditional CF models, we choose UCF [Herlocker et al. 1999], ICF [Deshpande and Karypis 2004] and BPR [Rendle et al. 2009], which are representatives in neighbor based models and matrix factorization models in CF. These models are designed to improve accuracy of recommendation results and perform very well in practice. For the diversity metric, we implement TDiv [Ziegler et al. 2005], WDiv [Hurley and Zhang 2011] and AsDiv [Vargas and Castells 2013] model. These three models are designed to encourage diversity of recommendation results. The details of these benchmarks are listed as follows:

— UCF: UCF is a typical neighborhood based method in CF, which tries to predict the target user's preference by suggestions of other liked-minded users. It can be seen as a special case of our proposed model when $\alpha = \beta = 1$.
— ICF: ICF is also a popular neighborhood based approach in CF. Instead of calculating user similarities in UCF, it tries to predict the target user's preference by suggestions of similar items.

— BPR: This is a popular matrix factorization model that is designed for implicit feedbacks in recommender system and is suitable for the binary preference data [Rendle et al. 2009]. This model was directly optimized for ranking and showed good performance in many benchmark datasets and competitions. We implemented it with the MyMediaLite Recommender System Library provided by the authors [Gantner et al. 2011].
— TDiv: This work introduced the intra-list similarity to access the topic diversification level of recommendations and designed a model to balance accuracy and diversity. In this experiment, for fair comparison with other models in the training process, we turn to Eq.(18) to calculate the similarity between items.
— WDiv: The authors formulated the trade-off between diversity and accuracy as a quadratic optimization problem with binary constraints and proposed models to solve this problem efficiently.
— AsDiv:It is an adaption of the search result diversification algorithm xQuAD [Santos et al. 2010]. The authors considered the different aspects of each user's interests directly and developed the diversity component by marginalizing the probabilities over an explicit set of user need aspects.

As to our own two-stage REC framework, the related strategies proposed in this article include the following:

— ECov: This model considers the equal importance of user interests and strong coverage function as shown in Eq.(5) and Eq.(12).
— EICov: This model considers the equal importance of user interests but with incremental coverage functions as shown in Eq.(7) and Eq.(14).
— WICov: This model combines both the weighted importance of user interests and incremental coverage functions as shown in Eq.(8) and Eq. (14).

Besides, we have briefly introduced how to incorporate the coverage measure into ICF and matrix factorization models. We would also like to include the following coverage based models proposed in the generalization part:

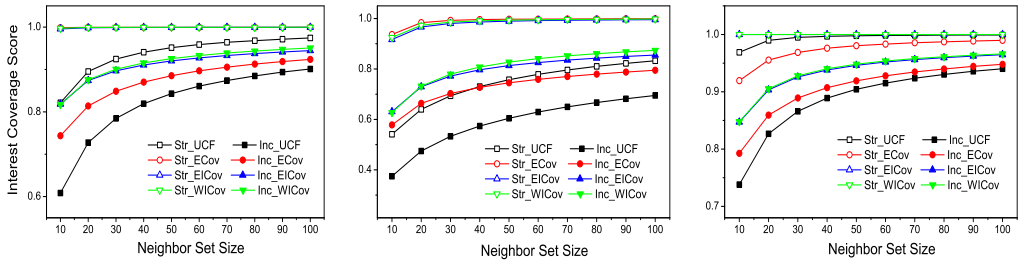
— ICFCov: It is an extension of adapting the coverage measure to ICF as introduced in Section 4.2.1. The comparison between this method and ICF shows whether it is effective to incorporate the coverage measure into ICF.
— MFCov: It is an extension of adapting the REC framework to matrix factorization models as introduced in Section 4.2.2. The latent representations of users are learned from the baseline BPR directly.

Since we mainly focus on introducing the coverage measure into UCF, in the following we would first introduce the influence of various parameters in the REC framework and then compare the various models mentioned above. There are several parameters in these comparison models, we will only report the optimal performance with tuned parameters for fair comparision. For each user, we split the latest 20% of her records for test and the remaining earlier records for training. As each user has at least 5 records, this splitting ensures each user appears at least once in the test data.

5.3. Effectiveness of the Coverage Measure

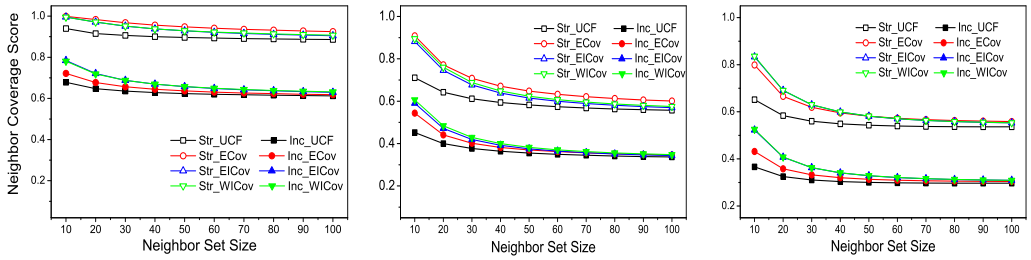
As discussed in Section 3, the main contribution of our work is to naturally diversify recommendation results by the coverage measure. Before detailed comparison with other models, we now investigate whether the proposed REC related models are effective in selecting neighbors that cover more of users' interests and generating recommendations that cover more neighbors.

Fig. 3 and Fig. 4 show the interest coverage scores and neighbor coverage scores with different neighbor set size K on the three datasets. The neighbor size ranges from 10 to 100 with an increment of 10. We set the controlling parameters as $\alpha = \beta = 0.5$ to balance relevance and coverage. In each stage, we calculate both the strong coverage score and the incremental coverage score, where the strong coverage score is shown with open interior marker and the corresponding incremental coverage score of this method is shown with the same marker but with solid interior. For simplicity, the strong coverage score is abbreviated as "Str" and the incremental coverage score as "Inc" in these two figures.



(a) MovieLens (b) Douban (c) Ihou

Fig. 3. The interest coverage score comparison with respect to different neighbor set size K.



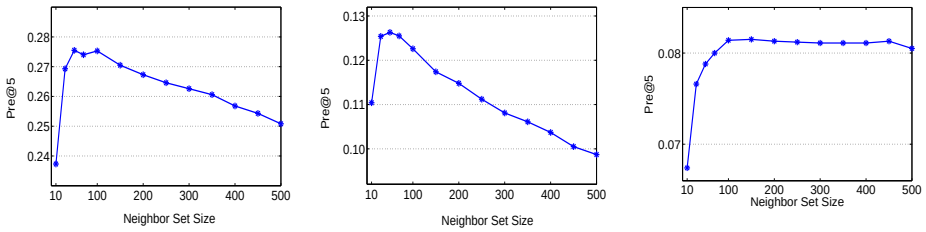
(a) MovieLens (b) Douban (c) Ihou

Fig. 4. The neighbor coverage score comparison with respect to different neighbor set size K.

5.3.1. The interest coverage measure. As depicted in Fig. 3, first we observe that without coverage based measures, both the strong interest coverage value (Eq. (5)) and incremental interest coverage value (Eq. (7)) in UCF are far from satisfactory. E.g., with $K = 10$, the strong coverage score is 0.54 and the incremental coverage score is 0.37 for Douban, thus nearly half of users' interests are not covered. With the increase of the neighbor set size, the coverage scores of UCF increase but there are still gaps compared to the proposed models that consider the coverage measure. Second, when looking at the strong interest coverage scores (open interior in this figure), they all saturate quickly as we increase the size of neighbor set. E.g., in MovieLens and Ihou, the strong coverage score reaches 1 for all models that consider coverage measure even when $K = 10$. As to the sparser Douban data, this score reaches 1 when $K = 30$. The reason is that the strong coverage score only considers whether an item has been covered or not. If it has been covered, this score would never gain even if it would be covered later. Thus after reaching the value 1, the remaining neighbors in ECov are selected from those users that have the largest similarities with the target user

without considering the coverage measure. Third, with the increase of the neighbor set size, the incremental coverage scores of all models get larger and slowly approach the perfect score 1 as this measure also takes the covered times into consideration. Please note that the values of strong coverage score and incremental coverage score can not be compared directly. Since even with the same number of selected neighbors, the strong coverage measure would get much higher score than the incremental measure. Last but not least, under both interest coverage measures, the WICov performs a litter better than EICov, then ECov ranks the third and UCF always has the lowest scores. Thus we conclude that the incremental coverage measure in EICov is more effective to improve the coverage score than the pure strong coverage measure in ECov. And considering the importance of items further improves performance. The improvement on the sparsest Douban data is the most prominent. On average, WICov, EICov and ECov improve the incremental interest coverage score of UCF over 14%, 23% and 26% respectively.

5.3.2. The neighbor coverage measure. Similarly, we plot the performance of the strong neighbor coverage score (Eq. 12) and incremental neighbor coverage value (Eq. 13) in Fig. 4. We set the recommendation size $N = 5$ as commonly done in top-N recommendation. When we do not consider the coverage measure, the neighbor coverage scores are quite low as shown in the results of UCF. The strong neighbor coverage scores are only 0.71 for Douban and 0.65 for Ihou, indicating about one third of the neighbors do not contribute for final recommendations. However, our proposed models can solve this problem to some extent, covering about 90% neighbors for Douban data and 83% for Ihou data. In fact, if we decrease the controlling parameters to smaller values ($\alpha = \beta = 0.5$ in this experiment), the coverage score would increase as we rely more on coverage measures. We would discuss the details of the impact of controlling parameters in the next subsection. With the increase of neighbor size, the neighbor coverage scores of all models decrease. As when more neighbors are involved, it is harder to cover all of these neighbors. But the models that consider the coverage measure still get higher coverage scores than UCF under all situations, indicating the superiority of incorporating neighbor coverage measure in recommendation process.

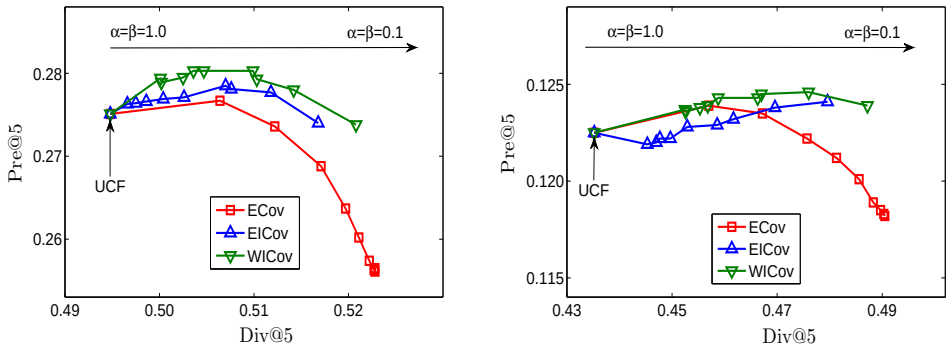


(a) MovieLens (b) Douban (c) Ihou

Fig. 5. The impact of neighbor set size for accuracy of UCF on different datasets.

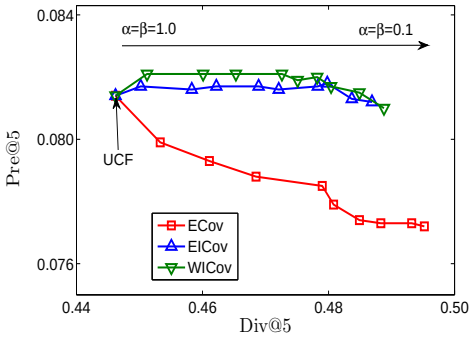
5.4. Sensitivity to controlling parameters

Before the detailed experiments on the final recommendation quality with different controlling parameters, we need to select the neighbor set size K. Fig. 5 shows the accuracy results of UCF with different neighbor size on these three datasets. These three sub figures share the same trends for accuracy. As we increase the neighbor set size at the beginning, the accuracy increases quickly since more neighbors are available to provide recommendations. After the K value exceeds 150, the results drop in all these datasets as we introduce noise when K is rather large. The best K values



(a) Pre and Div on MovieLens

(b) Pre and Div on Douban



(c) Pre and Div on Ihou

Fig. 6. Precision and diversity with regard to control parameters. The x-axis shows control parameters (α and β). For each dataset, the top part shows accuracy and bottom part diversity.

for accuracy range in $[50, 150]$ for these datasets. Thus in the following experiments the neighbor set size is set to be 100.

As discussed, REC diversifies recommendations through IntCov and NeiCov, each with a controlling parameter, i.e., α and β respectively. In both steps, the larger the controlling parameters, the more we rely on individual values, which is thought to increase accuracy while deteriorate diversity [Ziegler et al. 2005; McNee et al. 2006; Said et al. 2012]. To clearly show the balance between accuracy and diversity, Fig. 6 presents the accuracy and diversity results of these three datasets on top-5 recommendation. The x-axis shows the diversity results and the y-axis depicts the accuracy. The controlling parameters are set from $\alpha = \beta = 1$ to $\alpha = \beta = 0.1$ with a step of 0.1 decrease each time. The UCF results are shown on the left on each figure that do not change with those parameters. The righter the results, the better the diversity. The upper the results, the better the accuracy. There are several findings. First, when looking at the ECov model, it ranges to the rightmost of this figure with small values of parameters, indicating it reaches the highest diversity compared to other REC related models. However, the accuracy is deteriorated at this time, since there is a downward trend of the ECov curves of all datasets. On these three datasets, the accuracy results of ECov are worse than UCF when $\alpha = \beta = 0.7$. When turns to EICov and WICov, EICov curves are usually in the upper right position of UCF and WICov in the upper position of EICov, indicating that EICov outperforms UCF on diversity while at least has comparable accuracy to UCF. And considering the weight of interests in WICov

further improves both accuracy and diversity compared to EICov. To sum up, ECov has the largest improvement on Div, then the WICov ranks the second, followed by EICov. UCF always ranks the last on this metric. E.g., the improvement over UCF on the diversity metric reaches 12.6%, 6.1% and 7.3% for ECov, EICov and WICov respectively on Douban data with $\alpha = \beta = 0.2$.

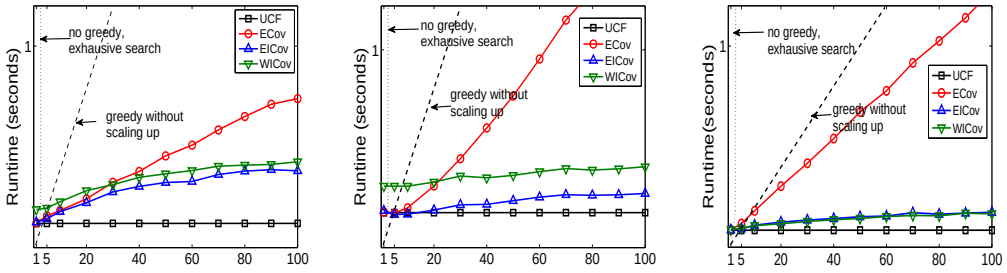
Combining both the results of accuracy and diversity, we summarize several guidelines for selecting models and the controlling parameters. If we put more emphasis on accuracy, then the WICov model seems to be the best choice as it provides better accuracy and diversity than pure UCF. On these three datasets, the WICov always has the largest accuracy values when the controlling parameters (i.e., α and β) is set to be the same value. The controlling parameters can be set as $\alpha = \beta = 0.2$ since it generates the best results of these two measures with regard to all datasets. On the densest MovieLens dataset, the improvements of accuracy and diversity are about 0.7% and 3.4% respectively. On the sparest Douban dataset, the improvements are prominent, with 0.2% for accuracy and 7.3% for diversity. However, if the system values more on diversity, then ECov model seems to be a better choice as it reaches the highest diversity with only a small loss on accuracy. As shown in the figure, when $\alpha = \beta = 0.1$, ECov gets the best individual diversity score of 0.491 on Douban, which has 12.7% improvement on diversity with only 3.3% loss on accuracy compared to UCF.

5.5. Time Efficiency of REC Framework

In the proposed framework, we discussed how to speed up the REC related framework and we will compare the actual runtime of these models. The experiments are conducted on a Core i7 2.94 GHz machine with Windows 8 and 8 GB memory. Fig. 7 shows the time efficiency of the proposed models for each user compared to UCF and some naive algorithms, including the exhaustive search and the naive greedy algorithm. First we observe that, the proposed lazy forward algorithm greatly improves the time efficiency compared to naive algorithms. In fact, REC needs less than 2 seconds to select neighbors and generate top-5 recommendations on all these datasets even when $K = 100$. Second, the actual runtime of all the proposed models divided by that of UCF are far lower than the neighbor size K due to lazy evaluations. In practice, WICov and EICov cost about 2 times as much as UCF even when $K = 100$. However, the ECov model costs about 2 time as much as UCF when $K = 10$ and about 5 times when $K = 100$. This is because when more neighbors are involved, the lazy forward algorithm in ECov needs more iterations to find a suitable neighbor to cover the user's uncovered interests. While in EICov and WICov, the incremental coverage measure would need fewer steps for selecting neighbors when the neighbor set is relatively large, thus has less runtime than ECov. When applied to real applications, the neighborset size is usually chosen in a relatively small value to avoid noise, thus our models have comparable time complexity to UCF.

5.6. Overall Comparison with Other Models

In this section, we compare the overall recommendation results of our models to other baselines with different recommendation set size. Our models include various REC based models and the generalization of adapting the coverage measures and the recommendation framework into other traditional CF models. The accuracy results are shown in Fig. 8 and diversity in Fig. 9. For better illustration, all our proposed models are marked with dashed lines, the traditional CF models are plotted with black solid lines and the remaining baselines of diversity-enhancing models are shown by colored solid lines. The largest recommendation set size is set to be 20 as users are unwilling to receive too many recommendations in the recommender system. For both metrics, the larger values the better performance.



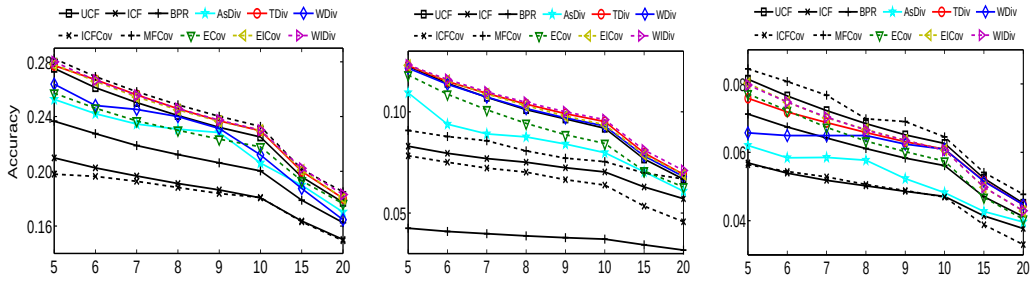
(a) MovieLens (b) Douban (c) Douban

Fig. 7. Runtime comparison with different K for each user.

We first compare the results of our models with regard to other traditional CF models. Among these three traditional CF models, UCF always outperforms over these two metrics. In fact, other researchers also observed that UCF performed better with regard to accuracy when compared to other traditional CF models for top-N recommendation [Gantner et al.]. BPR performs better than ICF on MovieLens and Ihou dataset. However, it has worse performance than ICF on Douban dataset. A possible reason is that the Douban dataset is very sparse (99.845% sparsity) and the number of items are larger than that of users in this dataset. In the learning process, BPR needs to create all pairs between liked items and each unknown item for each user. This limited available data lowers the final accuracy results. Compared to traditional CF models, all of our REC based models show comparable accuracy and much better diversity than UCF under all recommendation set size. E.g., our model WIDiv improves about 12% diversity than UCF with slightly better accuracy on Douban data. The improvement is even larger when compared to ICF and BPR. Similarly, ICFCov, which pushes the coverage measure into ICF, shows improvement on the diversity metric with a little loss of accuracy compared to the ICF model on all these three datasets. As to the MFCov, the experimental results on both accuracy and diversity are much better than BPR. And the accuracy results are even better than UCF on MovieLens and Ihou dataset, indicating it is more effective to calculate the relevance between users with the dimension reduction technique. In summary, the above results show that it is effective to incorporate the coverage measure into any traditional CF model. The diversity results are largely improved with comparable performance of accuracy.

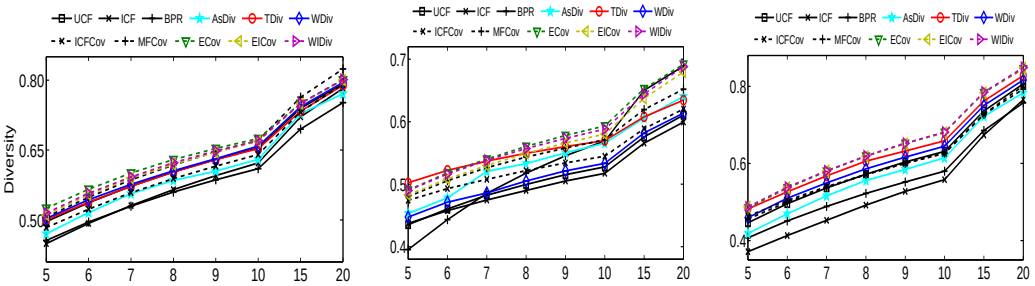
Then we compare our models with other diversity-enhancing models, i.e., TDiv, WDiv and AsDiv. Note that all these models need to generate a candidate recommendation list by traditional CF models in the first step. For fair comparison, we choose the top 50 recommendation results of UCF as the candidate set since UCF outperforms other traditional CF models. We first observe that all these diversity-enhancing models increase the diversity results. Among all these models, our models always perform the best, followed by TDiv on all these three datasets. On average, our models improve about 2% to 10% diversity over these diversity-enhancing models. Our models also have better accuracy results than these baselines since they lose a small amount of accuracy for promoting diversity directly from recommendation candidates. Note that, besides the widely used precision measure for accuracy comparison, we have also measured the recall values of different models and we found the overall trend for the performance of various models is the same as the trend of precision measure on these three datasets. Therefore, for the simplicity of result discussion, we do not report the detailed results of recall values.

For a better understanding of the correlations of recommendation results, Fig. 10 provides the Jaccard similarity of different models with top-5 recommendations. The



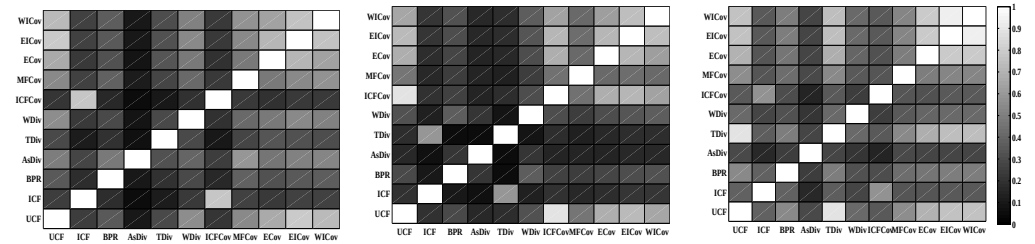
(a) MovieLens (b) Douban (c) Ihou

Fig. 8. The overall accuracy comparison of all models with different recommendation size.



(a) MovieLens (b) Douban (c) Ihou

Fig. 9. The overall diversity comparison of all models with different recommendation size.



(a) MovieLens (b) Douban (c) Ihou

Fig. 10. Jaccard similarity of different models.

darker the color between two models, the smaller similarity of recommendation results between them. It is clearly that ICF has a recommendation list that is very different from others, followed by BPR. We empirically conclude that traditional CF models would generate rather dissimilar results based on their different assumptions. Among all the diversity-enhancing models, AsDiv produced the most dissimilar results as it needed an additional step to model user's aspect preference from predicted preferences learned by tradition CF models. Then it generated recommendations based on user's aspect preference. The remaining diversity-enhancing models only needed to balance diversity and the predicted preferences learned from traditional CF models directly. In contrast, our proposed REC framework implicitly achieves diversity by incorporating the coverage function at each step naturally. Thus our models have about 70% overlap with UCF on these three datasets.

Table IV. Two example users and the Top-5 Recommendations

A typical user in Douban DataSet		
14 training records. Popularity: 10, 7 are written by Junji Ito; Technology:2, mainly about web design; Culture:2, including Emotional Design,which is also related to web design **Interested in comics written by Junji Ito and web design related books**		
UCF Recommendations	TDiv Recommendations	WICov Recommendations
Precision:40%,Diversity: 33.33%	Precision:20%, Diversity: 33.33%	Precision:40%, Diversity: 66.67%
The Hanging Ballons (Pop) Forbidden Fruits (Pop) Gyo 2 (Pop) Uzumaki (Pop) Gyo (Pop)	The Elements of User Experience (Economics) Uzumaki (Pop) Forbbidden Fruits (Pop) Bad Kids (Pop) The Hanging Ballons (Pop)	The hanging Ballons (Pop) The Elements of User Experience (Economics) The Long Tail (Economics) Uzumaki 2 (Pop) The Design of Everyday Things (Culture)
the recommended 5 books are written by Junji Ito	**four books belong to popularity, all are written by Junji Ito**	**two comics written by Junji Ito, the remaining books are related to web design and internet economics**

A typical user in MovieLens		
23 training records, Action: 10, Drama:9, Thriller:9, Sci-Fi:5, Comedy:3, Mystery:1, Adventure:1, War:1, Horror:1, Western:1, Crime:1 **The user has a wide range of interests**		
UCF Recommendations	TDiv Recommendations	WICov Recommendations
Precision:0%,Diversity: 18.19%	Precision:0%, Diversity: 27.27%	Precision:20%, Diversity: 63.64%
Grandfather (Drama) Shakespeare in Love (Comedy|Romance) Fight Club (Drama) Toy Story 2 (Comedy|Animation|Children) Bug's Life (Comedy|Animation|Children)	Toy Story 2 (Comedy|Animation|Children) Grandfather (Drama)|Comedy|Sci-Fi) Doors (Drama|Musical) Out of Africa (Drama|Romance) Elizabeth (Drama)	Shakespeare in Love (Comedy|Romance) Star Wars (Action|Adventure|Romance|Sci-Fi|War) Braveheartn (Action|Drama|War) Being John Malkovich (Comedy) Frequency (Drama|Thriller)
the recommendation set only covers the user's interests for comedy and drama	**these recommendations also only cover the user's interests for comedy and drama**	**cover most of the user's interests, including action, drama, thriller, Sci-Fi and so on.**

5.7. Case Study

We present the top-5 recommendation results of two typical users in Table IV. For better illustration, we only list the best baselines in the previous experiment (TDiv) and UCF. We list the user's category preferences in the training data without details of each record. We can clearly see that, our REC related model WIDiv captures the target user's interests better and recommends a more diversified recommendation set. Take the typical user in Douban for example. From the user's reading history, this user is a fan of the Japanese horror manga artist *Junji Ito* as more than half of his liked books are written by this author. Meanwhile, we guess this user's work is closely related to web design as he has read many books related to this topic. Our model WIDiv is able to recommend diversified results by covering the user's interests for both comic and web design. However, in UCF, only the user's major interests for *Junji Ito* were captured, and thus UCF can only recommend highly homogeneous recommendations of comics drawn by *Junji Ito*. However, in TDiv four books in the recommendation list are comics written by *Junji Ito* and only one book pertains to web design.

6. RELATED WORK

We summarize the related work in this section. In general, the related work can be grouped into the following four categories, namely *traditional relevance based CF models*, *results diversification in information filtering*, *coverage measure and its applications*, and the last category belongs to *submodular optimization and applications*.

Traditional relevance based CF models: CF suggests personalized recommendations to users based on the wisdom of crowds. This area has enjoyed much attention from both academia and industry during the last decade [Adomavicius and Tuzhilin 2005; Koren and Bell 2011; Su and Khoshgoftaar 2009; Zhu et al. 2014]. Given the historical preference data of users, CF models usually recommend a top-N list of items that are most *relevant* to the target user's previous interests. Thus the accuracy metrics, such as MAE and RMSE in rating prediction tasks, precision and recall in ranking related tasks dominated traditional CF evaluations [Herlocker et al. 2004; Herlocker et al. 1999; Bell and Koren 2007b; Sarwar et al. 2000b]. Generally speaking, traditional models in this area can be classified into two categories: neighborhood based models and matrix factorization models [Koren 2008; Gu et al. 2010]. Neighborhood based models tried to infer the relevance between the active user and each candidate item based on similar users' decisions [Breese et al. 1998; Sarwar et al. 2001; Herlocker et al. 1999;

Bell and Koren 2007a]. In contrast, matrix factorization models projected both users and items into the same low latent space and the relevance between them were directly comparable in this latent space [Koren et al. 2009; Mnih and Salakhutdinov 2007; Zheng et al. 2012]. Researchers further combined these two models to improve the accuracy of recommendation results [Koren 2008]. In a word, *traditional CF models were mainly driven by the accuracy goals and many efforts were devoted to designing more sophisticated means to calculate the relevance between users and items with the available sparse preference data*.

Result diversification in information filtering: Recently some researchers argued that accuracy was far from enough for measuring recommendation quality and we should turn to user-centric means to measure the overall quality of recommendation results [McNee et al. 2006; Said et al. 2013]. Among them, diversification has been accepted as an indispensable part for sensing users' satisfaction in both recommendation and search result presentation [Ziegler et al. 2005; Zhang and Hurley 2008; Hurley and Zhang 2011; Boim et al. 2011; Drosou and Pitoura 2012; Campochiaro et al. 2009]. In fact, both of these two applications serve as filtering tasks to mitigate information overload. Users are more willing to see diversified and informative results than highly homogeneous results. Most models for result diversification were based on multi-objective optimization, where the final results were balanced between accuracy and diversity. In practice, diversity metrics were defined in various means. Users can refer to [Vargas and Castells 2011] for a formal framework that summarized several state-of-the-art metrics. Some were directly based on the similarity measure, where the similarity between items was based on the semantic information of items or borrowed from the user-item preference history. Then the diversity of an recommendation set was defined as the average intra list dissimilarity of all pairs of items in that set ([Bache et al. 2013; Ziegler et al. 2005; Hurley and Zhang 2011; Boim et al. 2011]). While others argued novel items were a means to enhance diversity ([Oh et al. 2011; Yin et al. 2012; Cremonesi et al. 2011]). Among them, Said et al. showed that UCF usually generated recommendations that lacks diversity [Said et al. 2012]. Besides, there are several works that are closely related to ours [Ziegler et al. 2005; Hurley and Zhang 2011; Zhang and Hurley 2009; Vargas and Castells 2013]. For instance, [Ziegler et al. 2005] introduced item category to access intra-list dissimilarity of items and presented a topic diversification model to improve recommendation diversity. [Hurley and Zhang 2011] formulated a binary optimization problem with a control parameter that explicitly tuned the tradeoff between accuracy and diversity. These two models can be seen as re-ranking a larger candidate recommendation set from traditional CF models. [Zhang and Hurley 2009; Vargas and Castells 2013; Boim et al. 2011] argued that users' preferences have different sides, thus considering the subsets of the target user's interests was a natural idea for diversified recommendations. Usually the subsets can be obtained by pre-defined semantic information of items or some distance based clustering algorithms. Then the final diversified recommendations can be achieved by combining the partial recommendations based on users' sub profiles. The authors in [Vargas and Castells 2013] also claimed that extracting users' sub profiles in recommender systems is close to query aspects representation for reducing query redundancy in information retrieval [Santos et al. 2010]. Our work has explicit distinctions from these works. The diversified recommendations are reached naturally through the coverage measure and we do not need any category information of items. Thus our model can be easily applied to various recommendation scenarios even when no item information is available.

Coverage and its applications: The coverage measure has been extensively studied in document summarization related tasks. These tasks treated the summarization

problem as finding a low-cost subset that *covers* as much information in the document as possible [Lin and Bilmes 2011; Sipos et al. 2012]. Some researchers also extended this idea for recommending news related tasks with the text information of these items is available [El-Arini et al. 2009; Yue and Guestrin 2011; Pennacchiotti et al. 2012]. However, in the real world, most typical CF systems do not carry such information, e.g., it is hard to extract the content of music [Adomavicius and Tuzhilin 2005]. This limits the generality of previous works. Recently, Hammer et al. argued that instead of recommending the best sellers, one should maximize the probability that a customer makes a purchase [Hammar et al. 2013]. They proposed to use the maximum coverage to select K products that cover the most consumers. However, the recommendations were non-personalized and identical for all users. To the best of our knowledge, we are the first to extend the coverage measure into CF tasks where only user-item preference data is available.

Submodular optimization and its applications: Our work is also related to submodular optimization. Submodularity represents an intuitive diminishing returns property, stating that adding an element to a smaller set helps more than adding it to a larger set. This property has been widely used in applications such as social influence maximization [Kempe et al. 2003], outbreak detection [Leskovec et al. 2007] and document summarization [Lin and Bilmes 2011]. Users can refer to [Krause and Guestrin 2011] for its various applications in optimized information gathering. The fundamental result of greedy algorithms for maximizing submodular functions goes back to [Nemhauser et al. 1978]. Then [Leskovec et al. 2007] proposed to further exploit the submodularity to do lazy evaluations for further optimization. Though it has been widely studied, few have attempted to exploit submodularity for recommendation tasks.

Nevertheless, our proposed model is motivated by various previous works. And we believe this new perspective to look at recommendation opens a door for further research in this area.

7. CONCLUSIONS AND FUTURE WORK

In this paper we proposed a unified framework to generate diversified recommendations for top-N recommendation. Specifically, we first showed the insufficiency of traditional CF models. We argued that most models in this area were driven by the accuracy metrics that lacked diversity for recommendation results. We then proposed a coverage measure that considers the usefulness of the whole set. We detailed how to incorporate the coverage measure into the two steps of UCF for diversified recommendations. Furthermore, we generalized the coverage measure and the proposed recommendation framework from both a data perspective and an algorithm perspective. Experimental results on three real-world datasets showed the superiority of our proposed models.

In the future, we would like to continue this research from two directions. First, in this paper, the generalization of MF models is still mainly based on the REC framework, we would like to explore how to extend the coverage measure to the low dimensional representations of users and items from MF models directly. Second, as diversity is a rather subjective feeling of people, we leave the problem of how to better evaluate the diversity of recommendations from users' perspective as a future research plan.

REFERENCES

Gediminas Adomavicius and YoungOk Kwon. 2012. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Trans. Knowl. Data Eng.* (2012), 896–911.

Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* (2005), 734–749.

Kevin Bache, David Newman, and Padhraic Smyth. 2013. Text-based measures of document diversity. In *KDD'13*. ACM, 23–31.

William J Baumol and Alan S Blinder. 2011. *Microeconomics: principles and policy*. Cengage Learning.

Robert M Bell and Yehuda Koren. 2007a. Improved neighborhood-based collaborative filtering. In *KDD Cup and Workshop at the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. sn.

Robert M Bell and Yehuda Koren. 2007b. Lessons from the Netflix prize challenge. *ACM SIGKDD Explorations Newsletter* 9, 2 (2007), 75–79.

Rubi Boim, Tova Milo, and Slava Novgorodov. 2011. Diversification and refinement in collaborative filtering recommender. In *CIKM'11*. ACM, 739–744.

John S Breese, David Heckerman, and Carl Kadie. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *UAI'98*. Morgan Kaufmann Publishers Inc., 43–52.

Elica Campochiaro, Riccardo Casatta, Paolo Cremonesi, and Roberto Turrin. 2009. Do metrics make recommender algorithms?. In *Advanced Information Networking and Applications Workshops,WAINA*. IEEE, 648–653.

Gerard Cornuejols, Marshall L. Fisher, and George L. Nemhauser. 1977. Location of bank accounts to optimize float: an analytic study of exact and approximate algorithms. *Management Science* 23, 8 (1977).

Paolo Cremonesi, Franca Garzotto, Sara Negro, Alessandro Vittorio Papadopoulos, and Roberto Turrin. 2011. Looking for good recommendations: a comparative evaluation of recommender systems. In *Human-Computer Interaction–INTERACT 2011*. Springer, 152–168.

Abhinandan Das, Mayur Datar, Ashutosh Garg, and ShyamSundar Rajaram. 2007. Google news personalization: scalable online collaborative filtering. In *WWW'07*. 271–280.

Mukund Deshpande and George Karypis. 2004. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems* 22, 1 (2004), 143–177.

Marina Drosou and Evaggelia Pitoura. 2012. DisC diversity: result diversification based on dissimilarity and coverage. *VLDB'12* 6, 1 (2012), 13–24.

Khalid El-Arini, Gaurav Veda, Dafna Shahaf, and Carlos Guestrin. 2009. Turning down the noise in the blogosphere. In *KDD'09*. ACM, 289–298.

Uriel Feige. 1998. A threshold of ln n for approximating set cover. *J. ACM* 45, 4 (1998), 634–652.

Zeno Gantner, Steffen Rendle, Lucas Drumond, and Christoph Freudenthaler. Experimental results for some example datasets. http://mymedialite.net/examples/item_recommendation_datasets.html. (????).

Zeno Gantner, Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2011. MyMediaLite: A free recommender system library. In *Recsys'11*. ACM, 305–308.

Quanquan Gu, Jie Zhou, and Chris HQ Ding. 2010. Collaborative filtering: weighted nonnegative matrix factorization incorporating user and item graphs.. In *SDM'10*. 199–210.

Mikael Hammar, Robin Karlsson, and Bengt J Nilsson. 2013. Using maximum coverage to optimize recommendation systems in e-commerce. In *Recsys'13*. ACM, 265–272.

Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. 1999. An algorithmic framework for performing collaborative filtering. In *SIGIR'99*. ACM, 230–237.

Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems* 22, 1 (2004), 5–53.

Dorit S Hochba. 1997. Approximation algorithms for NP-hard problems. *ACM SIGACT News* 28, 2 (1997), 40–52.

Neil Hurley and Mi Zhang. 2011. Novelty and diversity in top-N recommendation–analysis and evaluation. *ACM Transactions on Internet Technology (TOIT)* 10, 4 (2011), 14.

David Kempe, Jon M. Kleinberg, and va Tardos. 2003. Maximizing the spread of influence through a social network. In *KDD'03*. 137–146.

Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD'08*. ACM, 426–434.

Yehuda Koren and Robert Bell. 2011. Advances in collaborative filtering. In *Recommender Systems Handbook*. Springer, 145–186.

Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *IEEE Computer* (2009), 30–37.

Andreas Krause and Carlos Guestrin. 2011. Submodularity and its applications in optimized information gathering. *ACM Transactions on Intelligent Systems and Technology* (2011), 32–32.

Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne M. VanBriesen, and Natalie S. Glance. 2007. Cost-effective outbreak detection in networks. In *KDD'07*. 420–429.

Hui Lin and Jeff Bilmes. 2011. A Class of submodular functions for document summarization. In *ACL'11*. 510–520.

Qi Liu, Yong Ge, Zhongmou Li, Enhong Chen, and Hui Xiong. 2011. Personalized travel package recommendation. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. IEEE, 407–416.

Qi Liu, Biao Xiang, Enhong Chen, Yong Ge, Hui Xiong, Tengfei Bao, and Yi Zheng. 2012. Influential seed items recommendation. In *RecSys'12*. 245–248.

Sean M. McNee, John Riedl, and Joseph A. Konstan. 2006. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI Extended Abstracts'06*. 1097–1101.

Andriy Mnih and Ruslan Salakhutdinov. 2007. Probabilistic matrix factorization. In *NIPS'07*. 1257–1264.

G. L. Nemhauser, L. A. Wolsey, and M. L. FISHER. 1978. An analysis of approximations for maximizing submodular set functionsłI. *Mathematical Programming* 14, 1 (1978), 265–294.

Jinoh Oh, Sun Park, Hwanjo Yu, Min Song, and Seung-Taek Park. 2011. Novel recommendation based on personal popularity tendency. In *ICDM'11*. 507–516.

Marco Pennacchiotti, Fabrizio Silvestri, Hossein Vahabi, and Rossano Venturini. 2012. Making your interests follow you on twitter. In *CIKM'12*. ACM, 165–174.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI'09*. AUAI Press, 452–461.

Alan Said, Ben Fields, Brijnesh J Jain, and Sahin Albayrak. 2013. User-centric evaluation of a k-furthest neighbor collaborative filtering recommender algorithm. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*. ACM, 1399–1408.

Alan Said, Benjamin Kille, Brijnesh J Jain, and Sahin Albayrak. 2012. Increasing diversity through furthest neighbor-based recommendation. *Proceedings of the WSDM* 12 (2012).

Rodrygo LT Santos, Craig Macdonald, and Iadh Ounis. 2010. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th international conference on World wide web*. ACM, 881–890.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2000a. Analysis of recommendation algorithms for e-commerce. In *EC'00*. ACM, 158–167.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2000b. *Application of dimensionality reduction in recommender system-a case study*. Technical Report. DTIC Document.

Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW'01*. 285–295.

Ruben Sipos, Adith Swaminathan, Pannaga Shivaswamy, and Thorsten Joachims. 2012. Temporal corpus summarization using submodular word coverage. In *CIKM'12*. ACM, 754–763.

Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence* 2009 (2009), 4.

Saúl Vargas and Pablo Castells. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 109–116.

Saúl Vargas and Pablo Castells. 2013. Exploiting the diversity of user preferences for recommendation. In *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval*. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 129–136.

Hongzhi Yin, Bin Cui, Jing Li, Junjie Yao, and Chen Chen. 2012. Challenging the long tail recommendation. In *VLDB'12*. 896–907.

Yisong Yue and Carlos Guestrin. 2011. Linear submodular bandits and their application to diversified retrieval. In *NIPS'11*. 2483–2491.

Mi Zhang and Neil Hurley. 2008. Avoiding monotony: improving the diversity of recommendation lists. In *RecSys'08*. 123–130.

Mi Zhang and Neil Hurley. 2009. Novel item recommendation by user profile partitioning. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*. IEEE Computer Society, 508–515.

Mi Zhang and Neil Hurly. 2009. Evaluating the diversity of top-n recommendations. In *ICTAI'09*. IEEE, 457–460.

Vincent W Zheng, Yu Zheng, Xing Xie, and Qiang Yang. 2012. Towards mobile intelligence: Learning from GPS history data for collaborative recommendation. *Artificial Intelligence* 184 (2012), 17–37.

Hengshu Zhu, Enhong Chen, Hui Xiong, Kuifei Yu, Huanhuan Cao, and Jilei Tian. 2014. Mining Mobile User Preferences for Personalized Context-Aware Recommendation. *ACM Trans. Intell. Syst. Technol.* 5, 4 (2014). DOI:http://dx.doi.org/10.1145/2532515

Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *WWW'05*. 22–32.