THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INCREASE RECOMMENDATION DIVERSITY BY CLUSTER-BASED TOP-N RECOMMENDER

HAOJUN SUI
SPRING 2017

A thesis
submitted in partial fulfillment
of the requirements
for baccalaureate degrees
in Computer Science, Mathematics, and Statistics
with honors in Computer Science

Reviewed and approved* by the following:

Conrad Tucker
Assistant Professor of Engineering Design
Thesis Supervisor

Jesse Barlow
Professor of Computer Science and Engineering
Honors Adviser

*Signatures are on file in the Schreyer Honors College.

Abstract

Recommendation systems have played an important role in helping users find interesting and relevant items in a large catalog of choices. Unfortunately, the performance of recommender systems was measured solely on how accurate the recommendations are with respect to users [1]. However, users may not be satisfied by such recommendations, since they may already purchased these most popular items or they want to know something different from their existing knowledge. Therefore, the amount of additional information that the users gain from the recommendations is limited. In fact, McNee et al. argued that recommendation systems which highly match with user's past behavior will provide users accurate recommendations but zero amount of information and cause narrow focus to the users [1]. Unfortunately, there exists conflict between the diversity and the accuracy in the recommendations. There is widespread skepticism that diversity should be sacrificed to increase accuracy or accuracy should be sacrificed to increase diversity. To date, this skepticism is arguably justified. B. Smyth et al. and D. McSherry have addressed this issue in their research [2, 3], which has focused on increasing the diversity in recommendations. N. Hurley et al. [4] has concluded that the diversity of the recommendation list can be increased at a cost of reducing recommendation accuracy, without providing any further analysis of the positive impact of increasing diversity. This thesis proposes a cluster-based approach to increase diversity in recommendations based on item novelty. The proposed diversity enhancing weighted selection algorithm is evaluated using real-world e-commerce product dataset and demonstrates substantial improvements in both diversity and accuracy, as compared to the recommendation re-ranking approaches, which have been introduced in prior literature for the purpose of diversity improvement. The diversity performance against system accuracy and diversity in the recommendation list is measured with different control parameter. The importance of the control parameter in obtaining the best diversity performance for the system is discussed. The method for evaluating sensitivity of diversity on matching value in recommendations is proposed.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank many people for their support throughout my college career. First, I would like to thank Dr. Conrad Tucker, who has served as my research advisor for the past several years. His guidance and wisdom has helped me tremendously throughout my research career. Through having him as my research advisor, I have learned a great deal that helped me develop my research ethics and will serve me well in the future. In addition, I would like to acknowledge Dr. Tucker's Design Analysis Technology Advancement (D.A.T.A) Lab, which has given insightful input to help guide my research. In particular, Mr. Sunghoon Lim helped me tremendously while at Penn State, from the very beginning of my research career to completing research projects. Also at Penn State, Dr. Jesse Barlow has served as an excellent thesis committee member and academic advisor throughout my academic career, answering all the questions that I had. Furthermore, I would like to thank my fellow colleagues at Penn State as well as my friends for their help and support throughout my life at Penn State. Lastly, I would like to thank my family, who has supported and loved me through my whole life and college career.

Chapter 1

Introduction

1.1 Background

Recommendation systems have played an important role in helping users find interesting and relevant items in a large catalog of choices. For example, Amazon.com, Inc. recommends potential products in their huge collection of items to users based on their purchase history in order to improve the item sales; Google Scholar recommends scholarly literature across disciplines to users based on their query. Figure 1.1 represents Amazon's recommendation system, which recommends products for a user who recently purchased electric fans and digital cameras. Unfortunately, the performance of recommender systems was measured solely on how accurate the recommendations are with respect to the users [1]. In the standard methodology, a travel recommender is rewarded for recommending places a user has already visited, instead of being rewarded for finding new places for the user to visit [1]. However, users may not be satisfied by such recommendations, since they may already acknowledge these accurate recommendations and they want to know something

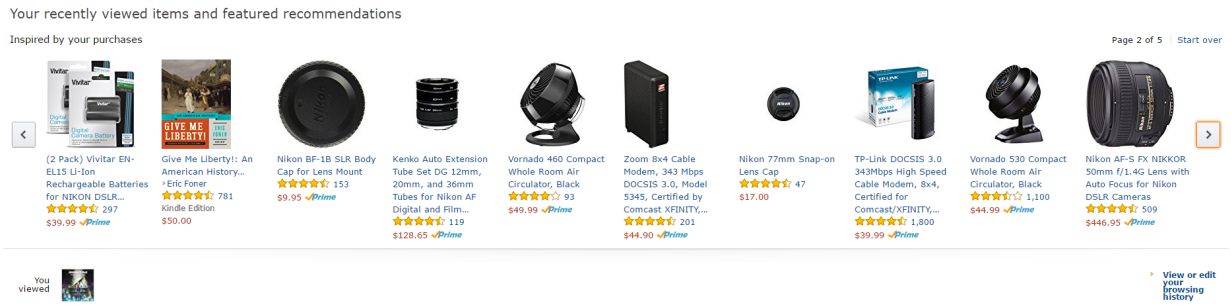

Figure 1.1: Amazon's Recommendations for A User.

different from their existing knowledge. Therefore, the amount of additional information that the users gain from the recommendations is limited. Hence, users may not consider such recommendation to be personalized. For example, an academic paper recommender, which recommends academic papers written by the same author that a user has read in the past; even the user may be strongly interested in these papers, the user will believe this is a poor recommendation, since they can easily find these papers in many other ways [4]. A better academic paper recommender in this setting would recommend less common or popular papers that also match the users interest. A.M. Rashee et al. presented a real world example where once a user rated one Star Trek movie she would only receive recommendations for more Star Trek movies [5]. In the meantime, business that adopts personalized recommendation systems also hopes that there will be varieties of items appeared in the recommendations to stimulate users' new shopping needs. In fact, McNee et al. argued that evaluating recommender system solely on accuracy metrics is not only misguided but also detrimental to the research in recommendation systems, since such act will provide users accurate recommendations with zero amount of information and cause narrow focus to the users [1].

Causing narrow focus to the users has been a major imperfection for Collaborative Filtering top-N recommenders. The goal of these recommenders is to predict a list of N products that a user will like or be interested in purchasing. The rationale behind collaborative filtering recommenders is that if users are similar with respect to their past transaction history, they are highly likely to have similar tastes with respect to their product choices [6, 7]. Thus, products will be recommended

based on similar users' transaction history. One consequence of collaborative filtering algorithm is that items that are well-matched to the user profile/user query are also likely to be highly similar to each other [4]. Hu Rong et al. showed that the recommendation diversity enhances users' perception of recommendation and more importantly their satisfaction [8].

Unfortunately, there exists conflict between the diversity and the accuracy in the recommendations. There is widespread skepticism that diversity should be sacrificed to increase accuracy or accuracy should be sacrificed to increase diversity. To date, this skepticism is arguably justified. B. Smyth et al. and D. McSherry have addressed this issue in their research [2, 3], which has focused on increasing the diversity in recommendations. N. Hurley et al. [4] have concluded that the diversity of the recommendation list can be increased at a cost of reducing recommendation accuracy, without providing any further analysis of the positive impact of increasing diversity.

In this paper, an algorithm named **Diversity Enhancing Weighted Selection** algorithm is described, and the method to integrate this algorithm into existing collaborative filtering algorithm is discussed. The performance of proposed algorithm is measured by the sensitivity of diversity on matching value and by the system precision analysis. The proposed algorithm indeed increases the diversity in the recommendations. The increase in diversity of the recommendations is at the expense of accuracy; however, the relevance between recommendations and user's past behavior is preserved. The algorithm itself generates clusters of items ranked by collaborative filtering algorithm and then selects a list of N items from the clusters. Applying clustering algorithm on items ranked by collaborative filtering algorithm and selecting items from the clusters is being done for the first time. The rationale behind proposed algorithm is that selecting items from different clusters with different characteristics will have items in the recommendation lists with more characteristics, therefore, increase the diversity in the recommendation list. Since increasing diversity indefinitely results in decreasing system accuracy, the metric to measure the sensitivity of diversity performance against system accuracy and diversity in the recommendation list is created. The novel features in algorithm itself:

- A Diversity Enhancing Weighted Selection algorithm that substantially improves in both

diversity and accuracy, as compared to prior literature has been proposed.

- A metric to measure diversity performance has been proposed.

1.2 Thesis Structure

The remainder of the thesis is organized as follows. Related work on increasing diversity in the recommendation list is discussed in Section 2. In Section 3, the existence of novel items is shown in a real-world dataset; the methodology for evaluating the precision and diversity performance of novel item retrieval is discussed; the method to integrate the proposed Diversity Enhancing Weighted Selection algorithm into a top-N collaborative filtering recommendation algorithm is explained. Finally, in Section 5, the proposed strategies are incorporated on benchmark Amazon dataset, which has been used for collaborative filtering research before, then the diversity and precision performance of proposed algorithm is evaluated and compared with conventional collaborative filtering result with different control parameter. Finally, the importance of the control parameter in obtaining the best diversity improvement for the system is discussed.

Chapter 2

Literature Review

2.1 Recommendation Systems

As discussed before, there has been many recommendation systems in the market currently, such as item-based collaborative filtering and case-based recommender systems. Previous researches have worked hard on improving accuracy metrics in the system. Nevertheless, recent research has shown that developing recommendation systems with accuracy as the single goal does not fully utilize the recommendation system. For example, it is argued in McNee and Konstan [9] that evaluation of recommendation system needs to move beyond conventional accuracy metrics. The justification behind this argument is the existence of long tail problem in statistics and business. The long tail problem suggests that the amount of unpopular items dominates the amount of popular items. If we focus on improving accuracy metrics in recommendation system solely, popular items will most likely to be recommended to the users, hence become more popular; in contrast, unpopular items will become more unpopular. C. Anderson stated in his book that

business is making less sales due to the fact that huge amount of unpopular items gets stored away instead of being sold to the consumers [10]. In this thesis, the existence of novel items, the unpopular items in the long tail category, is examined in real world dataset, and a strategy to recommend these items to the consumers is proposed.

2.2 Diversity in Recommendation System

Herlocker et al. discussed the importance of evaluating novelty dimension in the collaborative filtering recommender systems [11]; however, a concrete evaluation metric for evaluating novelty in the recommendation systems was not given in the paper. B. Smyth et al. first proposed algorithms which tackle the issue of increasing diversity in recommendation lists [2]. The authors proposed three heuristic algorithms for selecting recommended list that combine both similarity and diversity. The authors mentioned that the greedy selection algorithm is the best among these algorithms. Greedy selection algorithm each time appends one item into the recommended list based on a proposed heuristic measure combining diversity and similarity. The conflict between similarity and diversity has been discussed in Section 1.1. Hence, introducing more diversity in the recommendation list will impact the retrieval performance. However, the authors did not examine the impact on retrieval performance if diversity in the recommendation list increases. Hence, an evaluation metric to measure the sensitivity of diversity on retrieval performance is proposed in Section 3.6. In Section 4.2, this issue is further examined by introducing control parameter on diversity increase, then the retrieval performance is explicitly evaluated in different system settings. In more recent research, the metric for evaluating diversity in the recommendation list has been proposed by Ziegler et al. [12]. The authors proposed a similarity metric using a taxonomy-based classification and used it to compute an intralist similarity metric to determine the overall diversity of the recommended list [4]. The intralist similarity measurement is comparable to the diversity measurement in B. Smyth et al.'s [2] paper. The difference is that the intralist similarity measurement decreases when diversity is increased in the recommendation list; while B. Smyth et al.'s

diversity measurement increases when diversity is increased in the recommendation list. Ziegler et al. also proposed a heuristic algorithm to increase the diversity in the recommendation list. They methodology is to re-rank the items generated from the collaborative filtering algorithm. Their results matches the expected result, where re-ranked recommendation lists have smaller accuracy measurement than the unaltered ones. Nevertheless, users find the altered lists more satisfying. R. Devooght et al. proposed a collaborative filtering algorithm integrated with recurrent neural networks [13]. Their proposed algorithm increases the diversity in the recommendations by exploring the nearest-neighbors for each user and selecting items based on diversity bias measurement. However, the algorithm only works well for short term recommendations. If a user has past behavior in long term, the resulting recommendations are not promising. Also, the sensitivity of the diversity bias in the algorithm is not discussed. Finally, D. Fleder et al. [14] examined the impact of recommender systems on the diversity of sales. In their paper, a statistical dispersion measurement called the $Gini$ coefficient was proposed to measure sales diversity. The recommendation systems that they examined were top-N recommendation systems instead of rating prediction recommendation system. The goal of both recommendation system is to recommend a list of N products, which end-users may find them relevant or satisfying; however, the methodology for recommending these products is different.

2.3 Top-N Recommendation Algorithms

Top-N recommendation systems recommend items that match users' past behavior; while rating prediction recommendation systems first predict the rating of all items that the end-user may give and then select the top N products that have the highest ratings. The assumption that rating prediction recommendation systems make is that it is highly likely users will be satisfied with the products that have higher ratings. In this thesis, a good recommendation system is believed to be capable of classifying items as being relevant or not relevant to the users, and then recommending as many relevant items as possible. Many research have been carried out on rating prediction

recommendation algorithms. For example, Wang et al. proposed a method to unify the user-based and item-based collaborative filtering algorithms by similarity fusion [15]; Xue et al. combined the advantages of memory-based and model-based approaches by introducing a smoothing-based method, in order to improve the accuracy of the predictions [16]; G. Karypis proposed an item-based collaborative filtering algorithm [7], where an item distance matrix is generated first from the dataset, then, the items are sorted according to their average similarity to all items in the user profile, and the N best items are recommended. In fact, some of these research were motivated by the Netflix prize in recent years, where a team wins if the rating predictions generated from their recommendation system have smallest root mean square error compared to Netflix's real world movie rating dataset. However, the recommendation system that gives the highest accurate prediction does not necessarily perform well in recommending relevant products to the users. One way to think of it is that a recommendation system that recommends all Star Treks series to a user who gave high ratings for Star War series is not a good recommendation system. Even the user may give high ratings to the Star Trek series, the relevant products in the recommendations are limited. Nevertheless, rating prediction recommendation system gives rating predictions for a set of products, and in order to make recommendations, it still need to decide which of these should be recommended to the users. The most common approach, referred as predict-and-select-highest strategy, is to recommend those items with the highest predicted ratings. Table 2.1 shows related recommeder systems and the features they support. The green entries show features that the corresponding system supports. Table 2.1 reveals that, while others have implemented a subset of the features we are providing, to the best of our knowledge, none has achieved them in a combined manner. In this thesis, a strategy that integrates the predict-and-select approach and novel recommendation support is presented. Additionally, the sensitivity of diversity on similarity in recommendations is defined and evaluated for proposed strategy.

Table 2.1: Literature review of supported features, compared to what is being proposed in this work

Authors	top-N Recommendation	Rating Prediction Recommendation	Novelty Measurement	Sensitivity of Diversity on Similarity
G. Patil *et al.* (1982)	■		■	
D. Reynolds *et al.* (1997)	■		■	
G. Karypis *et al.* (2001)	■	■	■	
B. Smyth *et al.* (2001)	■	■	■	
K. Nehring *et al.* (2002)	■		■	
Herlocker *et al.* (2004)	■	■		
Ziegler *et al.* (2005)	■	■	■	
Xue *et al.* (2005)	■		■	
S. McNee *et al.* (2006)			■	
Wang *et al.* (2006)	■		■	
D. Fleder *et al.* (2007)	■		■	
N. Hurley *et al.* (2011)	■	■	■	
R. Devooght *et al.* (2017)	■	■	■	
H. Sui *et al.* (2017)	■	■	■	■

Not Implemented	
Full Feature	■

Chapter 3

Methods

In this chapter, the Diversity Enhancing Weighted Selection algorithm that will increase diversity in the recommendations is proposed. The proposed algorithm will be integrated into existing collaborative filtering top-N recommendation framework. The performance of the integrated system is evaluated based on diversity in the recommendations and precision analysis. The method used for computing a top-N recommendation list for a user u is given in Figure 3.1. In this thesis, G.Karypis's $SUGGEST$ item-based collaborative filtering algorithm is used as a baseline model and compared with the proposed algorithm, since many of previous works have compared their algorithms with G.Karypis's algorithm. The performance evaluation metrics are discussed in the following sections.

3.1 Matching Value of a Set

Many applications of information retrieval try to solve the problem of finding a subset of items that best match a query or request issued by a system user. These subset of items in reality could be

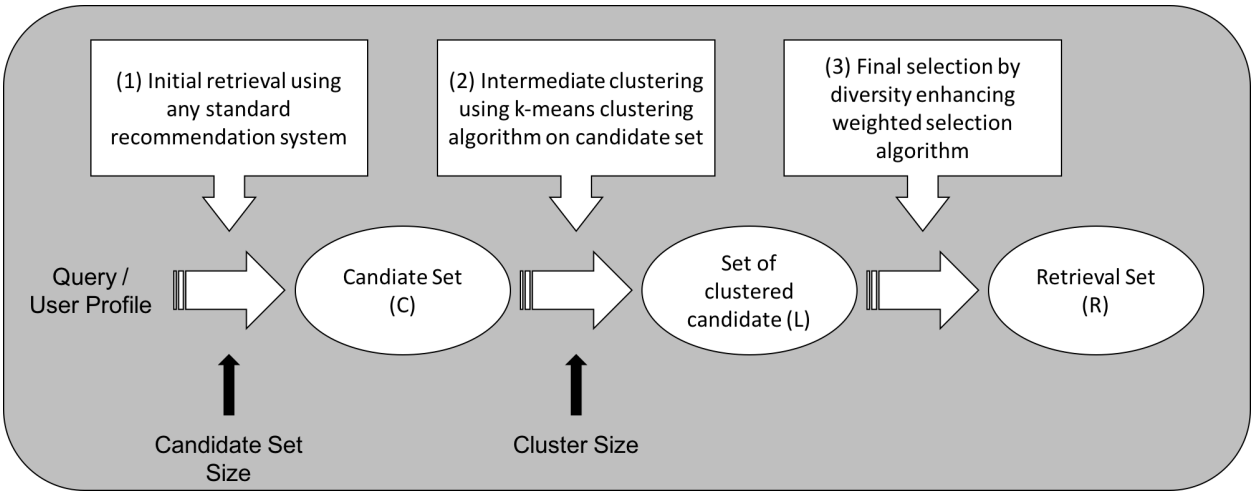

Figure 3.1: Framework for Top-N prediction.

documents returned by a search engine or products recommended for purchase in a recommender system. To translate the recommender system to mathematical language, let $\mathcal{Q}$ denote a set of all possible queries that can be issued by the users to the system and $\mathcal{I}$ denote the set of items in the system. Let $q_u \in \mathcal{Q}$ denote a query issued by user u and $R \subset \mathcal{I}$ denote the subset of items that returned by the recommender system. For future reference, this subset of items is referred to as recommended list. Therefore, the recommender system algorithm can be modeled as the following. Denote a *matching function* to be $f_m : \mathcal{Q} \times 2^{\mathcal{I}} \to \mathbb{R}$, such that $2^{\mathcal{I}}$ represents the power set of $\mathcal{I}$ and $f_m(q_u, R)$ is a real value associated with subset R. Without loss of generality, for a particular query issued by user u, the subset R, which gives higher $f_m(q_u, R)$, corresponds to better matches. Therefore, according to the matching function, the conventional collaborative filtering algorithm solves the problem of returning a subset R with size N which maximizes $f_m(q_u, R)$.

In reality, the method used to compute the matching value for a subset of items varies. In this thesis, the method that was used in Hurley et al.'s paper [4] is adopted to compute the matching value for the retrieval set. That is, the matching function f_m can be represented as the average of matching values of items contained in the retrieval set. In other words, there exists a function $g_m : \mathcal{Q} \times \mathcal{I} \to \mathbb{R}$, such that

$$f_m(q_u, R) = \frac{1}{|R|} \sum_{i \in R} g_m(q_u, i) \tag{3.1}$$

- $f_m(\cdot,\cdot)$: Matching value between two sets
- $g_m(\cdot,\cdot)$: Matching value between a set and an item
- R : Candidate recommended list
- $|\cdot|$: Cardinality of a set
- i : An item in R

3.2 Diversity of a Set

In Section 2.2, different methods to compute the diversity of a set in the literature are discussed. G. Patil et al. first defined the rarity of a single element in the set. Then, the authors proposed to model diversity of a set as the average rarity of the elements in the set [17]. In contrast, K. Nehring et al. and D. Reynolds et al. proposed to model diversity of a set as the average dissimilarity or distance value of all pairs of elements in the set [18, 19]. The distance value of a single pair of elements in the set is a function defined to be $d : \mathcal{I} \times \mathcal{I} \rightarrow \mathbb{R}$, such that $d(i, j) \geq 0$ for any $i, j \in \mathcal{I}$. Hence, the diversity $f_D(R)$ can be represented as the average distance value of all pairs of elements in set R. That is,

$$f_D(R) = \frac{1}{p(p-1)} \sum_{i \in R} \sum_{j \in R, j \neq i} d(i, j), \tag{3.2}$$

- $f_D(R)$: Diversity of set$R$
- p : Cardinality of set R
- i, j : A pair of items in R
- $d(\cdot,\cdot)$: Distance between two items

In K. Nehring el al's paper, the authors defined this distance function to be symmetric, in other words, $d(i, j) = d(j, i)$ for any $i, j \in \mathcal{I}$ [18]. In this thesis, K. Nehring et al. and D. Reynolds et al.'s approach is adopted to model the diversity of a set.

The distance function $d(i, j)$ is application dependent. For example, the distance between two documents can be calculated by the number of vocabularies that they differ; while the distance between two products can be calculated by the categories they belong to. Even the distance function can have many definitions, the only requirement of the distance function is argued by N. Hurley el al [4]. The authors argued that the requirement is that the distance between a pair of elements in the set and query matching values should be calculated, without restriction on the feature space or method used to calculate these values [4].

3.3 Novelty of an Item

In Section 2.1, the existence of novel items in real world example and the impact they have on user satisfaction are discussed. However, the question of how to calculate the novelty of an item in the set still remains. In the field of information retrieval, different methods of calculating the novelty of the retrieval set have been proposed in the literature. R. Baeza-Yates et al. defined the novelty of the recommended list to be the ratio of known relevant items and unknown relevant items with respect to a particular user [20]. Denote $H \subset R_u$ to be the subset of items in R_u, which R_u represents the retrieval set R that the recommender system recommended for a user u. The items in H can be categorized into two categories, the set of items that were already known to the user, denoted as H_K, and the set of items that were unknown to the user, denoted as H_U. In other words, $H = H_K \cup H_U$. Therefore, the novelty of the recommended list can be represented as,

$$n(R) \triangleq \frac{|H_U|}{|H_K| + |H_U|} = \frac{|H_U|}{|H|}. \tag{3.3}$$

- $n(R)$: Novelty of set$R$
- $|\cdot|$: Cardinality of a set
- H_U : Items known to user
- H_K : Items unknown to user
- H : Items in user past behavior

N. Hurley el al. argued that this definition is too restrictive and not practical, since there is limited knowledge on prior user behavior [4]. The authors suggested that instead of measuring the novelty of an item by the proportion of unknown items in the retrieval set, the novelty of an item should be measured by measuring how unusual this item is with respect to users normal tastes. That is, the novelty of an item $i \in R$ is defined as

$$n_R(i) = \frac{1}{p-1} \sum_{j \in R, j \neq i} d(i,j), \tag{3.4}$$

- $n_R(i)$: Novelty of item i in set R
- p : Cardinality of set R
- i, j : A pair of items in R
- $d(\cdot,\cdot)$: Distance between two items

With this definition, the diversity of the recommended list is set to be the average novelty of the items in the set. That is,

$$f_D(R) = \frac{1}{p(p-1)} \sum_{i \in R} \sum_{j \in R, j \neq i} d(i,j) = \frac{1}{p} \sum_{i \in R} n_R(i), \tag{3.5}$$

- $f_D(R)$: Novelty of set R
- $n_R(i)$: Novelty of item i in set R
- p : Cardinality of set R
- i, j : A pair of items in R
- $d(\cdot, \cdot)$: Distance between two items

Therefore, the items in the retrieval set associated with higher novelty value are the ones that have greater distance from other items in the retrieval set. Hence, a conventional similarity-based recommendation algorithm would have lower probability to recommend these novel items to the user. The existence of novel items in real world dataset is examined in the following. In this thesis, N. Hurley et al.'s method to calculate the novelty of items is adopted.

The real world dataset was collected by McAuley et al. [21, 22] from Amazon.com and mainly used for collaborative filtering recommender systems research. Duplicate items were removed in this dataset. The dataset contains user's past purchase history and the relationship between products, such as whether they are bought together or viewed together. Therefore, the novelty of products can be measured in a user's past purchase history, using the item distance function. The item relationships from the dataset are treated as a graph. Denote $G = (V, E)$ to be finite undirected graph for the item relationships. The set V contains the vertices in the graph, in this case, the set $\mathcal{I}$ of all products in Amazon dataset. The set E is the set of all edges in the graph. The edge between two products is identified if they are bought together or viewed together. Hence, the item distance function for product i and j can be defined as the graph distance $d(i, j)$ between vertices i and j in G, in other words, the minimum length of the paths connecting them. If no such path exists (i.e., if the vertices lie in different connected components), then the distance is set equal to ∞. In Figure 3.2, the histogram of the novelty of items in the Amazon dataset across all users' past purchase history is presented. The novelty values are normalized in order to obtain a smooth curve in the histogram plot. From Figure 3.2, it can be observed that there is a large distribution of products that associated with large novelty value. In fact, there are 31.47% of products in user past

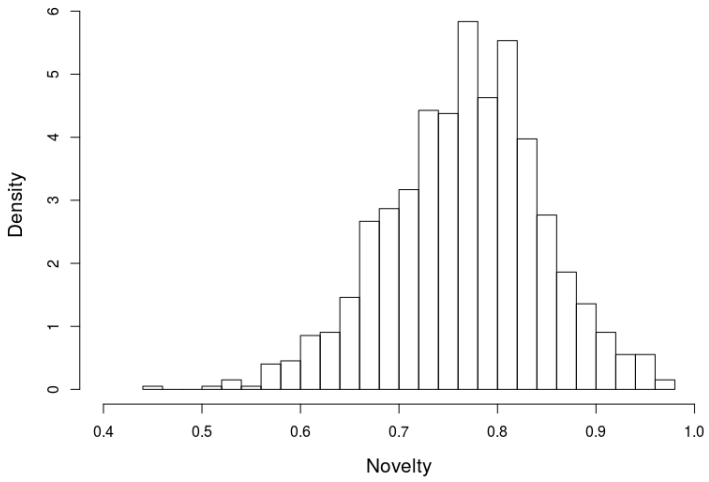

Figure 3.2: Histogram of the item novelty in Amazon dataset across all users' past purchase history.

purchase history that have novelty value larger than 0.8. These are the products that a conventional similarity-based recommendation algorithm finds the hardest to recommend to the users. Hence, a recommender system that does not incorporate the novel items in the recommendation is not considered to be a good recommender system.

3.4 Increasing the Diversity of the Recommendation List

The recommended list constructed by the conventional similarity-based algorithm contains items that are similar to the user query; therefore, these items are likely to be similar to each other. These items are retrieved based on their similarity metric, where the highest p items with similarity metric will be chosen indefinitely. Therefore, the retrieved items will never contain the novel items discussed before, which are less similar to the user query, but still relevant to the user. N. Hurley et al. argued that increasing the diversity of R will increase the number of overlapped items between R and L_u [4]. That is, $|R \cap L_u|$ will become larger as the diversity of R is increased. From this point of view, the rationale behind proposed algorithm is that selecting items from different clusters with different characteristics will have items in the recommendation lists with more characteristics, therefore increase the diversity in the recommendation list. Hence, in

order to force to retrieve items based on similarity and diversity, k-means clustering algorithm is used on the candidate set C to obtain set of clustered candidate L. Therefore, clusters with diverse characteristics will be formed by k-means clustering algorithm. Since items in C are already sorted based on their similarity score, the similarity or matching values are well-preserved inside the cluster. Thus, selecting items in the clusters will enforce diversity and maintain the similarity metrics in the recommended list.

3.5 Diversity Enhancing Weighted Selection Algorithm

The rationale behind Diversity Enhancing Weighted Selection algorithm is that, if the set of clusters has items sorted by corresponding item novelty metric, then selecting item each time with the least novelty metric from each cluster will maintain the accuracy metric but increase the diversity among the recommended list. Let L denote the set of clustered candidate obtained from last section. Let $\mathcal{I}_{l_i}$ denote the set of items included in cluster $l_i \subseteq L$ and $\mathcal{I}_{l_i}^{(0)}$ denote the item associated with lowest novelty metric in $\mathcal{I}_{l_i}$. Let $r_{l_i}^{(0)}$ denote the rank of the item associated with lowest novelty metric in $\mathcal{I}_{l_i}$ in C. Let n_{l_i} denote the number of items that have already extracted from l_i to the recommended list. Therefore, the proposed Diversity Enhancing Weighted Selection algorithm is shown in Algorithm 1. The algorithm itself is an iterative method. As Algorithm 1 shows, the item associated with higher rank in C will be more likely to be included in the recommended list. However, choosing items solely on its rank is insufficient, since the items in cluster which all of its items have higher rank than others in C will have higher probability to be chosen. This does not improve the diversity in the recommended list; in fact, it is equivalent to choosing top-N recommendations from C. Therefore, the penalty for choosing the items in same cluster is introduced. That is, each time the item gets selected from the cluster $\mathcal{I}_{l_i}$, there will be $1/|\mathcal{I}_{l_i}|$ less probability to choose items from $\mathcal{I}_{l_i}$ again. In this case, selecting items from multiple clusters introduces diversity into the recommended list.

Algorithm 1 Diversity Enhancing Weighted Selection Algorithm

Input: Set of clusters with L size p.
Output: Recommended list R with size p.

procedure DEWSA(L)
  $n \leftarrow 0$
  while $n \neq p$ **do** ▷ Recommended list R is not filled
    Choose $\mathcal{I}_{l_i}^{(0)}$ with highest $(|C| - r_{l_i}^{(0)})/r_{l_i}^{(0)} \times (|\mathcal{I}_{l_i}| - n_{l_i})/|\mathcal{I}_{l_i}|$
    $n_{l_i} \leftarrow n_{l_i} + 1$
    Append $\mathcal{I}_{l_i}^{(0)}$ to R
    Delete $\mathcal{I}_{l_i}^{(0)}$ from $\mathcal{I}_{l_i}$
    $n \leftarrow n + 1$
  return R

3.6 Trade-Off between Diversity and Similarity

The method that will potentially increase the diversity in the recommended list is discussed above. However, a recommender system that recommends diverse but poorly matched items is unlikely to produce satisfactory results. Therefore, the goal to recommend a set R that has high diversity stands against the goal to recommend a set R that highly matches with user query. In the following, the system performance measurement for diversity improvement that captures the trade-offs between these goals is presented. Let C_p be the top p items in C and R_p be the p items selected from the set of clustered candidate L. To compare the accuracy metric between C_p and R_p, the accuracy difference Φ is defined to be

$$\Phi(R_p, C_p) = \frac{f_m(R_p) - f_m(C_p)}{f_m(C_p)} \tag{3.6}$$

- $\Phi(\cdot, \cdot)$: Accuracy improvement between two sets
- $f_m(\cdot)$: Matching value of a set
- C_p : Initial recommended list from collaborative filtering algorithm
- R_p : Revised recommended list from proposed algorithm

Adopting the definition of accuracy difference, the diversity performance Θ is defined to be

$$\Theta(R_p, C_p) = -\frac{f_D(R_p) - f_D(C_p)}{\Phi f_D(C_p)} \tag{3.7}$$

- $\Theta(\cdot, \cdot)$: Diversity improvement between two sets
- $f_D(\cdot)$: Diversity of a set
- C_p : Initial recommended list from collaborative filtering algorithm
- R_p : Revised recommended list from proposed algorithm

Since there is an negative relationship between diversity and similarity, in order to make the diversity improving function to be an increasing function, the result from the diversity performance measurement is negated. From the above definition, if the diversity difference is larger than the accuracy difference, a diversity performance measurement higher than 1 should be obtained. In the case when the diversity difference is less than the accuracy difference, this is the cut-off point where the system is generating poor recommendations to the users.

3.7 A Toy Example

To provide a better picture of the trade-off between diversity and similarity metric, a toy example is presented in the following. Assume that the example dataset contains six items $\mathcal{I} = \{i_1, i_2, i_3, i_4, i_5, i_6\}$. The distance between each pair of the items is depicted in Table 3.1. P_u denotes user past behavior that was recorded and L_u denotes the user past behavior that was both recorded and unrecorded. In both cases, 1 indicates that the user has bought this item in the past, 0 indicates that the user has not bought this item in the past, - indicates that the particular record is missing.

Assume that the proposed recommender system wants to recommends $N = 2$ items to the user.

	i_1	i_2	i_3	i_4	i_5	i_6
i_1	0.0	0.5	0.75	0.8	0.7	0.7
i_2	0.5	0.0	0.6	0.75	0.6	0.6
i_3	0.75	0.6	0.0	0.8	0.5	0.5
i_4	0.8	0.75	0.8	0.0	0.9	0.7
i_5	0.7	0.6	0.5	0.9	0.0	0.9
i_6	0.7	0.6	0.5	0.7	0.9	0.0
P_u	1	-	-	-	-	1
L_u	1	1	1	0	1	1
$n_L(i)$	0.67	0.58	0.59	0.79	0.68	0.68

Table 3.1: Distance between items and their Novelty Values.

	R		$f_m(R)$	$f_D(R)$	Θ
R_1	i_2	i_3	0.41	0.6	-
R_2	i_2	i_4	0.35	0.75	1.71
R_3	i_2	i_5	0.32	0.6	0.0
R_4	i_3	i_4	0.31	0.8	1.37
R_5	i_3	i_5	0.28	0.5	-0.53
R_6	i_4	i_5	0.22	0.9	1.08

Table 3.2: Recommendation Lists, Diversity and Similarity, and Diversity Improvement.

The six possible recommendation sets $R_1, \cdots, R_6$ are shown in Table 3.2. Since the recommendation set only contains one pair of items, the set diversity $f_D(R)$ can be obtained from the item distance matrix directly, for example, $f_D(R_2) = d(i_2, i_4) = 0.75$. As proposed in Section 3, the matching value $f_m(R)$ is defined to be the average matching value of items contained in R to the user query. The usual definition for matching value between a pair of items is that the summation of matching value and distance value should sum up to one. That is $g_m(i,j) + d(i,j) = 1$. Therefore, the matching value for R_2 is

$$\begin{aligned} f_m(P_u, R_2) &= \frac{1}{4}(g_m(i_1, i_2) + g_m(i_6, i_2) + g_m(i_1, i_4) + g_m(i_6, i_4)) \\ &= \frac{1}{4}(0.5 + 0.4 + 0.2 + 0.3) = 0.35 \end{aligned}$$

The values of $f_m(R)$ and $f_D(R)$ for each candidate set are shown in Table 3.2.

Based on the result from Table 3.2, the conventional collaborative filtering algorithm is going

to select R_1 to be the recommended list, as this set has the highest matching value. From this on, the diversity performance for other candidate sets will be computed and compared with this baseline set. The diversity performance is defined to be the ratio between the diversity difference and accuracy difference, for example, the diversity performance for R_2 is

$$\Theta(R_2) = -\frac{(f_D(R_2) - f_D(R_1))/f_D(R_1)}{(f_m(R_2) - f_m(R_1))/f_m(R_1)} = -\frac{(0.35 - 0.41)/0.41}{(0.75 - 0.6)/0.6} = 1.71$$

One approach for increasing diversity in the recommended list is to select the candidate set that has the highest diversity value, which in case is R_6. However, the diversity performance for this candidate set is not the highest. In terms of selecting the recommended list based on diversity performance value, candidate set R_2 will be selected since it has the highest diversity improvement. It is clear from this example that the recommended list selected by the proposed optimization strategies will not always have the best accuracy value. However, in order to select the strategy that truly matches with user's past behavior, the prior knowledge of L_u is required, which is not available in practice. This is the reason why recommendation systems are built to predict the best subset of items that the users may find satisfied.

3.8 Diverse Recommendation Lists

The initial recommendation is obtained from G. Karypis's Collaborative Filtering algorithm. As the result from the algorithm, the items are already ranked based on their similarity to user profile. In order to produce recommended lists that are diverse and also similar to user profile, the strategies introduced in Chapter 3 are applied to the candidate set of items C obtained from the initial recommendation. Then, the k-means clustering algorithm is performed on the candidate set C to form set of clustered candidate L, which has size $|L| = p$. Then, the proposed Diversity Enhancing Weighted Selection algorithm is applied to select N items from the clusters L. The optimal cluster size p is the one associated with highest diversity performance in the recommendations. Since the time complexity of the Diversity Enhancing Weighted Selection algorithm highly

depends on the size of the cluster it needs to form, it can be desirable to limit the candidate set to a tractable size. In the experiments on collaborative filtering systems, the size of C is upper bounded by $|C| \leq N^2$.

3.9 Model Performance Evaluation

Precision and recall are widely used in the field of information retrieval to evaluate system accuracy. Let T_u denote the the items that are known to be relevant to the user query and R_u denote the recommended list of items for user u from the recommender system. B.Sarwar et al. proposed the precision and recall metrics in the context of recommendation systems [20] to be:

$$\text{Precision} = \frac{|T_u \cap R_u|}{|R_u|} = \frac{|T_u \cap R_u|}{N} \tag{3.8}$$

$$\text{Recall} = \frac{|T_u \cap R_u|}{|T_u|} \tag{3.9}$$

- T_u : Set of items that are known to be relevant to the user query
- R_u : Set of items obtained from the recommendation system
- N : Number of recommendations that the users receive
- $|\cdot|$: Cardinality of a set

Precision metric measures the fraction of all recommended items that are relevant; while recall metric measures the fraction of all relevant items that are recommended. The precision metric decreases when the size of recommendations increases; while the recall metric increases when the size of recommendations decreases. Even though the system performance can be measured by these two metrics, they often cannot determine if one recommendation system is superior to another. For example, if one recommendation system has higher precision metric but lower recall metric than other, it is unfair to conclude that the one with higher precision is superior. There-

fore, F_1 score is introduced to determine which recommendation system provides more accurate recommendations to user. F_1 score is defined to be

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.10}$$

The F_1 is the harmonic mean of the precision and recall metrics, and it can take value between 0 and 1. The recommendation system associated with higher F_1 score is more accurate.

Chapter 4

Evaluation

In the evaluation of collaborative filtering recommender systems, the following three retrieval strategies are examined as baselines. The first retrieval strategy, denoted as RR, is to randomly select a list of N items from C with equal probability. The second retrieval strategy, denoted as SR, is proposed by G. Karypis et al. [7], which selects a list of N items from C that have the highest similarity metric. The third retrieval strategy, denoted as GR, is the greedy optimization algorithm proposed by N. Hurley et al. [4], which selects a list of N items one at a time based on the greedy metric. The Diversity Enhancing Weighted Selection algorithm proposed in this thesis is denoted as $DEWS$.

4.1 Dataset

The dataset used in this thesis is crawled by McAuley et al. [21, 22] from $Amazon.com$, whose characteristics are shown in Table 4.1. The authors collected this data in order to perform a breadth-first search on the user-product-review graph until termination, meaning that the dataset

Category	Users	Items	Reviews	Edges
Men's Clothing	1.25M	371K	8.20M	8.22M
Women's Clothing	1.82M	838K	14.5M	17.5M
Music	1.13M	557K	6.40M	7.98M
Movies	2.11M	208K	6.17M	4.96M
Electronics	4.25M	498K	11.4M	7.87M
Books	8.20M	2.37M	25.9M	50.0M
All	21.0M	9.35M	144M	237M

Table 4.1: Dataset statistics for a selection of categories on Amazon.

is a fairly comprehensive collection of English language product data. According to the authors, duplicate items have been removed [21, 22]. For each category, the users dataset contains all the users that have purchased items in this category, the items dataset contains all the items that were bought or viewed by any user in the users dataset, the reviews dataset contains the textual review and the rating value up to 5 that the user gives to a particular product, and the edges dataset contains the item relationship within or across the categories. As suggested in Section 3, the item relationships are treated as a graph. Denote $G = (V, E)$ to be finite undirected graph for the item relationships. The set V contains the vertices in the graph, in this case, the set $\mathcal{I}$ of all products in Amazon dataset. The set E is the set of all edges in the graph. The edge between two products is identified if they are brought together or viewed together. Hence, the item distance function for product i and j is defined as the graph distanced $d(i, j)$between vertices i and j in G, in other words, the minimum length of the paths connecting them. If no such path exists (i.e., if the vertices lie in different connected components), then the distance is set equal to ∞.

4.2 Evaluation of DEWS Algorithm

In section 3.5, diversity enhancing weighted selection algorithm is explained. The system performance of this algorithm is examined on the Amazon dataset. In this evaluation, the size of the final recommended list R is set to be $|R| = 20$. Therefore, there are $|R|^2 = 400$ products extracted from the candidate set C obtained from G. Karypis's $SUGGEST$ Collaborative Filtering

algorithm. Then, the k-means clustering algorithm is applied on the extracted set of products with different cluster size k. Finally, the Diversity Enhancing Weighted Selection algorithm is applied on the clusters formed by k-means algorithm.

For a given value of p, a recommended list R is generated as follows. The Amazon dataset is divided into a training database Y_T and validation database Y_V in a 80:20 ratio. The item distance matrix D will be obtained from G. Karypis's algorithm on the training dataset Y_T. The item distance matrix D is trained using the cosine similarity metric. Then, a user is selected at random from the validation dataset Y_V. Next, an initial recommended list with size 20 – the top 20 products that have the highest similarity value – is generated from the candidate set C using the G. Karypis's $SUGGEST$ recommendation algorithm. From this point, RR , GR, and $DEWS$ are applied on the candidate set C. RR algorithm randomly selects a list of 20 products from candidate set C with equal probability for each product. GR algorithm selects a list of 20 products from candidate set C one by one based on the greedy metric. $DEWS$ algorithm is applied in the following strategy. First, k-means clustering algorithm is applied to form p clusters on the candidate set C to obtain the set of cluster candidate L. Finally, a recommendation list R' is generated by $DEWS$ algorithm. The diversity and the matching value of R' are calculated, and the corresponding diversity performance is also calculated. Along with the process, the dataset was re-split into Y_T and Y_V in total of 5 times, and each time, the desired metrics are evaluated for 1000 different random users. In order to measure the sensitivity of diversity on accuracy in the recommendations, the process was simulated with different values of p – the number of cluster formed. The average diversity and similarity of all the sets R' with respect to p are presented in Figure 4.1 and Figure 4.2. The average diversity improvement measurement is presented in Figure 4.3.

It's clear from Figure 4.1 and Figure 4.2 that the diversity in the recommendation list is increased for RR, GR, and $DEWS$; the similarity in the recommendation list is decreased for RR, GR, and $DEWS$. From Figure 4.1 and Figure 4.2, the diversity in R generated by $DEWS$ starts off with the same value as the one generated by SR, then increases and converges as the number of clusters formed increases; while the similarity in R generated by $DEWS$ starts off with the

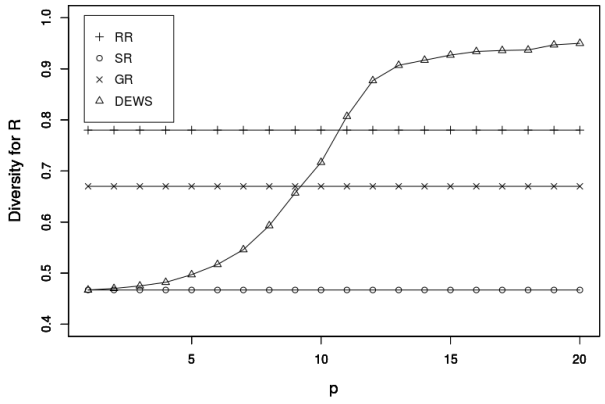

Figure 4.1: Mean diversity of the recommended set with respect to the number of clusters at $N = 20$.

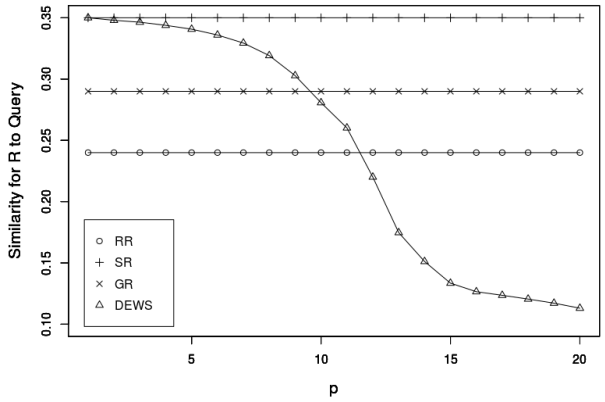

Figure 4.2: Mean similarity of the recommended set with respect to the number of clusters at $N = 20$.

same value as the one generated by SR, then decreases and converges as the number of clusters formed increases. For sanity check, the $DEWS$ should obtain the same recommended list as SR, since each rank is multiplied by the same penalty value each time, therefore the ranking of the products is preserved. The increasing and decreasing behavior in the solutions are expected as we discussed this behavior in Section 3. The red horizontal line in Figure 4.3 represents the cut off condition discussed in Section 3, where the recommendation system generates poor recommendations. It's noteworthy that there exists increasing pattern in the density improvement from Figure 4.3 and it has maximum value occurred at $p = 8$. Even though the diversity difference at $p = 8$ for

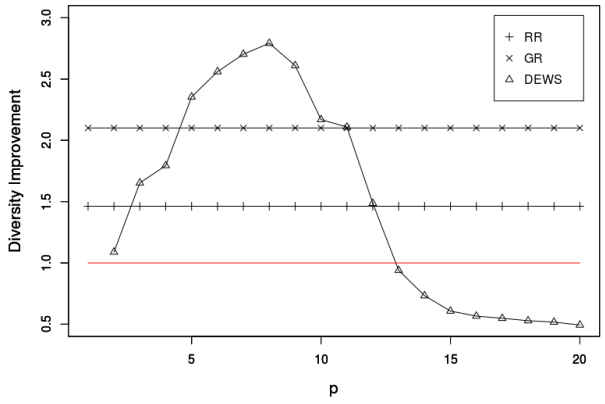

Figure 4.3: Mean diversity improvement of the recommended set with respect to the number of clusters at $N = 20$.

$DEWS$ is smaller than other two diversity enhancing algorithm, the similarity difference at $p = 8$ for $DEWS$ is also smaller than other two diversity enhancing algorithm. The accuracy of these algorithm is compared in the later analysis.

The sensitivity of number of clusters on diversity in the recommendation is evaluated again at $N = 10$, in other words, the recommender system recommends a list of 10 products to the user. The corresponding results for mean diversity, mean similarity, and mean diversity improvement are shown in Figure 4.4, Figure 4.5, and Figure 4.6. In the case of recommending 10 products

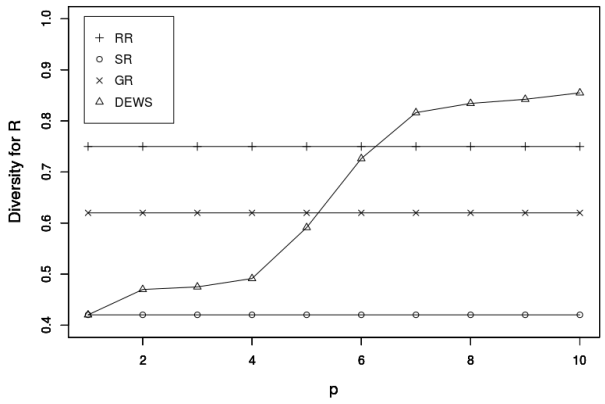

Figure 4.4: Mean diversity of the recommended set with respect to the number of clusters at $N = 10$.

to the user, the expected behavior is observed in the figures; however, the number of cluster that

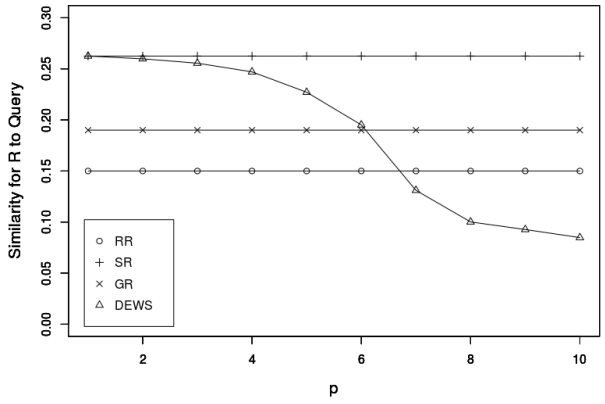

Figure 4.5: Mean similarity of the recommended set with respect to the number of clusters at $N = 10$.

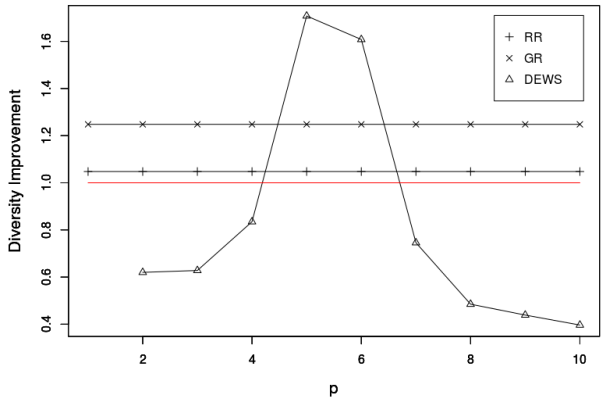

Figure 4.6: Mean diversity improvement of the recommended set with respect to the number of clusters at $N = 10$.

maximizes the diversity performance measurement becomes $p = 5$. From Figure 4.6, the diversity performance measurement is close to 0 when the cluster size is small. The behavior is expected since the 10 products associates with the highest similarity metric with the user query are relevant to each other, as discussed in Section 3.6. Hence, there is no general rule of thumb to decide the cluster size for a particular N – the size of the recommendation list. In order to find the cluster size that maximizes the trade-off between diversity and improvement, one must explore all possible cluster sizes.

From the above evaluations, it's clear that choosing different cluster size p for $DEWS$ al-

gorithm can sometimes impact on the matching precision of the recommendation. This suggests that some extra criteria to decide the cluster size p is necessary. This will be an avenue for future research.

4.3 Precision Analysis

The diversity increase in the recommendations has been shown in the last section. In the section, the precision analysis is carried out to measure the system perform of proposed algorithm in terms of precision. The system accuracy performance is measured by precision and recall metrics over all test users. Precision metric evaluates the fraction of all recommended items that are relevant, in other words, the fraction of products in the recommendations that were indeed purchased by a test user. The recall metric evaluates the fraction of all relevant items that were recommended, in other words, the fraction of products in a test user's purchase history that were included in the recommendations. N. Hurley et al. have examined the system accuracy performance when adopting the RR algorithm [4]. They showed that the random strategy RR failed to show good performance in system accuracy. Diversification of the recommendations only makes sense when it is considered in conjunction with maintaining reasonable matching quality. Hence, the system accuracy performance for RR algorithm is not further examined. In Figure 4.7, the recall metric is plotted against different size of recommendations. For each recommendations size N, the recommendations are obtained from SR, GR,and $DEWS$ algorithm. The recommendations obtained from $DEWS$ algorithm is the one that has the highest diversity performance measurement that is defined in Section 3.6. It is clear from Figure 4.7 that the recall metric for $DEWS$ algorithm is larger than the ones for SR and GR algorithm when $N \geq 2$. That is, the recommendations obtained from proposed algorithm contain more relevant items to the users. Even though the recall metric for GR algorithm is higher than the one for $DEWS$ at $N = 1$, it is not reasonable to recommend 1 product for users and the difference between these two value is relatively small. Additionally, the precision metric is plotted against different size of recommendations in Figure

4.8. It is clear from Figure 4.8 that the precision metric for $DEWS$ algorithm is larger than the ones for SR and GR algorithm when $N \geq 2$. Even though the precision metric for GR algorithm is higher than the one for $DEWS$ at $N = 1$, it is not reasonable to recommend 1 product for users and the difference between these two value is relatively small. The system accuracy performance is further examined by F_1 score. It's clear from Figure 4.9 that the F_1 score is larger than the ones when $N \geq 2$. This is consistent with the result generated from the precision and recall metrics. Even though GR obtains higher F_1 score when $N = 1$, it's not reasonable and sometime time-consuming to recommend one product to user. Therefore, the proposed $DEWS$ algorithm increases the diversity in the recommendations and performs better in accuracy than SR and GR algorithms.

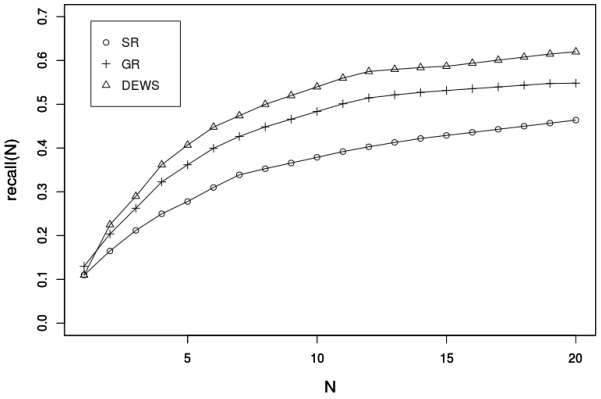

Figure 4.7: Recall metric with respect to the size of recommendations

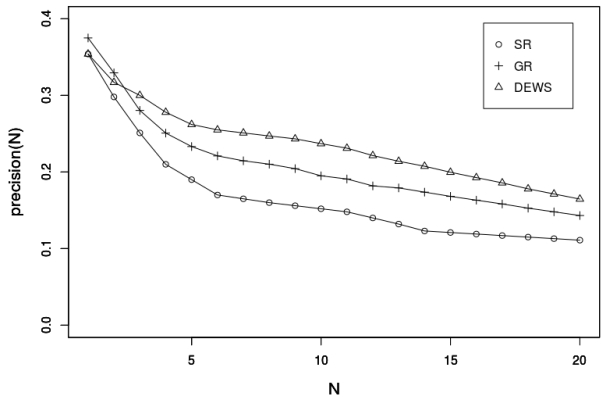

Figure 4.8: Precision metric with respect to the size of recommendations

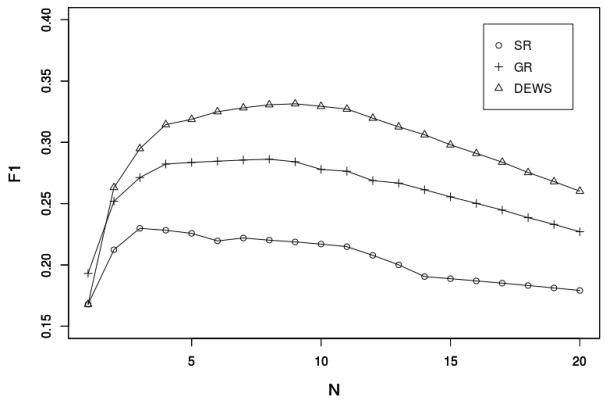

Figure 4.9: F_1 score with respect to the size of recommendations

Chapter 5

Conclusion

This thesis has shown that measuring the performance of recommendation systems solely on accuracy is not sufficient. Users will be more satisfied if the recommendations contains relevant and diverse items. While previous studies have tackled issue on increasing diversity in the recommendations, this thesis proposes an algorithm, which not only increases diversity in the recommendations but also captures the sensitivity of diversity on matching value in the recommendations. The proposed Diversity Enhancing Weighted Selection algorithm is examined with the diversity in the recommendations and the precision recall metrics. The proposed algorithm substantially improvements in both diversity and accuracy, as compared to the recommendation re-ranking approaches, which have been introduced in prior literature for the purpose of diversity improvement. In this thesis, the models for retrieving novel items in collaborative filtering recommender systems are discussed. Diversity Enhancing Weighted Selection algorithm that allows to recommend novel but relevant items have been introduced. The proposed algorithm is compared with existing algorithms and obtains a better diversity performance measure. It is worth noting that proposed Diversity Enhancing Weighted Selection algorithm may be applied to the output of

any recommendation algorithm including user-based or model-based Collaborative Filtering algorithms. Moreover, the sensitivity of cluster size on diversity in recommendations is evaluated by the diversity performance with respect to the number of clusters that formed during the process. The results have shown that the cluster size p is critical to obtaining the best diversity performance measure. The limitation in this thesis is that in order to find the best trade-off between accuracy and diversity, the diversity performance metric must be measured for different cluster size. In practical, this can be time consuming. Instead of using k-means clustering to form the clusters, one may adopt hierarchical clustering algorithm, which incorporates different cluster sizes automatically in the algorithm itself. However, since most of the hierarchical clustering algorithms is quadratic with respect to the data size (i.e. the time complexity is $O(n^2)$), while k-means clustering algorithm is linear in the number of data objects (i.e. the time complexity is $O(n)$). It is suggested to use k-means clustering algorithm in practice. The sensitiveness of the cluster size p on the diversity performance measure and evaluation time will be examined in the future research.

Bibliography

[1] Sean M. McNee, John Riedl, and Joseph A. Konstan. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, pages 1097–1101, New York, NY, USA, 2006. ACM.

[2] Barry Smyth and Paul McClave. Similarity vs. diversity. In *International Conference on Case-Based Reasoning*, pages 347–361. Springer, 2001.

[3] David McSherry. Diversity-conscious retrieval. In *European Conference on Case-Based Reasoning*, pages 219–233. Springer, 2002.

[4] Neil Hurley and Mi Zhang. Novelty and diversity in top-n recommendation – analysis and evaluation. *ACM Trans. Internet Technol.*, 10(4):14:1–14:30, March 2011.

[5] Al Mamunur Rashid, Istvan Albert, Dan Cosley, Shyong K Lam, Sean M McNee, Joseph A Konstan, and John Riedl. Getting to know you: learning new user preferences in recommender systems. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 127–134. ACM, 2002.

[6] John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[7] George Karypis. Evaluation of item-based top-n recommendation algorithms. In *Proceedings*

of the tenth international conference on Information and knowledge management, pages 247–254. ACM, 2001.

[8] Rong Hu and Pearl Pu. Helping users perceive recommendation diversity. In *DiveRS@ RecSys*, pages 43–50, 2011.

[9] Sean M McNee, John Riedl, and Joseph A Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI'06 extended abstracts on Human factors in computing systems*, pages 1097–1101. ACM, 2006.

[10] Chris Anderson. *The long tail: Why the future of business is selling less of more*. Hachette Books, 2006.

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

[12] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pages 22–32. ACM, 2005.

[13] Robin Devooght and Hugues Bersini. Collaborative filtering with recurrent neural networks. *arXiv preprint arXiv:1608.07400*, 2016.

[14] Daniel M Fleder and Kartik Hosanagar. Recommender systems and their impact on sales diversity. In *Proceedings of the 8th ACM conference on Electronic commerce*, pages 192–199. ACM, 2007.

[15] Jun Wang, Arjen P De Vries, and Marcel JT Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 501–508. ACM, 2006.

[16] Gui-Rong Xue, Chenxi Lin, Qiang Yang, WenSi Xi, Hua-Jun Zeng, Yong Yu, and Zheng Chen. Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 114–121. ACM, 2005.

[17] GP Patil and Charles Taillie. Diversity as a concept and its measurement. *Journal of the American statistical Association*, 77(379):548–561, 1982.

[18] Klaus Nehring and Clemens Puppe. A theory of diversity. *Econometrica*, 70(3):1155–1198, 2002.

[19] Douglas A Reynolds. Comparison of background normalization methods for text-independent speaker verification. In *Eurospeech*, 1997.

[20] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.

[21] Julian McAuley, Rahul Pandey, and Jure Leskovec. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2015.

[22] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52. ACM, 2015.

Haojun Sui

Education

B.S. in Computer Science, Pennsylvania State University — Graduation: May, 2017
Schreyer Honor Student

B.S. in Mathematics, Pennsylvania State University — Graduation: May, 2017
Concentrations in System Analysis Option

B.S. in Applied Statistics, Pennsylvania State University — Graduation: May, 2017

Certifications

Mathematical Contest In Modeling - Successful Participant — April, 2016
edX Verified Certificate for Introduction to Big Data with Apache Spark — July, 2015
edX Verified Certificate for Scalable Machine Learning — August, 2015

Publications

Parallel Entity Resolution with Apache Spark — Summer, 2016
- (In Review) International Journal of Grid and Distributed Computing

Projects

MarkPy, v1.1 on Mac App Store — July, 2016 - Present
- A lightweight IDE for Python scripting, which also supports user installed Python packages, such as NumPy, Matplotlib, SciPy.
- Downloads: 729

witness-syntax, v1.0.2 on apm — July, 2016 - Present
- An Atom syntax theme inspired by The Witness.
- Downloads: 57

Experience

Undergraduate Researcher, D.A.T.A Lab — Spring, 2016 - Present
- Mainly focus on social media data mining and textual mining.
- Currently working on online marketing data mining.

Research Assistant Internship, Institute of CyberScience — Fall, 2015 - Present
- Managing Apache Hadoop and Spark server on campus
- Helping university professors resolve technical issues while using Apache Hadoop and related data analysis software

DuckDuckHack Contributor, DuckDuckGo — October, 2015 - Present
- Created Instant Answer for DuckDuckGo to display prime numbers between range of numbers given by users.

Software Developer, Team BeepBeep, STATEWARE, Penn State — Fall, 2015
- Created a traffic system builder and simulator in Unity
- Implemented road system and vehicle interaction

Sofware Engineer Internship, Shanghai Software Centre of China — Summer, 2015
- Researched in Big Data Analysis using Apache Spark
- Performed Text Analysis and Entity Resolution on Amazon and Google products database
- Implemented TF-IDF algorithm, clustering algorithm, and other machine algorithm using Apache Spark and Apache Hadoop

Academic Honors

Dean's List of Distinguished Students — Fall 2013 - Present
Accepted in Schreyer Honor College in Penn State — Summer 2015 - Present

Technical Skills

Programming Languages:	C, C++, Java, Python, R, SQL, C#, Scheme, JavaScript, Scala, Swift, Objective-C
Tools & Technologies:	Apache Spark, Apache Pig, Apache Hadoop, Apache Tomcat, Minitab, Unity, Matlab
Databases:	MySQL, MongoDB