Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 12 / ISSUE 3 / SEPTEMBER 2017 / 27-49
© 2017 Universidad de Talca - Chile

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762017000300004

Fattening The Long Tail Items in E-Commerce

Bipul Kumar[1] and Pradip Kumar Bala[2]

[1] Indian Institute of Management, Operation Management and Decision Sciences, Ranchi, India, bipul.kumar12fpm@iimranchi.ac.in
[2] Indian Institute of Management, Information Systems, Ranchi, India, pkbala@iimranchi.ac.in

Received 7 June 2016; received in revised form 8 April 2017; accepted 27 May 2017

Abstract

Channelizing product sales with the aid of Recommender Systems is ubiquitous in e-commerce firms. Recommender systems help consumers by reducing their search cost by directing them to interesting and useful products. It also helps e -commerce firms by pushing the range of products a user may purchase on their e-commerce platform. The emergence of marketplace model provides platform for large fragmented buyers and sellers, where shelf space is not a constraint. Owing to unlimited shelf space, it is in the interest of e-commerce platforms to push niche products to idiosyncratic users. However, the current recommender systems, in general, recommends popular and obvious products leading to a few Long-Tail items. In this paper, our focus is on matching the niche products to idiosyncratic users such that the needs of users are satiated. We propose an innovative and robust model of matrix factorization that engenders recommendations based on a user's optimal liking of the long-tail items. We also propose an adaptive model that pursues to promote the long tail items in the recommendation list. Comprehensive empirical evaluations consistently show the gains of the proposed techniques for handling the long tail on real world data sets like Amazon dataset over different algorithms.

Keywords: Collaborative filtering, E-commerce, Long-tail, Matrix factorization, Novelty, Diversity

1 Introduction

Improving customer experience in the digital world is the prime focus of most e-commerce firms. The trend is to provide a unique shopping experience in the digital arena to every user so that overall customer satisfaction increases [31]. Recommender systems are designed to serve as important decision support systems for matching almost every customer's expectations. Recommender systems are a specific type of personalized web-based decision support systems that analyse data about customers and products to help customers find items of interest [16].

Online retailers take the advantages of unlimited shelf space over brick and mortar retailers, with the aid of recommender systems (RS), by pushing the *niche* products to *idiosyncratic* users. These niche products are mainly *non-hit* or *miss* products that account for significant sales in online platforms. It has been observed that very few items account for many *hits*, some of the items are moderately hit and most of the items account for *miss* or *non-hit*. *Miss* (non-hit) or insignificant number of hits for most of items are often referred as *the long tai*l (TLT) phenomenon in the context of recommender systems [18].

The key promise of a RS is to help the consumer discover new and relevant items outside their sphere of interest [19]. It is in the interest of both e-commerce platforms and customers to deploy such RS which increases novelty and diversity of recommendation list. Since, popular items are anyway popular, recommending such popular and obvious recommendations do not add any significant value for the customers. Recommending not so obvious yet useful recommendations adds value to the customers as well as impacts the return on investment (ROI) for e-commerce players [17]. In order to keep up the promises, RS has developed from accuracy centric DSS to other quality seeking DSS such as novelty and diversity. The qualities such as novelty and diversity are related with promotion of TLT items; promoting TLT items have positive effect on novelty and sales diversity [42]. There are several approaches that have been proposed in the literature to cater recommendations over such metrics, majority of which applies re-sorting the top-K items generated by a baseline recommendation algorithm [2], [41].

However, the previous approaches in RS have not taken into account the effect of individual customer's preferences for novel and diverse items. The previous approaches have assumed the response of every user to be uniform over novelty and diversity. However, it has been observed by empirical investigation that a user's personality influences the degree of novelty and diversity that can be enjoyed by a user in the recommendation list [17]. In other words, different users have different degree of liking/taste towards novel and diverse set of items. Incorporating a user's liking/taste for TLT items in recommendation models is a challenge which has not been attempted till date to the best of our knowledge.

Keeping in mind the above challenge of modelling a RS, we propose an extension of matrix factorization model, which is state-of-the-art algorithm in rating prediction [44], [49].The matrix factorization technique maps the latent features of both items and corresponding features of users in the joint latent factor space of dimensionality *s*uch that inner product of user-item interactions are modelled in the latent space [24]. In an attempt to incorporate customer taste/liking for novel and diverse set of items, this work introduces a parameter to capture the degree of liking of each user towards novelty of items. Novelty of an item acts as a surrogate measure for long-tail. Further, as explained in [42], promotion of long-tail items positively impact the diversity of recommendation which has also been validated in this work. Matrix factorization approach has been suitably extended to capture the impact of the incorporated parameter to measure the user's tastes of TLT items. The optimization of the extended matrix factorization model ensures that optimal liking of every user for long tail items is realised and unseen items are recommended to every user based on his/her interest of long-tail items.

If firm choses to aggressively promote niche items, so that diversification in sales help the firm to grow, a suitable strategy could be targeting only those customers who have previously shown inkling towards such items. For targeting such customers the model parameters can be trained so that a user's disposition can be suitably improved towards long-tail items. Further, we have also focused on promoting the number of long-tail items and diversity of recommendation list in the proportion of user's taste for long tail items. This is accomplished by training the user's taste in such a manner that every user would more often like long tail items. This would facilitate the RS to be flexible and recommend more of long tail items to idiosyncratic users.

The remainder of our paper is organized as follows. In Section 2, we review the related works and identify the research gap. Section 3 describes the basic notations and proposed models that accounts for taste of long-tail items for each user and promote long-tail items to idiosyncratic users. In section 5, the proposed models are experimented on two publicly available benchmark datasets with results reported and validated. Finally, section 6 discusses the results of proposed model and suitable conclusions are drawn based on discussions.

2 Related Work

The Long Tail, notion was introduced by Chris Anderson in his book which described about a couple of conditions to exploit the content available in niche segments [6]. These are: (i) make everything available, and (ii) help me find it. Due to e-commerce platforms, the former seems to be fulfilled as costs incurred by e-commerce for distribution and inventory are nearly negligible and therefore most of the items are available online. The focus, therefore, now is on later condition and the question to be asked is, do RS in the current shape helping to find the niche products? There is an on-going debate whether RS generate niche items or popular items [13]. In most of the studies it is found out that with the use of RS, more popular items are being recommended in video sales and benefits as expected from *long tail* phenomenon has not been realized [13].

Accuracy is a preferred measure for evaluating RS but it has been argued that accuracy alone is not enough to measure the performance of RS [30]. RS can be useful if it helps users in discovering new items and therefore increase the sales diversity of E-commerce platform. Sales diversity can be achieved by recommending diverse items to users. Diversity of recommendation list can be classified into two ways, individual diversity and aggregate diversity. Individual diversity of recommendations for a user can be measured by calculating average dissimilarity between all pairs of item that has been recommended to a user [46]. Aggregate diversity, on the other hand, is calculated across all users which is measured by total number of distinct items among top-N items recommended across all user [3].

Similar to diversity, novelty is another metric that was introduced in order to measure the impact of RS. Novel items are those which a user does not know about [21]. One of the works related to reducing the error rate of long-tail items proposed partitioning of the whole item set into the head and the tail parts and clusters only the tail items. Then recommendations for the tail items are based on the ratings in these clusters and for the head items on the ratings of individual items. If such partition and clustering are done properly, the authors show that this reduces the recommendation error rates for the tail items, while maintaining reasonable computational performance [33]. Serendipity and coverage as performance metrics have also been introduced in literature apart from novelty and diversity in order to capture the performance of RS [15], [38].

There are a few papers which describe the approaches of diversity used in recommender systems. One approach of solving this problem was by introduction of intra-list similarity metric and topic diversification for recommendation lists [30]. Intuitively, diversity and accuracy are thought as a trade-off between each other. This means diversity can be achieved at the expense of accuracy and vice-versa. In another approach variance based approach was introduced in order to solve the accuracy-diversity tradeoff. Firstly, k-nearest neighbor approach is adopted for predicting the rating of an unseen item by an active user followed by ranking of items for a user based on variance of k neighbor's rating for unseen items [40]. Diversity metric in this paper was taken as total number of distinct items recommended across all users [40].

There is another view of aggregate diversity metric or average diversity metric in the literature. It can be defined as dissimilarity between recommended items based on distance between feature vectors of user ratings. In order to solve accuracy- diversity tradeoff in this case a new strategy was introduced. It is a two-step process, in the first step recommendation was generated based on popular algorithm like k-nearest neighbour and in the second step a quadratic optimization function is used in order to filter items having high diversity in the recommendation list [46]. Similar to the above approach of optimization between accuracy and diversity, a Genetic Algorithm (GA) was proposed to maximize accuracy, novelty and diversity simultaneously by hybridizing various popular models in RS [36]. The basic approach in the method is to search for Pareto-optimal hybrids using evolutionary search technique in GA [36].

Continuing with improving aggregate diversity of a RS, re-ranking approach has also been proposed in contrast to the optimization approach. The first step is to optimize ratings based on popular models and then adopting a re-ranking method to generate top-k recommendation in such a way that diversity and novelty are catered along with accuracy. Several approaches of re-ranking the items have been discussed by Adomavicius & Kwon [2]. Other approaches such as *item average rating* re-rank the recommendation list according to an average of all known ratings; *item absolute likeability* rank item according to number of likes (rating greater than some prefixed threshold) by the users in the database, *item relative likeability* rank items according to percentage of likes, *item rating variance* rank items according to rating variance of each item, and lastly *neighbors rating variance* rank items according to the rating variance of neighbors of a particular user by particular item. Further, aggregate diversity is measured according to three index viz. Entropy, Gini coefficient, and Herfindhal index. The proposed re-ranking approach shows a direction of solving accuracy-diversity dilemma [2].

Taxonomical based diversity measure was also introduced which captures the distance between items based on the contents as features e.g. genre, language, actor, director, etc. [52]. The authors also proposed topic diversification algorithm in order to generate a recommendation list which is diverse and more useful for a user [52]. A different perspective on enhancing diversity and diversity measure has been proposed in reference [45]. The author borrows the concept of *concentration index* from economics and applies to RS. *Concentration index* is calculated by first plotting a curve of the cumulative proportion of items in user profile against cumulative proportion of successful recommendations achieved on these items and then finding the difference between area below the diagonal and the area below [45], [47]. In order to generate a diverse recommendation list three strategies have been laid out by the author. The first strategy is to optimize an objective function that consists of diversity and accuracy as maximization function, after a recommendation list is generated based on accuracy. The second strategy is to cluster items and then match the items of various clusters with the user's profile based on SUGGEST algorithm [45], [48].The third strategy is to apply singular value decomposition over item profiles before clustering the items. The next step follows the same as second strategy [45]. A few other algorithms have also tried to address the diversity-accuracy dilemma. In one of the algorithms the paper describes a hybrid method of generating accurate and diverse recommendations [50]. The authors introduced a heat-spreading (HeatS) algorithm and combined with probabilistic spreading (ProbS) using a parameter λ to generate diverse and accurate recommendations [50].

A conference (DiveRS) was entirely dedicated on novelty and diversity in Recommender systems and was held in conjunction with association of computer machinery (ACM) in the year 2011. In the first paper presented in the conference, [3] proposed a graph-theoretic approach for maximizing aggregate diversity. A new concept of *unexpectedness* has also been proposed which is different than novelty, diversity and serendipity [1]. Expectedness is defined as the set of items that the user is thinking of as serving his current needs or fulfilling his intentions indicated by visiting the RS [1]. Unexpectedness metric has been derived from the definition of expectedness by inculcating some unimodal function of the distance between items. Authors have argued that unexpected items without its utility do not add value to RS, so a metric which combines both unexpectedness and utility has been proposed.

Since it is still a growing field no consensus has been drawn on the metric formulation of diversity and novelty [41]. In order to formulate a formal definition of diversity and novelty in context of RS, [41] proposed metrics based on different perspective. Novelty and diversity were defined by taking two notions as a base for building the metric. The two notions are popularity of items and similarity of items. Further, based on user-item relationship the metrics can be modified. Choice, discovery and relevance are the three user-item relationship which can exist and based on these user-item relationship different novelty and diversity metric can be formulated [41]. One of the recent approaches to recommend diverse and novel item has been proposed taking into account the social relationship among the users [14].

Based on the above literature surveys, we find few interesting insights that will be used as the base in building the proposed model.

The first key insight is about novelty in recommendation list; where of long-tail items is a common way in which novelty is understood [41]. Therefore, novelty can be taken as a surrogate measure for long-tail. We have amended this measure by taking into account the time dependent aspect of measuring novelty as item novelty is time variant. In other words, with passage of time a novel item may turn up to be a popular item or vice-versa owing to increased/decreased preferences of users. Introducing time dependent modelling of novelty is an important contribution to the present literature on long-tail RS research.

The second key insight is the relationship between novelty and diversity of recommendation list. Novelty and diversity are different though related notions. The novelty of a piece of information generally refers to how different it is with respect to *what has been previously seen*, by a specific user, or by a community as a whole. Diversity generally applies to a set of items, and is related to how different the items are with respect to each other. This is related to novelty in that when a set is diverse, each item is *novel* with respect to the rest of the set. Moreover, a systems that promotes novel recommendation tends to generate global diversity over time in the user experience; and also enhances the global *diversity of sales* from the systems perspective [41].

The third and important insight is that the existing algorithms that address long-tail items have assumed the uniform response of every user irrespective of his liking on the aspect of novelty and diversity. However, it would be naïve to assume that each user will have same response towards novel and/or diverse items. It is worth mentioning two empirical studies to corroborate the above finding. In one of the empirical studies, it is found that personality has an impact on choice of diverse products [12]. More specifically, reference [12] established significant correlations between some personality values and diversity of item. It also provides an instance that; the choice of *director* is significantly positively correlated with the personality factor neuroticism, which suggests that more reactive, excited and nervous person is more inclined to choose diverse directors. This implies that the inclination towards diversity and novelty will vary from individual to individual. The inference of this concept is vital for the proposed model as the proposed model takes into account the variation of taste for novel/diverse items at individual level and not at the personality level. This is also validated by another empirical study. The second empirical study looks into diverse domains-movies, music, web search, and web browsing. The findings suggest that, some users draw disproportionately from the head (popular) while others draw disproportionately from the tail(novel) [17]. Thus, we can conclude from above findings that indeed different users accord different weightages for novelty/diversity in e-commerce platforms. This forms the basis of the research problem embodied in the current work. The proposed models in the current research work are based on above identified insights of individual preferences for long-tail and popular items.

3 Material and Method

The aim of this paper is to promote long tail items by taking care of the responses of the user for long-tail products captured in their prior transactions, without compromising the accuracy of RS. The long-tail phenomena mostly occur in all e-commerce platforms, viz., movie, videos, songs, books etc. To propose a model that accounts users' interest for long-tail items, benchmark databases of MovieLens and Amazon are used in the paper. Long-tail items are barely rated by users and in extreme cases they may not be considered by users at all. To delimit the scope of this paper, we have assumed that long-tail items are rated by at least one user to develop a meaningful model. This restriction is binding for any matrix factorization scheme as there must be co-occurrence patterns between users and items. The propose model also adopts the same restriction as is on matrix factorization. Based on this periphery, the proposed models introduce the concept of optimal promotion of TLT products. For this purpose, we use the user-item rating matrix $'R'$ and novelty matrix $'\tau'$ to first train the model and later use the parameters of the trained model on unseen items to predict their ratings. The detailed model is described in subsequent sections.

3.1 Proposed Models

This section proposes two models, the first model learns from the historical behavior of users towards long-tail items and recommends items to users in a proportion similar to historical records. The second model learns by augmenting the behavior of users towards long-tail items and then recommends more of long-tail items to users. This sub-section starts with description of notations being used in the model building process.

3.1.1 Notations

With many users and large number of products in the database, it is convenient to represent such database in form of a matrix. So we will consider a user-item matrix $R \in \mathbb{N}_o^{n \times m}$, with n rows and m columns, whose n rows correspond to users and m columns correspond to available products. The cells of this matrix R are ratings provided by a user corresponding to the items. These ratings represent the preferences of a user for items rated by him. Generally, these ratings are in range of 1 to 5, where 1 is the rating provided on least preferred items and 5 is the rating provided for the most preferred item. In the matrix form, every user is identified by its row index, $u \in \{1, \dots, n\}$, every product by its column index, $i \in \{1, \dots, m\}$ and r_{ui} stands for the rating given by user u for item i. In order to capture the novelty of an item at the time of rating by a user we will also consider another matrix $\tau \in \mathbb{N}_o^{n \times m}$ which is similar to matrix R, the only difference being the value of cells in the matrix. The values of the cells in the matrix τ represent the novelty of items at the time when these items are rated. The novelty of an item is calculated using formula described in sub-section item novelty.

3.2 Overcoming the Long-tail

In this section, we propose an algorithm that learns the taste of a user for long-tail items by optimizing measure of long-tail of an item and corresponding ratings of item by a user. The above formulation is inspired by simple linear regression problem where the aim is to find the parameter that optimizes dependent and independent variables. Drawing analogy from linear regression problem, the dependent variables are rating vector of a user and independent variables are long-tail measure of items which are also a vector. The model that is based on the above algorithm may prioritise novel items in recommendation at an expense of accuracy. Since, accuracy is also an important aspect of a recommendation algorithm, we will incorporate Regularized singular value decomposition (RSVD) model into the proposed scheme. Lastly, we have also considered training parameters of proposed model such that it is adaptive to promote long-tail items at the time of requisite.

3.2.1 Item Novelty

Before developing a model that optimizes the taste of a user with long-tail items, we need to first develop a measure of long-tail. As noted in [10], [42], novelty and diversity are stimulated by promotion of long-tail items, it is prudent to take item novelty as a surrogate to depict the long-tail measure of items. Also, item novelty measures have been developed in previous works; we can use those measures for the current proposed model.

In the context of information retrieval, the novelty of a retrieval set has been defined as the proportion of known and unknown relevant items in the recommended list with respect to the end-user [7]. Similarly, in the case of RS, item novelty has been defined as log of inverse of popularity [10], [51].

$$\text{novelty(i)= } \log_2\left(\frac{1}{\text{p(i)}}\right) \tag{1}$$

Here, p(i) represents popularity of an item.

$$\text{p(i)=}\frac{\text{\# users prefering item i}}{\text{\#total users}} \tag{2}$$

Popularity is the ratio of number of users who have considered the item as relevant to the total number of users in the database.

Implicit in the concept of novelty is the point of time when novelty is being considered, as both the number of users considering the item relevant and total number of users change with time. This means that with passage, of time the novelty of an item may change. An item which is moderately popular or moderately novel at the time of the introduction in e-commerce database may be very popular or not so novel after passage of certain months. So novelty is a function of time and therefore, to make the representations clearer, we have used subscript $'t'$ with item i.

$$\text{novelty}(i_t)=\log_2(1/p(i_t)) \tag{3}$$

The novelty values can be represented in a matrix $\tau \in \mathbb{N}_o^{n \times m}$ corresponding to each rating provided by a user for an item at time. The dimension of this matrix is same as the matrix R and the novelty values are also in the corresponding cells as those of matrix R.

Feature scaling of variables ensures that there is no domination of one variable over another in machine learning algorithms. So, we will first scale the item novelty values obtained from above expression before using it for integrating novelty in the proposed model. For scaling, we can change the base of logarithm used to calculate novelty such that the resultant value lies between 0 and 1. The steps to be followed in feature scaling are:

1. Calculate $(1/p(i_t))$ f item i at all instances of time t (timestamp of rating provided by a user for an item).
2. Obtain minimum of $(1/p(i_t))$ calculated in step-1, which is denoted by min $(1/p(i_t))$.
3. Divide each $(1/p(i_t))$ by min $(1/p(i_t))$
4. The value of $(1/p(i_t))$ corresponding to maximum value in step-3 is taken as the $base$ of the logarithm.
5. Novelty is calculated by the following formula.

$$\text{novelty}(i_t^N)= \log_{base}(\frac{1}{p(i_t)}) \tag{4}$$

I he resultant of the above is feature scaled value of item novelty $\text{novelty}(i_t^N)$ for each item at every instant when the rating is provided by a user for an item. A feature scaled novelty matrix, $\tau_{scaled} \in \mathbb{N}_o^{n \times m}$ is obtained using the resultant feature scaled values. The calculation of the novelty values may be done depending on the number of users and/or items added to the database. A weekly training of CF algorithm are generally carried out for prediction [26], a similar strategy may be employed in novelty calculation.

The below toy example illustrates the calculation of novelty values of items at different instant of time. It is important to note that e-commerce platforms report UNIX timestamp of a transaction log, therefore, for all practical purposes timestamp corresponding to each rating is used in model building. Table 1 presents a toy example with user, item, rating and timestamp. It may be noted that smaller values of timestamp mean early occurrence and greater values of timestamp means late occurrence. Table 2 represents the user-item rating matrix obtained from table1.

Table 1: A toy example for user-item rating

User ID	Item ID	Rating	Timestamp
U1	I1	5	874965758
U1	I3	3	876893171
U2	I2	5	878542960
U2	I4	2	876893119
U3	I1	4	889751712
U4	I2	3	875071561
U5	I3	5	875072484

Table 2: Representation of user-item rating matrix

User/items	I1	I2	I3	I4
U1	5		3	
U2		5		2
U3	4			
U4		3		
U5			5	

In the above toy example, if there are only 5 users in the database, item I1 has been rated by user U1 and U3 at t=874965758 and 889751712 respectively. To calculate featured scaled novelty of I1 following steps may be applied.

1. Calculate popularity at t=874965758 and 889751712

$$\mathrm{p}(\mathrm{I1}_{874965758}) = \frac{\text{\# users prefering item I1 till time t1}}{\text{\#total users}} = \frac{1}{5} \;\&\; \mathrm{p}(\mathrm{I1}_{889751712.}) = \frac{2}{5}$$

2. Min $\{ (\frac{1}{\mathrm{p}(\mathrm{I1}_t)}), (\frac{1}{\mathrm{p}(\mathrm{I2}_t)}) \}$ =Min { 5, 2.5} =2.5

3. Divide $(\frac{1}{\mathrm{p}(\mathrm{I1}_t)})$ by min $\{ (\frac{1}{\mathrm{p}(\mathrm{I1}_t)}), (\frac{1}{\mathrm{p}(\mathrm{I2}_t)}) \} = \frac{5}{2.5}$=2 and $(\frac{1}{\mathrm{p}(\mathrm{I2}_t)})$ by min $\{ (\frac{1}{\mathrm{p}(\mathrm{I1}_t)}), (\frac{1}{\mathrm{p}(\mathrm{I2}_t)}) \} = \frac{2.5}{2.5}$=1

4. Logarithm base is corresponding maximum value of $(\frac{1}{\mathrm{p}(\mathrm{i}_t)})$ =5

5. $\mathrm{novelty}(\mathrm{I1}_{874965758}) = \log_{\mathrm{base}}\left(\frac{1}{\mathrm{p}(\mathrm{I1}_{874965758})}\right) = \log_5 5 = 1$

   and $\mathrm{novelty}(\mathrm{I2}_{889751712.}) = \log_{\mathrm{base}}(\frac{1}{\mathrm{p}(\mathrm{I2}_{889751712.})}) = \log_5 2.5 = 0.57$

It may be noted that the featured scaled novelty values obtained in such a manner are normalized between 0 and 1. Similar steps can be followed for other items in database. A novelty matrix, table 3, can thus be obtained after calculating novelty values in above manner for all items at each instant of time.

Table 3: Resultant novelty matrix

	I1	I2	I3	I4
U1	1 (874965758)		0.57 (876893171)	
U2		0.57 (878542960)		1 (876893119)
U3	0.57 (889751712)			
U4		1 (875071561)		
U5			1 (875072484)	

Having obtained the novelty matrix, the next step is to integrate it with matrix factorization. The next subsection briefly introduces the matrix factorization as applied for rating prediction which will be used to build two models, viz. PM-1 and PM-2. PM-1 integrates matrix factorization with novelty to recommend items based on prior disposition towards novel set of items. On the other hand, PM-2 promotes long-tail items in the recommendation list.

3.3 Basics of Matrix Factorization (RSVD Model)

Matrix factorization is a technique of decomposing a matrix into several matrices depending upon the suitability of the context. In this section, we will introduce the basics of Regularized singular value decomposition (RSVD), a variant of MF that was a part of the winner's solution in Netflix competition [23]. The beauty of RSVD over other collaborative filtering algorithms is the ability of RSVD in handling sparser matrices. The basic idea incorporated in RSVD is that users and items may be described by their latent features. Every item can be associated with a feature vector (Q_i)

which describes the type of product e.g. convenience vs. specialty, durable vs. non-durable, etc. Similarly, every user is associated with a corresponding feature vector (P_u). The inner product between user feature vector and item feature vector is approximated as the predicted rating given by a user u for an item i. Mathematically, it can be expressed as:

$$\hat{r}_{ui} \approx P_u Q_i^T \tag{5}$$

Along with latent features, the inherent characteristic of user and item also contribute to the rating of an item by a user. For e.g.: a functional product may always be rated higher than average rating while an innovative product may always be rated below average. Similarly, a demanding user tends to rate on the lower side while a docile user may rate on higher side. These inconsistencies are called as bias of user and item and have to be captured in a model. Therefore, user bias (b_u) and item bias (b_i) are added to the model given in equation [5]. User bias (b_u) is the observed deviation of a user u from average rating of all users. Item bias (b_i) is the observed deviation of item i from average rating for all items. An overall mean of ratings represented by μ is also added to the model to capture the bias of the dataset. The resulting model is given by following equation.

$$\min_{P_*\,Q_*} \sum_{(u,i\,\in\kappa\,)} \left(r_{ui}-\mu-b_u-b_i-P_u Q_i^T\right)^2 + \lambda(\|P_u\|^2_{Fro} + \|Q_i\|^2_{Fro} + \|b_u\|^2 + \|b_i\|^2 \tag{6}$$

Here, κ is set of known ratings in matrix R and $\|\cdot\|_{Fro}$ denotes the Frobenius norm. The parameter $\lambda > 0$ is a regularization parameter and is used to avoid over fitting of the model. Over fitting is a common term in machine learning algorithm which suggests that the model will perform very well in te training set but will perform very bad in test set upon which the model is not trained. Training of the model is done only on the ratings available in the user-item matrix while the missing values in the matrix are skipped during training.

3.4 Integrating Novelty with Matrix Factorization (PM-1)

Before delving into formulations that integrate novelty with RSVD, it is prudent to develop baseline estimates that capture the characteristics of the database. Baseline estimates help in countering the item and user effect. [22] We have adopted the baseline estimates that are widely used in matrix factorization formulation. Baseline estimates capture bias of a user (b_u) towards item, item bias (b_i) and overall bias (μ) of the dataset. The baseline estimate b_{ui} is formulated using the following equation.

$$b_{ui}=\mu+b_u+b_i \tag{7}$$

Further, we introduce an optimization model that is formulated by taking into account the novelty of items with respect to every user. The model assumes that there is a linear relationship between rating of an item rated by a user and corresponding item novelty. Using this assumption, we will gradually construct the various parameters of the model, through an on-going refinement in the formulations. We represent the initial formulations by following equation, where f denotes a function.

$$r_{ui}=f(\text{novelty}(i_t^N)) \tag{8}$$

To formulate a definitive relationship, we will also assume that there is a linear relationship between item novelty and rating given by a user. The parameter β_u will be used to formulate a linear relationship which can be interpreted as the weight, a user ascribes to items' novelty. Since, the weight ascribed with items' novelty would be different for different user therefore, parameter β_u has a subscript u. In order to capture influence of novelty and the biases attached with users and items, we formulate an additive scheme that integrates baseline estimates with the normalized item novelty. The formulations are like a linear regression problem and are often used to solve problem in RS [4]. The predictors are the novelty values of items, while responses are the rating provided by each user for items. The coefficient of linear regression is determined by minimizing the sum of squares of errors which establishes a relationship between predictors and responses. Similar approach of linear regression has been proposed to estimate the parameter β_u in the following model.

The resultant additive prediction scheme for estimating the rating of an unseen item can be solved by estimating the parameters of the following model. The parameters can be estimated by solving the following optimization problem.

$$\min_{b_*\,\beta_*} \sum_{(u,i\,\in\kappa\,)} \left(r_{ui}-\mu-b_u-b_i-\beta_u(\text{novelty}(i_t^N))\right)^2 + \lambda_1(\,\|b_u\|^2+\|b_i\|^2+\|\beta_u\|^2) \tag{9}$$

The above optimization problem can be solved by using stochastic gradient descent (SGD) algorithm. The item novelty is pre-computed for unseen items, and the parameters so obtained using SGD will predict the rating of an unseen item. This formulation is an innovative idea in RS to optimize the ratings given by a user for an item with the corresponding novelty of the items.

However, optimizing the model solely based on novelty may result in diminished accuracy of RS. Therefore, integrating this model with RSVD can fetch a solution that would result in optimal novelty without compromising on accuracy. A simpler scheme to integrate novelty with MF is to generate an additive model.

This additive model is simpler yet effective in prediction of ratings for unseen items with optimal novelty without compromising on accuracy. In order to formulate an additive scheme with RSVD we need to add user item interaction ($P_u Q_i^T$) from equation [5] with equation [9]. For estimating the parameters of the above model, we have to solve the following optimization problem.

$$\min_{b_*\ \beta_*\ P_*\ Q_*} \sum_{(u,i\ \in \kappa\ )} \left(r_{ui}-\mu-b_u-b_i-P_u Q_i^T-\beta_u(\text{novelty}\left(i_t^N\right))\right)^2+\lambda_2(\ \|b_u\|^2+\ \ \|b_i\|^2\ \ +\left\|\beta_u\right\|^2_{Fro}+\|P_u\|^2_{Fro}+ \|Q_i\|^2_{Fro}) \tag{10}$$

The integrated additive model proposed above can be seen as combination of three stages. In the first stage, $\mu+b_u+b_i$ refers to a general description about user and item without accounting for any other interaction. The second stage, $P_u Q_i^T$ describes the interaction between features of items and corresponding features of users. The final stage, $\beta_u(novelty(i_t^N)$ calls for item novelty in the model which has to be optimized with respect to the taste of each user corresponding to the item.

$$\hat{r}_{ui}=\mu+b_u+b_i+P_u Q_i^T+\beta_u(\text{novelty}\left(i_t^N\right)) \tag{11}$$

The above optimization learns the latent features of users (P_u) and items (Q_i), user bias (b_u) and item bias (b_i), and parameter β_u by minimizing the sum of square losses with respect to corresponding rating. The rating predictions of unseen items are obtained using following equation.

Based on predicted ratings, top k items to each user is presented and these are recommendations for a user taking into account the taste of long-tail items and accuracy of the model.

3.5 Fattening the Long Tail (PM-2)

In the previous section we proposed a model that makes an attempt in recommending items to idiosyncratic users based on their taste of long tail items. However, it is often valuable for both users and sellers of e-commerce platform to recommend more of long-tail items [39]. To make this feasible we developed a new model of RS that endeavours recommending long-tail items to such users which have positive inclination to long-tail items. The basic idea of the proposed model is to train the parameters of the model on a given dataset such that the long-tail items are promoted to users having positive interest in long-tail items.

For training such model, we can think of training the model in stratified manner. It means that the same model shall be adaptive to train in such a manner that novel but relevant items are more emphasized while other items are trained with their implicit significance. This will ensure that the parameters of users are learnt in the scheme so that long-tail items are given better rating that may be used to rank the items for generating recommendation for each user. One such adaptive scheme can be formulated by extending the model described in the above section. Let us consider the following adaptive formulation.

$$\min_{b_*\ \beta_*\ P_*\ Q_*} \Sigma_{(u,i\ \in\kappa\ )}\left(r_{ui}*\alpha-\mu-b_u-b_i-P_u Q_i^T-\beta_u(\text{novelty}\left(i_t^N\right))\right)^2+\lambda_3(\ \|b_u\|^2+\ \|b_i\|^2+\left\|\beta_u\right\|^2+\ \|P_u\|^2_{Fro}+\ \|Q_i\|^2_{Fro}) \tag{12}$$

Where, $\alpha=\ \alpha>1\ ;\ \forall\ \theta_N > \theta$ and

$\alpha=1\ ;\qquad \forall\ \theta_N\leq\theta\ ;\ \theta$ is a prefixed value of item novelty ($\text{novelty}\left(i_t^N\right)$) for relevant rating ($r_{ui}$) to be used in determining long-tail and relevant items.

Here α is taken as discrete numeric constant and can be interpreted as training factor associated with stratified long-tail and popular items. Stratified training is a key to formulate the scheme so that relevant long tail items are boosted while the corresponding parameters of users are adjusted automatically in the learning process of the model. For a simpler scheme, we will use α as a discrete value greater than 1 for items which are in long tail and value equals 1 for items other than in long tail. Long tail items can be defined by a prefixed item novelty threshold value (θ_N). For identified relevant novel items training factor (α) greater than 1 implies that the rating is boosted by α times the original rating. For e.g. if an item that is relevant and novel has original rating of *4* may be boosted to *4.8* when training weight is taken as *1.2* while, if the training weight is taken less than *1* (say 0.8) then the novel and relevant items will be diminished to *3.2* from the original rating of *4*. Therefore, the value of training weight (α) must be chosen in such a way that it tends to boost the rating rather than diminish it in order to fulfil our objective of promoting long tail items.

The key idea of this formulation is to train latent features of users and items as well as training parameter β_u such that long-tail items are more preferred than popular items. The training weight (α) is multiplied to ratings of the long-tail but accurate items. This ensures that these ratings are boosted from actual rating, which thereafter trains latent features and β_u with preference of long-tail.

In order to learn the parameters of the algorithm, stochastic gradient descent optimization is applied which was popularized by Funk and successfully practiced by many others [35], [37], [43]. The algorithm loops through all ratings in the training data. For each given rating r_{ui}, a prediction $\hat{r}_{ui}$ is made, and the associated prediction error $e_{ui} \stackrel{\text{def}}{=} r_{ui} - \hat{r}_{ui}$ is computed. The parameters associate with the algorithm can be learnt by moving in the opposite direction of the partial gradient corresponding to each parameter of the following squared error.

$$\left(r_{ui}*\alpha-\mu-b_u-b_i-P_uQ_i^T-\beta_u(\text{novelty}(i_t^N))\right)^2+\lambda_3\left(\|b_u\|^2+\|b_i\|^2+\|\beta_u\|^2+\|P_u\|^2_{Fro}+\|Q_i\|^2_{Fro}\right) \quad (13)$$

The sum of the squared error is computed by summing the above squared error for each rating available in the training data. The summation thus generated for each iteration can be represented by equation 13. The key idea is to learn the parameters of the loss function in each iteration till the sum of the squared error converges.

$$\text{error (step)} = \sum_{(u,i\ \in\kappa\ )}\left(r_{ui}*\alpha-\mu-b_u-b_i-P_uQ_i^T-\beta_u(\text{novelty}(i_t^N))\right)^2+\lambda_3(\|b_u\|^2+\|b_i\|^2+\|\beta_u\|^2+\|P_u\|^2_{Fro}+\|Q_i\|^2_{Fro}) \quad (14)$$

Algorithm: To generate a recommendation list for each user by promoting optimal long-tail items.

Variables (symbols and their denotation used in the algorithm)

- R : A matrix of rating, dimension N x M (user item rating matrix)
- r_{ui} : Actual rating of an item i by a user u
- $\hat{r}_{ui}$: predicted rating for an item i by a user u
- κ : Set of know ratings in matrix R
- P_u : A matrix of dimension N x l (User feature matrix) (to be initialized by a random value between 0 and 1)
- Q_i : A matrix of dimension M x l (Movie feature matrix) (to be initialized by a random value between 0 and 1)
- b_u : Bias of user u. (to be initialized by a random value between 0 and 1)
- b_i : Bias of item i. (to be initialized by a random value between 0 and 1)
- μ : Average rating of all users. (to be deduced from the dataset)
- l : Number of latent features to be trained. (input by the user and can be varied)
- τ_{scaled} : A matrix of normalized item novelty value. (to be calculated from the dataset)
- β_u : Novelty weight. (to be initialized by a random value between 0 and 1)
- e_{ui} : error between actual and predicted rating (computed inside the algorithm)
- $error\ (step)$: sum of squared error at each iteration. (computed inside the algorithm)
- α : Training factor (input by user; it is taken as (1.2, 1.4, 1.6, 1.8, 2))

Parameters:

- η: learning rate (input by user taken as 0.001)
- λ_2 : over fitting regularization parameter (input by user taken as 0.01)
- step: Number of iterations

Output: A recommendation list promoting optimal long-tail items for active users

```
Method:
1. Initialize random values to matrix $P_u$, $Q_i$ and vectors $\beta_u$, $b_u$and $b_i$
2. Fix value of l, η, α and $\lambda_2$.
3. error (step) = $\sum_{(u,i \in \kappa)} \left(r_{ui} * \alpha - \mu - b_u - b_i - P_u Q_i^T - \beta_u (\text{novelty}(i_t^N))\right)^2 + \lambda_3 ( \|b_u\|^2 + \|b_i\|^2 + \|\beta_u\|^2 + \|P_u\|^2_{Fro} + \|Q_i\|^2_{Fro}$.
4. do till error converges [ error(step-1) - error(step) < ε ]
5. for each $R \in \kappa$
       Compute $e_{ui} \overset{\text{def}}{=} r_{ui} - \mu - b_u - b_i - P_u(\cdot) Q_i^T - \beta_u(\text{novelty}(i_t^N))$
       Update training parameters
       $b_u \longleftarrow b_u + \eta(2e_{ij} - \lambda_2 b_u)$
       $b_i \longleftarrow b_i + \eta(2e_{ij} - \lambda_2 b_i)$
       $P_u \longleftarrow P_u + \eta(2e_{ij} Q_i^T - \lambda_2 P_u)$
       $Q_i \longleftarrow Q_i + \eta(2e_{ij} P_u - \lambda_2 Q_i)$
       $\beta_u \longleftarrow \beta_u + \eta(2e_{ui} \text{novelty}(i_t^N) - \lambda_3 \beta_u)$
6. endfor
7. end
8. return $P_u$, $Q_i$, $\beta_u$, $b_u$and $b_i$
9. for each $R \notin \kappa$ predict the ratings for user and item
   $\hat{r}_{ui} = \mu + b_u + b_i + P_u(\cdot) Q_i^T + \beta_u(\text{novelty}(i_t^N))$
10. rank and return top-k predicted ratings for each user in a descending order
```

4 Experimentation and Evaluations

There are two main objectives to be established by experimentation, viz., i) providing personalized recommendations to every user by taking into account their respective taste for long-tail items, and ii) promoting long-tail items to idiosyncratic users. The first objective is fulfilled by applying PM-1 model and the second objective is fulfilled by applying PM-2 model. The experimentation establishes the first objective by comparing the values obtained by PM-1 on accuracy, novelty and diversity with the values obtained by state-of-the-art RSVD model. Further experimentation establishes the efficacy of PM-2 on promoting long-tail item to idiosyncratic users by varying the values of training factor and threshold item novelty values. The measure to evaluate the occurrence of long-tail items has been introduced in section 4.1.2. Since, there is a positive relation between promotion of long-tail and diversity of the recommendation list, while negative relation with precision of recommendation, we have verified these two relations. The measures used for measuring precision and diversity have been introduced in section 4.1.1 and 4.1.2 respectively.

For offline evaluation of the experimentation, we make use of two different benchmark datasets. The first one is a publicly available Movie Lens dataset (ml-100k). The dataset consists of ratings of movies provided by users with corresponding user and movie IDs. There are 943 users and 1682 movies with a total of 100000 ratings in the dataset. Had every user would have rated every movie total ratings available should have been 1586126 (i.e. 943×1682); however only 100000 ratings are available which means that not every user has rated every movie and dataset is very sparse (93.7%). Also, corresponding to every rating by a user for an item, timestamp is also provided in the dataset. The timestamp is useful while computing novelty of an item at that point of time which will be used in proposed models.

The second dataset consists of movie reviews from Amazon [29]. The data spans a period of more than 10 years, including approximately 8 million reviews up to October 2012. Reviews include product and user information, ratings, timestamp, and a plaintext review. The total number of users is 889176 and total number of products is 253059. In order to use this dataset for experimentation purpose we have randomly sub-sampled the dataset to include 6466 users and 25350 products with only users, items, ratings and timestamp intact in the data. The total number of ratings available in the sub sampled dataset is 54996 which make the data sparser than ml-100 k dataset (99.67% sparsity).

To distinguish relevant (positive) and non-relevant (negative) items we will set a rating of *5* as relevant items. The categorization of relevant and non-relevant items will be used in calculating accuracy measures like precision. This will also be helpful in calculating number of relevant long-tail items recommended by a RS.

4.1 Evaluation Measures

This section gives a brief introduction to various measures to evaluate the proposed models and compare them with standard existing algorithms. The measures to evaluate accuracy, long-tail and diversity are briefly described in following sub-sections.

4.1.1 Accuracy Measures

In order to evaluate accuracy, the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are popular metrics in recommender systems. Since, RMSE gives more weightage to larger values of errors while MAE gives equal weightage to all values of errors, RMSE is preferred over MAE while evaluating the performance of RS. RMSE is popular metrics in RS until very recently and many previous works have based their findings on this metrics, therefore this metrics has been used primarily to exhibit the performance of the proposed models and RSVD model on various datasets. For a test user item matrix Γ the predicted rating $\hat{r}_{ui}$ for user-item pairs (u, i) for which true item rating r_{ui} are known, the RMSE is given by

$$\text{RMSE} = \sqrt{\frac{1}{|\Gamma|}\ \sum_{(u,i\in\Gamma)}(\hat{r}_{ui} - r_{ui})^2} \tag{15}$$

MAE on the other hand is given by

$$\text{MAE} = \frac{1}{|\Gamma|}\ \sum_{(u,i\in\Gamma)}|\hat{r}_{ui} - r_{ui}| \tag{16}$$

However, when the task is to find good items, the MAE and RMSE metric might not be appropriate. Therefore, a different accuracy metric that considers the frequency with which a RS makes correct or incorrect decisions (classifications) about whether an item is good or not has been in practice [11]. The top-k good items are presented to the user and based on the acceptance or rejection of items by the user accuracy is measured. Precision is a measure to evaluate the accuracy in such circumstances and is defined as the ratio of relevant items presented, N_{rs}, to the total number of items presented, N_s.

$$\text{Precision} = \frac{N_{rs}}{N_s} \tag{17}$$

4.1.2 Evaluation Measures of Long-Tail

The main notion of this paper revolves around the items in long-tail. Since, the purpose of the paper is to optimize TLT items with respect to every user and to promote TLT items to idiosyncratic users; we will quantify our results by directly measuring the overall frequency and coverage of long-tail items presented to all users by a RS. For categorizing long-tail items and popular items in various datasets we will follow a simple rule. We will first calculate the popularity of all items in dataset by finding the number of users who have rated the items as relevant. Then a cut-off value for categorization of long-tail and popular item will be determined based on top 10% of items arranged in decreasing order of popularity. It means that items in dataset will be arranged in decreasing order of popularity and cut-off value of popularity is the one which covers top 10% of the items in popularity rank. Using this cutoff value, long-tail items can be identified which will be used to evaluate performance of recommender systems with respect to long-tail. Frequency of long-tail (η) is defined as the performance measures to evaluate the prevalence of *long-tail* items in recommended list.

Frequency (η) of long-tail items is defined as summation of the frequencies of all the long-tail items appearing in the recommended lists by the RS for all users.

$$\eta = \sum_{(u,i\in\Gamma)} \#\text{ of long-tail items in recommended list of } i^{th} \text{ user} \tag{18}$$

As noted in [10], [42], diversity is also positively stimulated by promotion of long-tail items, and the same has been used as performance measure of RS in the present work. The measure of diversity as used in recommender systems has been derived from measurement of species diversity in ecology. Shannon entropy has been used as a measure of species diversity in ecology [34]. Similarly, in case of recommender systems Shannon entropy is a measure of item coverage and diversity of a RS model. It measures what percentages of items are recommended and how evenly they are distributed in the recommendations lists of users [20]. The following formula will be used to calculate the Shannon entropy based diversity.

$$\text{Diversity} = -\sum_{i=1}^{n} I_i \log_2 I_i \tag{19}$$

l_i represents the percentage of the recommendation lists that contains item i and n is the number of items in recommended list.

4.2 Cross Validation

Cross validation is a well-established technique in machine learning algorithms used in evaluation purposes. This technique ensures that the evaluation results are unbiased estimates and are not due to chance. The dataset is split into disjoint k-folds; (k-1) folds are used as training set while the left out set is used for testing. The procedure is repeated k times so that each time a unique test set can be used for performance evaluation. The measures such as RMSE, MAE, precision, long-tail and diversity used for evaluation of RS models will be calculated k times and then averaged to get the resultant unbiased estimate of the performance measures [8].

4.3 Results

In this section, we will now reveal the value of parameters used for each dataset in experimental setup and report the corresponding observations.

For cross validation, we have used 5-fold cross validation in both the datasets. The number of latent factors (*l*) used in RSVD, proposed model-1 and proposed model-2 are kept constant at the value for which RSVD performs best on cross-validated RMSE accuracy measure for respective datasets in order to compare with the proposed models on long-tail measures. Since normalized item novelty ($novelty(i_t^N)$) varies between *0* and *1*, where close to *0* indicates that the items are very popular at that instance and close to *1* indicates a novel item, we have varied the threshold value (θ_N) of item novelty from *0.1* to *0.9* in steps of *0.1* in the proposed model-2 to capture the variation in number of long-tail items with the varying threshold values of normalized novelty scores. We have also varied training factor $'\alpha'$ that adjusts the training of latent factors to promote long tail items in proposed model-2 between *1* and *2* in steps of *0.2.*

For evaluating precision, relevant (positive) items are those whose rating equals to *5*. In order to evaluate frequency (Ƞ) of long tail items present in the recommended list, we will categorize the long-tail items in each dataset according to the stated rule in previous the section. In ml-100k dataset the value of top 10% popular items are covered above the popularity value 36, so by predefined 10% rule, 36 is the cut-off value for classifying tail (non-hits) and head (hits) items. Similarly, in Amazon dataset the cut-off value is found to be 2. A recommendation list of top-k (top 5 and top 10) items has been generated for each user based on RSVD, PM-1 and PM-2 models. Precision, frequency (Ƞ) of long tail items in recommended lists and diversity of recommended lists are then calculated with top-k recommendations using various models for both datasets.

4.3.1 Comparison with Models on Long-Tail

We will present a comparative summary in table 4 and table 5 between Regularized SVD and PM-1 which illustrates the relative values of precision, long-tail (Ƞ) and diversity for both datasets. The number of latent factors (*l*) in case of RSVD and PM-1 for ml-100k dataset is taken as 5 since cross validated RMSE calculated is least for RSVD at *l* =5. The cross validated RMSE calculated for Amazon dataset is least at *l* =20, therefore number of latent factor (*l*) for PM-1 is also chosen as 20.

Table 4: Performance measures for ml 100k dataset

Model	MAE	RMSE	Precision for k=5	Precision for k=10	Ƞ for k=5	Ƞ for k=10	diversity for k=5	Diversity for k=10
RSVD	0.733	0.936	0.391	0.343	983.00	2283.00	4481.323	6125.70
PM-1	0.735	0.938	0.384	0.335	1023.200	2318.800	4564.628	6238.079

Table 4 and table 5, depicts performance measures on ml-100k dataset and Amazon dataset respectively. It is to be noted that the traditional RSVD generates less number of long-tail items in recommendation list with slightly better precision value than PM-1. PM-1 demonstrates that with the proposed scheme, long-tail items can be recommended to idiosyncratic users according to the taste of TLT items. As one can note that the precision values on both datasets have not deteriorated but measure of long-tail when experimented with PM-1 is greater than that of RSVD. This implies that PM-1 has indeed taken into account the taste of TLT items for each user and accordingly recommended items to each user. The proposed model, PM-1, promotes long-tail items at the cost of marginally less precision which is in synchronous with previous literatures. Long-tail and diversity are also positively related and it is verified in the above experiment.

Table 5: Performance measures for Amazon dataset

Model	MAE	RMSE	Precision for k=5	Precision for k=10	Ո for k=5	Ո for k=10	diversity for k=5	Diversity for k=10
RSVD	0.686	1.057	0.590	0.578	3745.000	4579.600	66715.287	78369.296
PM-1	0.688	1.058	0.589	0.578	3947.000	4679.400	68768.097	78861.495

4.3.2 Promoting Long-Tail Items

The second part of the experiment is carried out using proposed model PM-2. The goal is to promote more of long-tail items to idiosyncratic users. The experimentation is carried on both datasets. For each dataset, relation between training factor (α) and frequency of long-tail item is plotted by varying the values of training factor (α). The impact of this variation of training factor (α) on precision and diversity vis-à-vis frequency of long-tail items of the recommendation list is also plotted.

The training factor (α) is varied between 1 and 2 in steps of 0.2 viz., 1.2,1. 4…2, which apparently provides thrust to the items that are relevant and are in long-tail. This in turn adjusts the latent features of user and items accordingly and thus promotes more of long-tail items. This step indisputably deteriorates the precision value of the model, but it can be used as a trade-off parameter for promoting long-tail products at the expense of precision. The variation of diversity with promotion of long-tail items has also been plotted to understand the impact of promoting long-tail on recommendation list. The other parameter is item novelty values (θ_N), that varies from 0.1 to 0.9, and determines novel and relevant items from a list of all the items.

For ml-100k dataset (Figure 1), at various item novelty threshold values (θ_N) which varies from 0.1 to 0.9 for ml-100k dataset, long-tail (Ո) increases with increasing training factor (α). The graph is drawn for top-5 and top-10 recommendation list and it is clearly seen that more of TLT items are recommended invariably with increasing training factor (α). However, increasing threshold value (θ_N) does not necessarily result in more of TLT items. Maximum number of TLT items is promoted at $\theta_N = 0.5$, while least number of TLT items is promoted when $\theta_N = 0.1$ in both the recommendation lists. At $\theta_N = 0.1$ almost all items (hits and non-hits) in the dataset are equally boosted while at $\theta_N = 0.5$ almost all non-hits items in the dataset are boosted resulting in maximum Ո. As θ_N is increased from 0.5 to 0.9 number of non-hits items in the dataset decreases resulting in proportional decline in Ո. So based on the above result we can say that $\theta_N = 0.5$ is the best threshold value for partitioning of hits and non-hits items in ml-100k dataset.

In the case of Amazon dataset (Figure 2), we find that with rise of both training factor (α) and item novelty threshold values (θ_N) there is an increasing trend in Ո. The increasing trend of Ո with increasing α is model driven as is the case in previous dataset, however the increase in θ_N also results in increase in Ո invariably in both the recommendation lists. The reason of this increase is attributed to presence of TLT items heavily in the tail. At $\theta_N = 0.1$ almost all items (hits and non-hits) in the dataset are equally boosted while at $\theta_N = 0.5$ there are more hits items than in non-hits items and at $\theta_N = 0.9$ almost all non-hits items in the dataset are boosted resulting in maximum Ո. If the strategy of an e-commerce firm is only to promote long-tail items to idiosyncratic users without other constraints the above graphs solely can help in deciding the parameters of the model to implement in RS. However, solely promoting long-tail items may lead to decline in precision which may undesirably affect the trust of a customer on RS and may result in trust deficit in the e-commerce firm. Hence, other measure like precision has to be looked upon before deciding the parameters of the model. Therefore, we also present precision-long-tail relationship and long-tail-diversity relationship for both the datasets which may give a better insight to e-commerce firms while deciding the parameters of the models based on the strategy of firms.

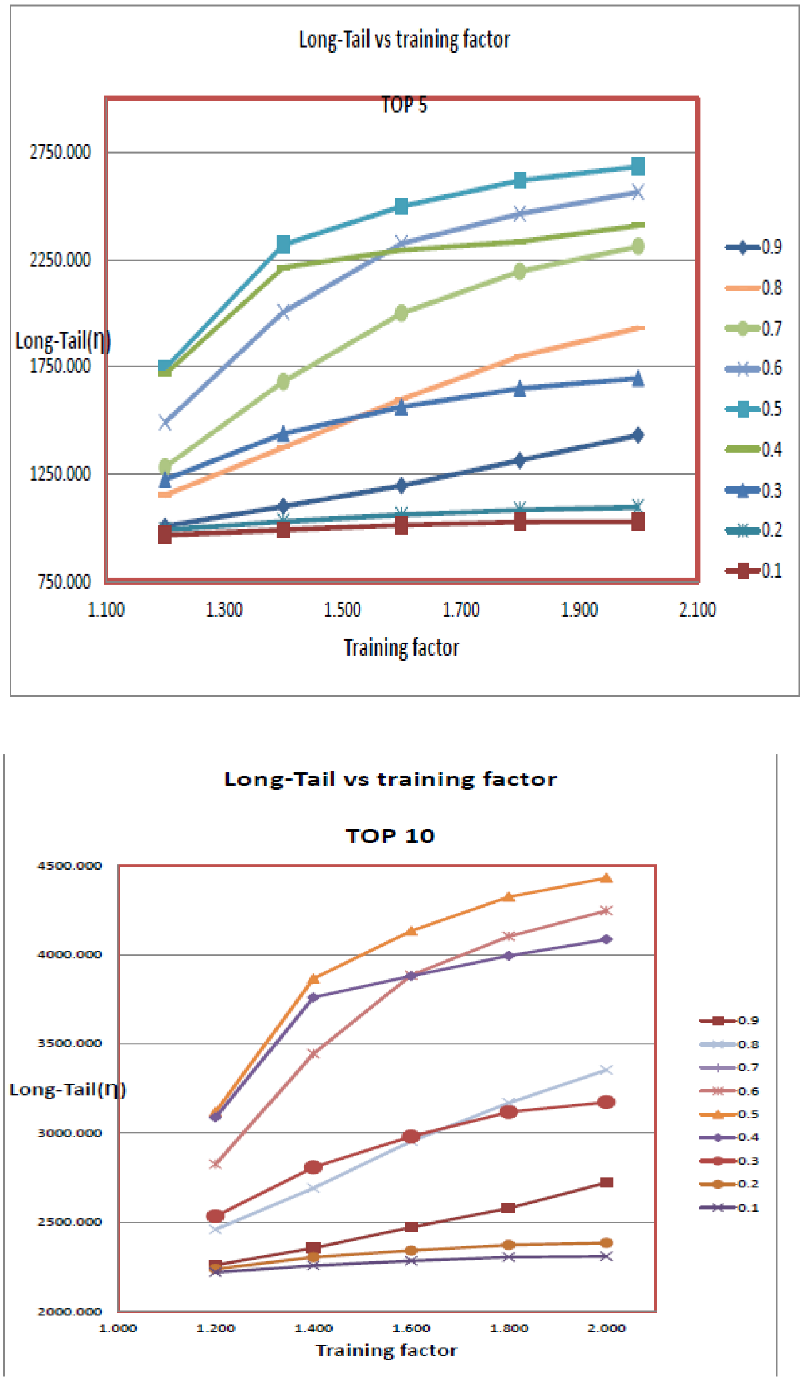

Figure 1: Relation between training factor (α) and long-tail (Ƞ) for ml-100k dataset

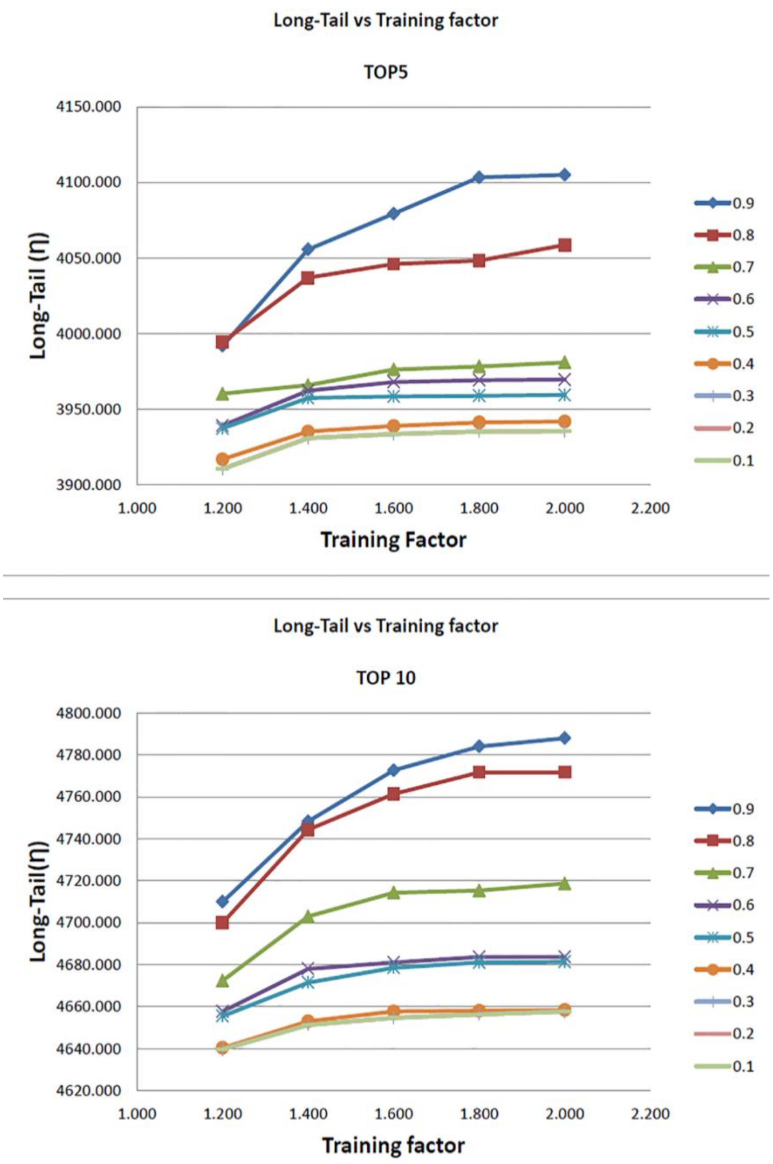

Figure 2: Relation between training factor (α) and long-tail (Ƞ) for Amazon dataset

There is a trade-off between precision and long-tail as shown in Figure 3 in ml-100k dataset. It is observed from the figure that, for top-5 recommendation list the highest value of precision occurs at $\theta_N = 0.2, \alpha = 1.4$ however Ƞ is very low. Similarly, in case of Top-10 recommendation the maximum value of precision is at $\theta_N = 0.2, \alpha = 1.6$ but Ƞ is very low. In case of both the top-5 recommendation and top-10 recommendation highest value of Ƞ occurs at $\theta_N = 0.5$, $\alpha = 2$ with low precision value which may be desirable for a firm which gives more weight to promotion of long-tail than precision. In case of top-5 recommendation list if an e-commerce firm puts almost equal weightage to both long-tail and precision then $\theta_N = 0.4$, $\alpha = 2$ could be more sensible choice for model parameters. Similarly, for top-10 recommendation list $\theta_N = 0.5, \alpha = 1.4$ could be practical choice considering the equal weightage provided to both precision and promotion of long-tail ítems.

Similar to ml-100k dataset, Amazon dataset also depicts an inverse relationship between precision and long-tail (Ƞ) at fixed item novelty threshold values (θ_N). It is important to note from figure 3 and figure 4, that a lower level of precision does not automatically results in more number of long-tail items but it also depend on another parameter θ_N. Therefore, it is worth considering all the parameters viz. θ_N and α for choosing a right model for achieving a desired objective. The analysis done for ml-100k dataset for choosing parameters of a model depending on varying objective holds true in case of Amazon dataset as well. For top-5 recommendation list the highest value of precision occurs at $\theta_N = 0.5, \alpha = 1.2$ however Ƞ is comparatively lower than $\theta_N = 0.5$. Similarly, in case of top-10 recommendation the maximum value of precision is at $\theta_N = 0.5$ but Ƞ is lower. In case of both the top-5 recommendation and top-10 recommendation highest value of Ƞ occurs at $\theta_N = 0.9$, $\alpha = 2$ with lower precision value. However, the value of precision varies between 0.572 and 0.588 for top-5 recommendation and between 0.569 and 0.581 for top-10 recommendation list. The range of variation is very low which suggests that precision is nearly constant for either of the parameters in the model and therefore, one can only look at Ⴖ while deciding the parameters of the model.

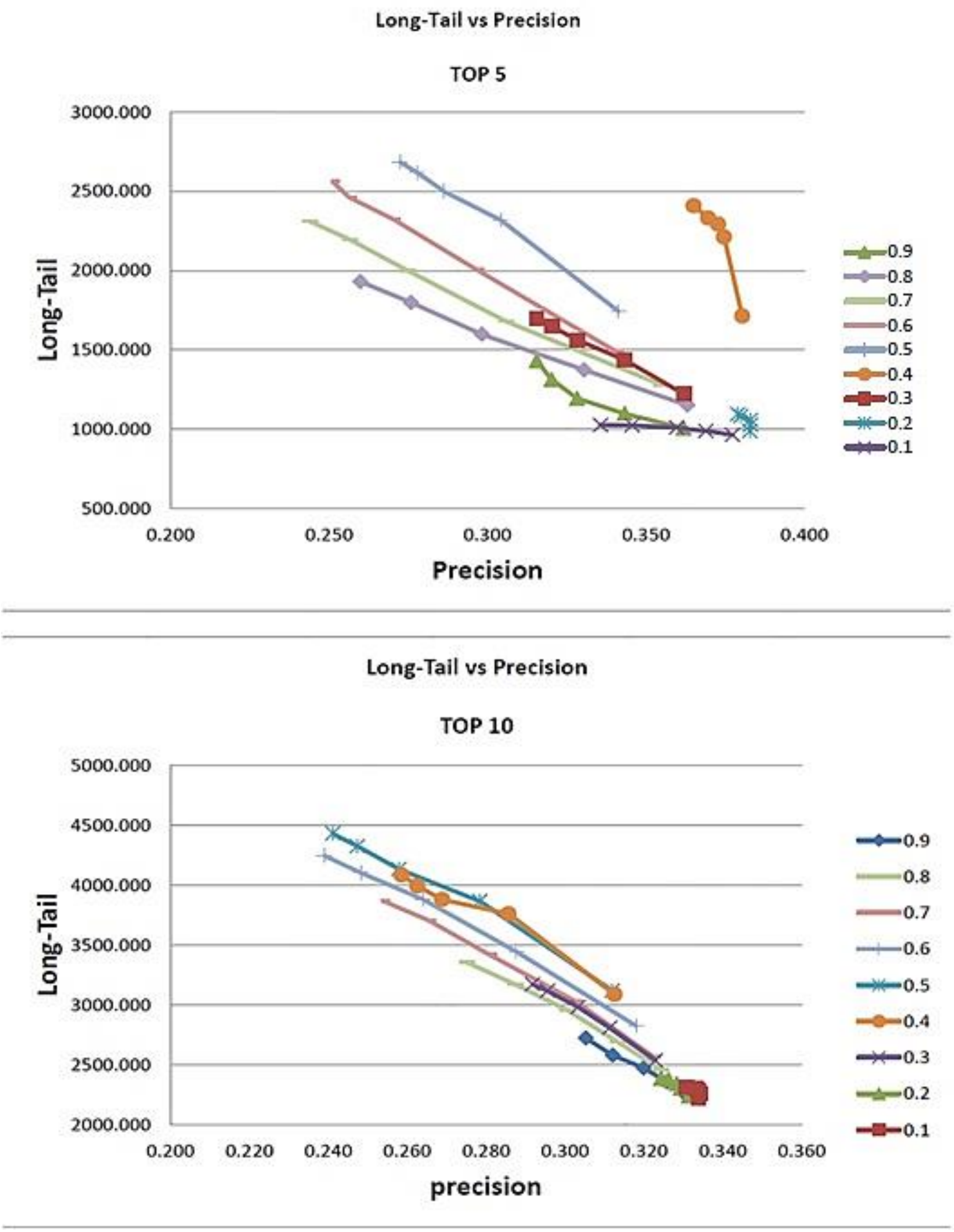

Figure 3: Relation between precision and long-tail (Ⴖ) for ml-100k dataset

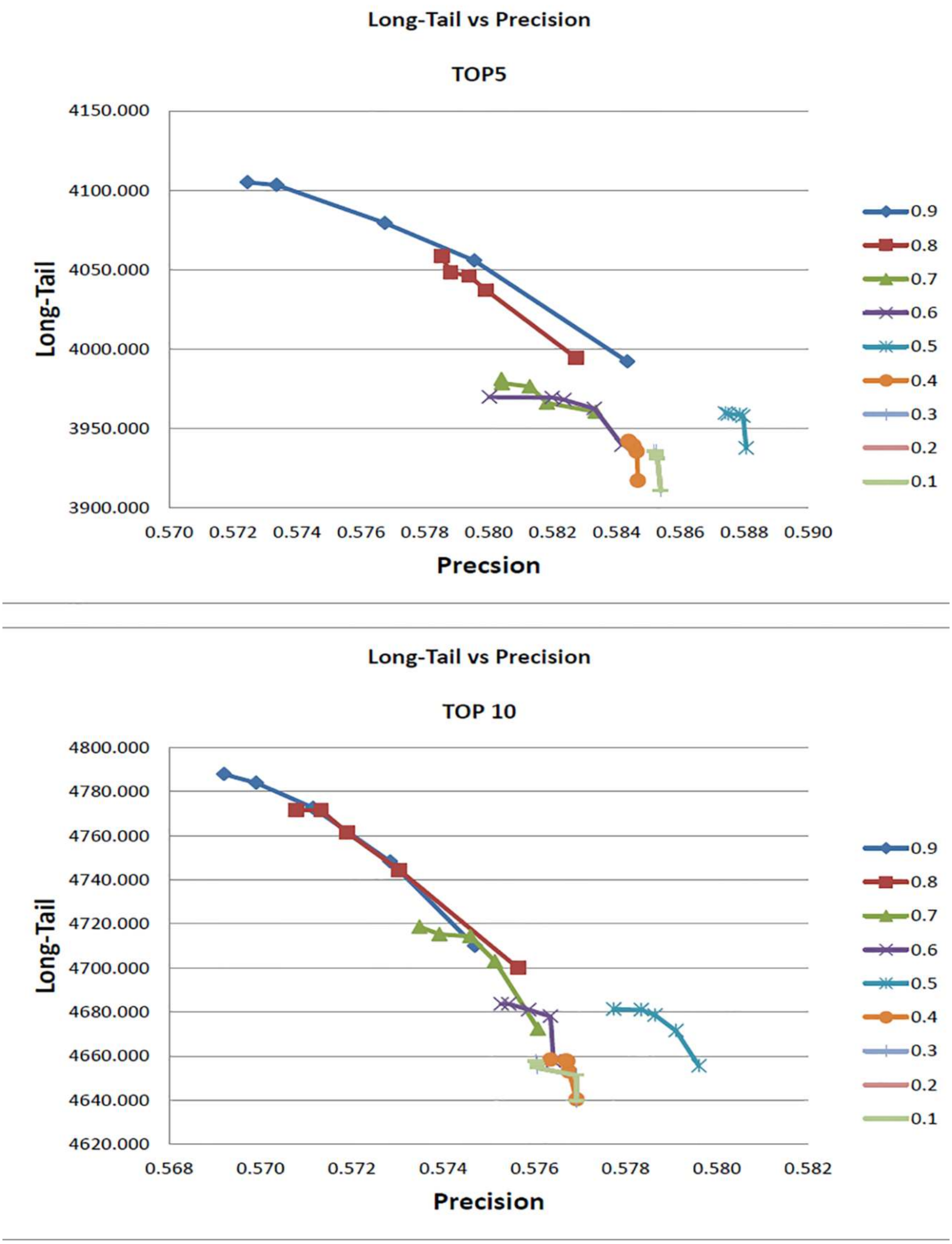

Figure 4: Relation between precision and long-tail (Ƞ) for Amazon dataset

It is also worth observing the relationship between long-tail and diversity of the model for both datasets at various levels of θ_N and α. It is observed from figure 5 and figure 6 that diversity of recommendation list is positively stimulated by long-tail for a given θ_N. Also, diversity can be another measure to look into before deciding the parameters of the proposed model. One can look at pairwise or overall trade-off between long-tail, diversity and precision before deciding the parameter of the model given the preference of the measures.

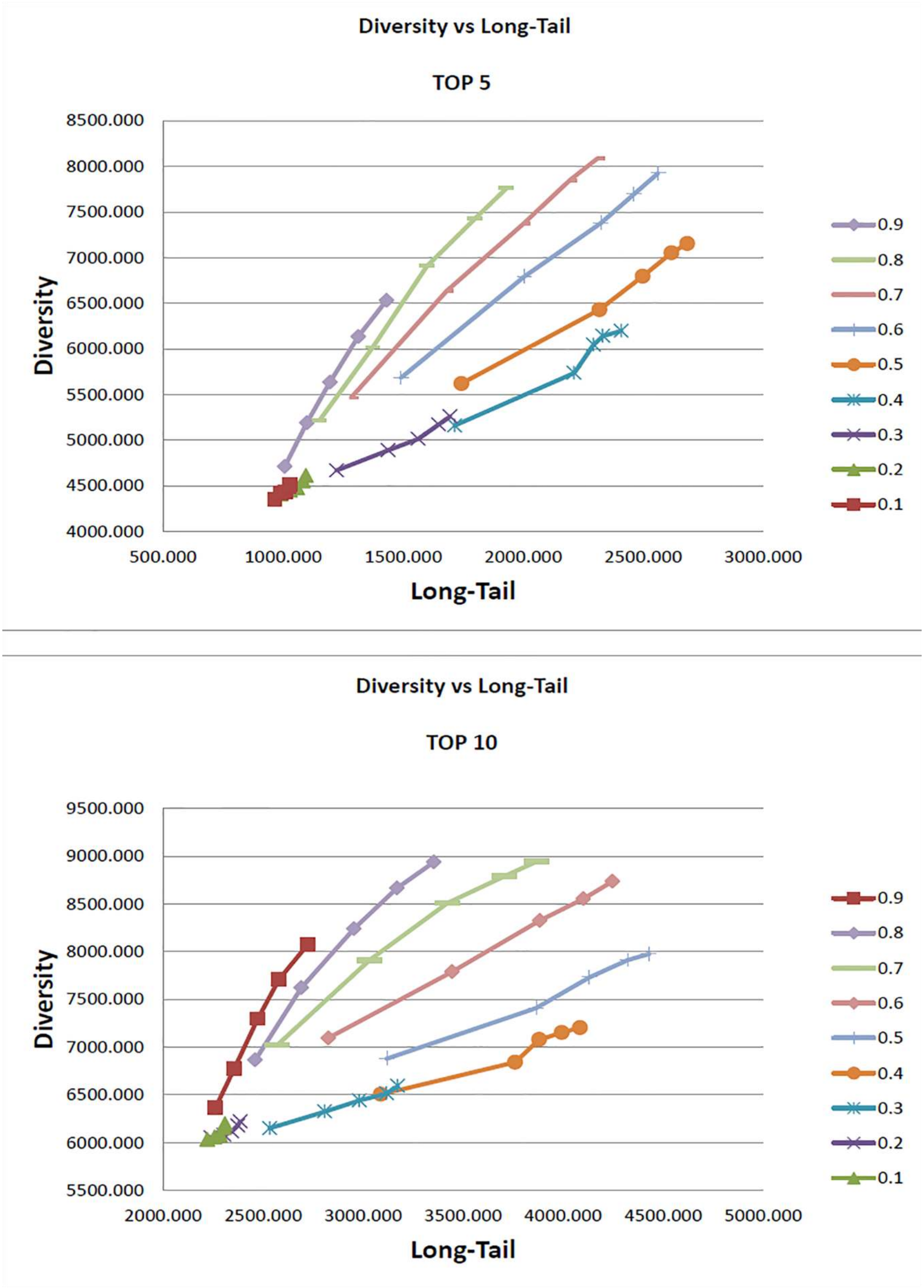

Figure 5: Relation between diversity and long-tail (Ƞ) for ml-100k dataset

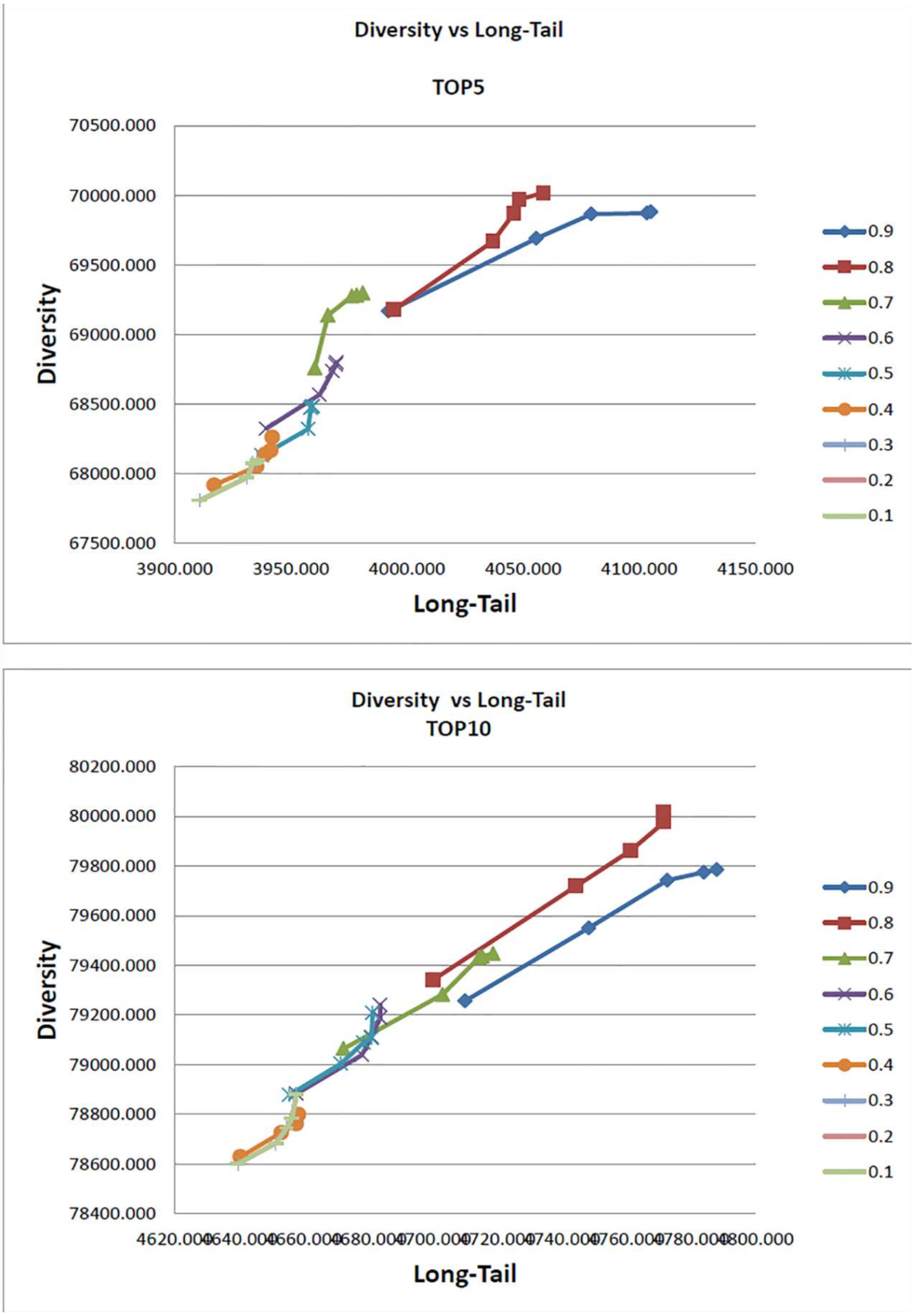

Figure 6: Relation between diversity and long-tail (∏) for Amazon dataset

Depending on the objective and strategy of an e-commerce firm on precise, diverse and long-tail item the parameters of the proposed model can be suitably be modified. In case of ml-100k dataset if the preference is for recommending more of diverse items only, then considering top-5 recommendation list $\theta_N = 0.7, \alpha = 2$ will be chosen and considering top-10 recommendation list $\theta_N = 0.8, \alpha = 2$ will be chosen as parameter. Similarly, in case of Amazon dataset the suitable parameters for recommending only diverse item considering top-5 and top-10 recommendation list can be $\theta_N = 0.8, \alpha = 2$ for the both scenarios.

If the objective is to promote diverse as well as long-tail items giving equal preference to both the performance measures $\theta_N = 0.6, \alpha = 2$ could be suitable choice inferred from figure 5 in case of top-5 recommendation list and $\theta_N = 0.7, \alpha = 2$ in case of top-10 recommendation list for ml-100 dataset. Likewise, in case of Amazon dataset with the above objective $\theta_N = 0.9$, $\alpha = 2$ and $\theta_N = 0.9$, $\alpha = 2$ can be a parameter chosen for top-5 and top-10 recommendation list.

5 Discussion and Conclusion

The experimentation of the proposed models (PM-1 and PM-2) on the publicly available benchmark datasets demonstrates their suitability i) to provide personalized recommendations to a user by taking into account the respective taste for long-tail items, and ii) to promote long-tail items to idiosyncratic users.PM-1 is an additive model of matrix factorization and novelty measure which ensures that user's taste towards long-tail item is captured during personalized recommendation. Matrix factorization method has been well known in recommendation systems, which ensures that a user receives accurate recommendations. In the additive model, PM-1 utilizes the characteristics of matrix factorization and supplements it with a parameter that measures taste of each user for long-tail items. The experiment shows that the accuracy of the model does not change significantly from matrix factorization model, but the measure of long-tail and diversity increases on both the datasets. This has been achieved by the additive model used in PM-1 which trains itself to learn the parameters of users' taste for long-tail items and recommending the long-tail yet relevant items.

The second objective, promoting long-tail items to idiosyncratic users, has been fulfilled by PM-2. A training factor (α), and novelty threshold (θ_N) are the additional parameters in the model PM-2 over PM-1. The additional parameters ensure that the latent features of users and items; and parameter accounting the user's taste for long-tail items are learnt. Novelty threshold (θ_N) , measure of long-tail, is fixed to a value that distinguishes between long-tail and hit items, the ratings of long-tail items by a user are then boosted by a training factor (α) in order to learn the parameters of PM-2. The parameters so learnt recommend more of long-tail items and this claim has been demonstrated in experimentation by varying training factor (α) for different novelty threshold (θ_N).

Since matrix factorization is formulated on co-occurrence patterns between users and items, it may be possible that the latent features learnt may not be meaningful for *fewer rated items* or *users who have rated few items*. In the extreme cases, there is a possibility of entirely new user or novel items which may not be dealt with matrix factorization scheme. The proposed model also suffers from these weaknesses; however, content based filtering can be used initially for such cases as demonstrated in previous works [5], [9], [25], [27], [28], [32].

The proposed model, PM-1 explores the possibility it offers to a variety of strategic business advantage for online business platforms. Proposed model, PM-1, guides the users to choose from a recommendation list based on the implicit liking/disliking of long-tail items to users. Since this will help users navigate to niche items of their choice thereby reducing the search time for suitable items, it increases their trustworthiness towards such recommender systems. This may in turn generate a *recommendation-buying* spiral cycle which implies that relying on recommender systems, users will tend to buy from recommendation list that will effectively improve with more information available about the users' behaviors [44]. Proposed model, PM-2, on the other hand can prove to be expedient tool to promote long-tail, diverse and precise items with flexibility, based on past purchase behavior of users. Unexpected yet useful recommendation will indeed help e-commerce sellers to utilize their unlimited shelf space which is a major advantage over brick and mortar shops.

References

[1] P. Adamopoulos and A. Tuzhilin, On unexpectedness in recommender systems: Or how to better expect the unexpected, ACM Transactions on Intelligent Systems and Technology (TIST), vol. 5, no. 4, pp. 1-50, 2015.

[2] G. Adomavicius and Y. Kwon, Diversity using ranking-based techniques, IEEE Transactions on Knowledge and Data Engineering, vol. 24, no. 5, pp. 896-911, 2012.

[3] G. Adomavicius and Y.O. Kwon, Maximizing aggregate recommendation diversity: A graph-theoretic approach, in Proceedings of the Workshop on Novelty and Diversity in Recommender Systems (DiveRS 2011), Chicago, Illinois, USA, 2011, pp. 3-10.

[4] D. Agarwal and B.-C. Chen, Regression-based latent factor models, in Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, Paris, 2009, pp. 19-28.

[5] H.J. Ahn, A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem, Information Sciences, vol. 178, no. 1, pp. 37-51. 2008.

[6] C. Anderson, The Long Tail: How Endless Choice Is Creating Unlimited Demand, London: Random House Business Books ,2006

[7] R.A. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[8] Bellogin, P. Castells and I. Cantador, Precision-oriented evaluation of recommender systems: An algorithmic comparison, in: Proceedings of the Fifth ACM Conference on Recommender Systems, ACM, New York, NY, USA, 2011, pp. 333-336.

[9] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, A collaborative filtering approach to mitigate the new user cold start problem, Knowledge-Based Systems, vol. 26, pp. 225-238, 2012.

[10] P. Castells, S. Vargas and J. Wang, Novelty and diversity metrics for recommender systems: Choice, discovery and relevance, in Proceedings of International Workshop on Diversity in Document Retrieval (DDR), Dublin, Ireland, 2011, pp. 29-37.

[11] N. Cerpa, M. Bardeen, C.A. Astudillo and J. Verner, Evaluating different families of prediction methods for estimating software project outcomes, Journal of Systems and Software, vol. 112, no. 2016, pp. 48-64, 2016.

[12] L. Chen, W. Wu and L. He, How personality influences users' needs for recommendation diversity?, in Proceedings CHI '13 Extended Abstracts on Human Factors in Computing Systems on - CHI EA '13, Paris,France, 2013, p. 829.

[13] A. Elberse and F. Oberholzer-Gee, Superstars and underdogs: An examination of the long tail phenomenon in video sales, Harvard Business School Working Paper Series, No. 07-015, 2006.

[14] Y. Feng, H. Li and Z. Chen, Improving recommendation accuracy and diversity via multiple social factors and social sircles, International Journal of Web Services Research, vol. 11, no. 4, pp. 32-46, 2014.

[15] M. Ge, C. Delgado-Battenfeld and D. Jannach, Beyond accuracy: Evaluating recommender systems by coverage and serendipity, Proceedings of the Fourth ACM Conference on Recommender Systems on Recommender Systems, ACM,New York, USA, 2010, pp. 257-260.

[16] D. Geiger and M. Schader, Personalized task recommendation in crowdsourcing information systems - current state of the art, Decision Support Systems, vol. 65, pp. 3-16, 2014.

[17] S. Goel, A. Broder, E. Gabrilovich, and B. Pang, Anatomy of the long tail : Ordinary people with extraordinary tastes, in Proceedings of the Third ACM International Conference on Web Search and Data Mining, ACM, New York,USA, 2010, pp. 201-210.

[18] B. Gu, Q. Tang and A.B. Whinston, The influence of online word-of-mouth on long tail formation, Decision Support Systems, vol. 56, pp. 474-481, 2013.

[19] K. Hosanagar, D. Fleder, D. Lee, and A. Buja, Will the global village fracture into tribes? Recommender systems and their effects on consumer fragmentation, Management Science, vol. 60, no. 4, pp. 805-823, 2014.

[20] M. Izadi, A. Javari and M. Jalilii, Unifying inconsistent evaluation metrics in recommender systems, in Proceedings RecSys Conference, REDD Workshop, Silicon Valley, USA, 2014, pp. 1-7.

[21] J. a Konstan, S.M. Mcnee, C. Ziegler, R. Torres, N. Kapoor, and J.T. Riedl, Lessons on applying automated recommender systems to information-seeking tasks, Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2, AAAI, Boston, USA, 2006, pp. 1630-1633.

[22] Y. Koren, Factorization meets the neighborhood: A multifaceted collaborative filtering model, in Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, Las Vegas, Nevada, USA, 2008, pp. 426-434.

[23] Y. Koren, The bellkor solution to the netflix grand prize, Netflix Prize Documentation, vol.81, pp. 1-10, 2009.

[24] Y. Koren, R. Bell, Advances in Collaborative Filtering, in Recommender Systems Handbook (F. Ricci, L. Rokach and B. Shapira, Eds.). New York: Springer, 2011, pp. 145-186.

[25] X. T. Lam, T. Vu, T. D. Le, and A. D. Duong, Addressing cold-start problem in recommendation systems, in Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication, Suwon, Korea, 2008, pp. 208-211.

[26] N. Lathia, S. Hailes, L. Capra, and X. Amatriain, Temporal diversity in recommender systems, in Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '10), Geneva, Switzerland, 2010, pp. 210-217.

[27] C.W. Leung, S.C. Chan and F. Chung, An empirical study of a cross-level association rule mining approach to cold-start recommendations, Knowledge-Based Systems, vol. 21, no. 7, pp. 515-529, 2008.

[28] B. Lika, K. Kolomvatsos and S. Hadjiefthymiades, Facing the cold start problem in recommender systems, Expert Systems with Applications, vol. 41, pp. 2065-2073, 2014.

[29] J. McAuley and J. Leskovec, Hidden factors and hidden topics: understanding rating dimensions with review text, in Proceedings of the 7th ACM Conference on Recommender Systems - RecSys '13, Hong Kong, China, 2013, pp. 165-172.

[30] S.M. McNee, J. Riedl and J. a Konstan, Being accurate is not enough: how accuracy metrics have hurt recommender systems, in Proceedings CHI'06 Extended Abstracts on Human Factors in Computing Systems, Montreal, Canada, 2006, pp. 1097-1101.

[31] R. Mishra, P. Kumar and B. Bhasker, A web recommendation system considering sequential information, Decision Support Systems, vol. 75, pp. 1-10, 2015.

[32] T. Pahikkala, M. Stock, A. Airola, T. Aittokallio, B. De Baets, and W. Waegeman, A two-step learning approach for solving full and almost full cold start problems in dyadic prediction, in Proceedings Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Nancy, France 2014, pp. 517-532.

[33] Y.-J. Park and A. Tuzhilin, The long tail of recommender systems and how to leverage it, in Proceedings of the 2008 ACM Conference on Recommender Systems RecSys 08, Lausanne, Switzerland, 2008, p. 11.

[34] E. Pielou, Shannon's formula as a measure of specific diversity: Its use and misuse, The American Naturalist, vol. 100, no. 914, pp. 463-465, 1966.

[35] É. Polytechnique, F. De Lausanne and K. Aberer, Towards a dynamic top-n recommendation framework, in Proceedings of the 8th ACM Conference on Recommender Systems - RecSys '14, California, USA, 2014, pp. 217-224.

[36] M.T. Ribeiro, A. Lacerda, A. Veloso, and N. Ziviani, Pareto-efficient hybridization for multi-objective recommender systems, in Proceedings of the Sixth ACM Conference on Recommender Systems, ACM, Dublin, Ireland, 2012, pp. 19-26.

[37] R. Salakhutdinov and A. Mnih, Bayesian probabilistic matrix factorization using Markov chain Monte Carlo, in Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 2008, pp. 880-887.

[38] G. Shani and A. Gunawardana, Evaluating recommendation systems, in Recommender Systems Handbook (F. Ricci, L. Rokach and B. Shapira, Eds.). New York: Springer, 2011, pp. 257-298.

[39] H. Steck, B. Labs and M. Hill, Item Popularity and Recommendation Accuracy, in Proceedings of the 5th ACM Conference on Recommender Systems - RecSys ’11, Chicago, 2011, pp. 125-132.

[40] A. Umyarov and A. Tuzhilin, improving rating estimation in recommender systems using aggregation- and variance-based hierarchical models, in Proceedings of the Third ACM Conference on Recommender Systems - RecSys ’09, New York, USA, 2009, p. 37.

[41] S. Vargas and P. Castells, Rank and relevance in novelty and diversity metrics for recommender systems, in Proceedings of the Fifth ACM Conference on Recommender Systems, ACM, Chicago, USA, 2011, pp. 109-116.

[42] S. Vargas and P. Castells, Improving sales diversity by recommending users to items, in Proceedings of the 8th ACM Conference on Recommender Systems - RecSys ’14, Foster City, Silicon Valley, USA, 2014, pp. 145-152.

[43] X. Yang, Y. Guo, Y. Liu, and H. Steck, A survey of collaborative filtering based social recommender systems, Computer Communications, vol. 41, pp. 1-10, 2014.

[44] V.Y. Yoon, R.E. Hostler, Z. Guo, and T. Guimaraes, Assessing the moderating effect of consumer product knowledge and online shopping experience on using recommendation agents for customer loyalty, Decision Support Systems, vol. 55, pp. 883-893, 2013.

[45] M. Zhang, Enhancing diversity in Top-N recommendation, in Proceedings of the Third ACM Conference on Recommender Systems - RecSys ’09, New York, USA, 2009, p. 397.

[46] M. Zhang and N. Hurley, Avoiding monotony: improving the diversity of recommendation lists, in Proceedings of the 2008 ACM Conference on Recommender Systems, ACM, Lausanne, Switzerland, 2008, pp. 123-130.

[47] M. Zhang and N. Hurley, Statistical modeling of diversity in top-n recommender systems, in Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 01, IEEE Computer Society, Washington DC, USA, 2009, pp. 490-497.

[48] M. Zhang and N. Hurley, Novel item recommendation by user profile partitioning, in Proceedings - 2009 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2009, Washington DC, USA, 2009, pp. 508-515.

[49] Y.C. Zhang, D.Ó. Séaghdha, D. Quercia, and T. Jambor, Auralist: introducing serendipity into music recommendation, in Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, Seattle, USA, 2012, pp. 13-22.

[50] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J.R. Wakeling, and Y.-C. Zhang, Solving the apparent diversity-accuracy dilemma of recommender systems, Proceedings of the National Academy of Sciences of the United States of America, vol 107, no. 10, pp. 4511-4515,2010.

[51] T. Zhou, R.Q. Su, R.R. Liu, L.L. Jiang, B.H. Wang, Y.C. Zhang, Accurate and diverse recommendations via eliminating redundant correlations, New Journal of Physics, vol. 11, no. 12, pp. 1-19, 2009.

[52] C. N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, Improving recommendation lists through topic diversification, in Proceedings of the 14th International Conference on World Wide Web, ACM, Chiba, Japan, 2005, pp. 22-32.