# Incorporating Aggregate Diversity in Recommender Systems using Scalable Optimization Approaches

İbrahim Muter
School of Management, University of Bath, BA2 7AY, UK, i.muter@bath.ac.uk,

Tevfik Aytekin
Bahçeşehir University, Department of Computer Engineering, Beşiktaş, 34353 Istanbul, Turkey,
tevfik.aytekin@eng.bahcesehir.edu.tr,

The success of a recommender system is generally evaluated with respect to the accuracy of recommendations. However, recently diversity of recommendations has also become an important aspect in evaluating recommender systems. One dimension of diversity is called aggregate diversity which refers to the diversity of items in the recommendation lists of all users and can be defined with different metrics. The maximization of both accuracy and the aggregate diversity simultaneously renders a multi-objective optimization problem which can be handled by different approaches. In this paper, after providing a thorough analysis of the multi-objective optimization approaches for this problem, we propose a new model which takes into account both accuracy and aggregate diversity. Different from previous works our model is specifically designed to incorporate distributional diversity metrics, which measure how evenly the items are distributed in the recommendation lists of users. In order to solve the large-scale instances, we propose a column generation algorithm and a Lagrangian relaxation approach based on the decomposition of the model. We present the results of the mathematical models and the performance of the proposed methodology that are obtained by computational experiments on real world data sets. These results reveal that our model successfully captures the trade-off between the objectives and reaches very high levels of distributional diversity.

*Key words*: recommender systems; multi-objective optimization; integer programming; column generation.
*History*:

## 1. Introduction

Recommender systems help people to find items of interest in large product lists (Ricci et al. 2011, Adomavicius and Tuzhilin 2005). They are currently utilized in most of the e-commerce sites. The success of a recommender algorithm is generally evaluated with the accuracy of its recommendations, that is, how well the algorithm predicts whether a user will like an item or not. The accuracy of recommendations is no doubt an important aspect of a successful recommender system. However, recently researchers working on recommender systems have recognized that there are other aspects of recommender systems which are important for both user satisfaction and system performance. One such important aspect is the diversity of recommendations.

Diversity in recommender systems has two dimensions: individual and aggregate diversity. Individual diversity refers to the diversity of items in a recommendation list of each individual user. The aggregate diversity, on the other hand, refers to the diversity of items in the recommendation lists of all users. While the individual diversity of recommendation lists is important with respect to user satisfaction, the aggregate diversity is important from the business point of view. Being able to recommend most (if not all) of the items in the inventory of an e-store will increase sales compared to recommending only the popular items. Also in some domains, such as recommending movies, the unpopular items (that is, items in the long tail) tend to have lower license fees compared to the popular items (Goldstein and Goldstein 2006). Hence, being able to recommend the unpopular items has the potential for reducing costs. In this paper, our focus will be on improving aggregate diversity.

Different metrics can be used to define various aspects of aggregate diversity. One such metric used by Adomavicius and Kwon (2012) is as follows:

$$\textit{Diversity-in-top-N} = \left| \bigcup_{u \in U} L(u) \right| \tag{1}$$

where $U$ is the set of all users in the system, $L(u)$ is the recommendation list of size $N$ of user $u \in U$. According to this metric, the aggregate diversity is simply the total number of distinct items in the recommendation lists of all users. Adomavicius and Kwon (2012, 2014) developed methods for increasing the aggregate diversity based on this metric. However, this simple and easy-to-understand metric has an important flaw: it is possible to increase *diversity-in-top-N* without having a balanced distribution of items in the recommendation lists of the users. For example, in principle, it is possible to recommend each item in the system to a distinct user and fill in the rest of the recommendation lists of the users with the same set of items. In this case, *diversity-in-top-N* will have the maximum possible value, however, the distribution of the items in the user lists will be very imbalanced. In order to capture this aspect of aggregate diversity, Adomavicius and Kwon (2012) employed additional measures referred to as distributional diversity metrics. Adomavicius and Kwon (2014) employed one of these distributional diversity metrics, namely Gini-diversity, to evaluate the performance of their approach. In this paper, in addition to this metric, we also employ an entropy-based metric to measure the distributional diversity which is defined in Adomavicius and Kwon (2012). These metrics are given below:

$$\textit{Entropy-diversity} = -\sum_{i=1}^{n} \left( \frac{rec(i)}{total} \right) ln \left( \frac{rec(i)}{total} \right) \tag{2}$$

$$\textit{Gini-diversity} = 2 \sum_{i=1}^{n} \left[ \left( \frac{n+1-i}{n+1} \right) \times \left( \frac{rec(i)}{total} \right) \right] \tag{3}$$

In both of the definitions, $i$ refers to an item in the item set $I$, $rec(i)$ is the number of users who has been recommended item $i \in I$, $n$ is the number of items which are available for recommendation, and $total$ is the total number of recommendations across all users. In the calculation of Gini-diversity, the items are sorted in non-decreasing order of $rec(i)$, and in Entropy-diversity, we take $ln(0) = 0$ when $rec(i) = 0$ for some $i \in I$. These metrics measure how evenly the items are recommended to the users.

Adomavicius and Kwon (2012) experimentally show that distributional diversity metrics are correlated with *diversity-in-top-N*. However, as we show in Section 5, the capacity of the optimization method designed by Adomavicius and Kwon (2014) for increasing *diversity-in-top-N* is very limited in increasing distributional diversity metrics. Being able to increase distributional diversity (i.e., having a balanced distribution of the recommended items) as much as possible is a desirable capability in many situations. Take job recommendation as an example. Many e-recruitment platforms provide job recommendations as a service to their users (Wu et al. 2014) and users rely on these recommendations to find new jobs. Companies pay to these e-recruitment platforms for displaying their job ads. Given that companies pay an equal amount of money for their job ads it is important for the e-recruitment platform to give an equal share as much as possible to each ad in the recommendation lists. It is really undesirable if a significant portion of the job ads never appear (or appear in very small numbers) in the recommendation lists and a small portion of the job ads gets the lion's share. Hence, it is important to develop methods for increasing distributional diversity as much as possible without reducing accuracy dramatically.

Let us also give a visual illustration of the effect of an increase in these distributional diversity metrics. Figure 1 shows the distribution of $rec(i)$ for each item $i \in I$. This figure is generated using a standard recommender algorithm which considers only the accuracy of recommendations applied to the Movielens[1] 1M data set. The items are sorted in non-increasing order of $rec(i)$ in the horizontal axis. As it can be seen from Figure 1, a small portion of the items are recommended to many users while most of the items are either not recommended at all or recommended to only few users. This is a known effect of recommender algorithms (Fleder and Hosanagar 2009). One explanation of this is that unpopular products have limited ratings (or consumer interaction) and are difficult to recommend. Thus, it is clear that new methods need to be developed to support recommender systems in recommending unpopular items which can have positive effects on the system as explained previously. Methods which can increase the distributional diversity metrics given in (2) and (3) are useful in this regard. An increase in the distributional diversity metrics will correspond to an upward shift in the tail of this distribution and a downward shift in the head of the distribution (i.e., a change

[1] http://grouplens.org/datasets/movielens/

towards a more even distribution). In this paper, we will develop methods specifically designed to improve these distributional diversity metrics along with the accuracy of recommendations. Such problems with more than one objective are called multi-objective optimization problems, which are shortly reviewed in Section 3.

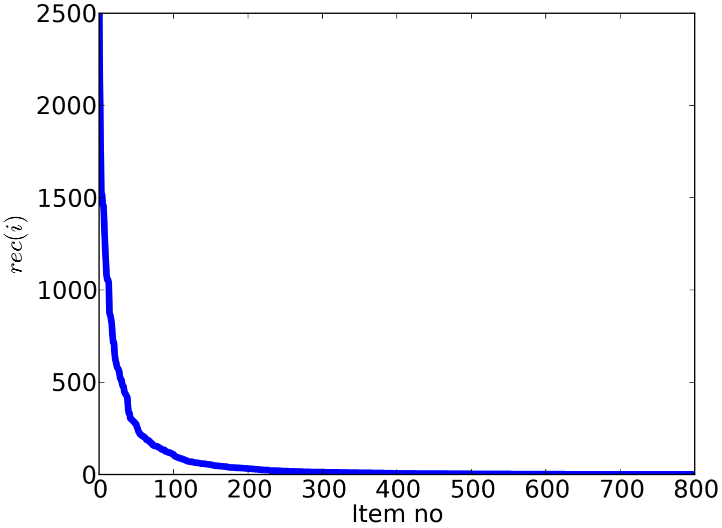

Figure 1: Distribution of $rec(i)$ in sorted order.

In this work, we first analyze two existing models in the literature proposed by Adomavicius and Kwon (2014) which take into account the *diversity-in-top-N* metric along with accuracy and discuss their properties from a multi-objective optimization point of view. Then, we propose an integer programming formulation that considers both objectives by controlling their trade-off via a penalty parameter. Hence, our first contribution is a thorough analysis that fills the gap in the multi-objective optimization problems handling both accuracy and *diversity-in-top-N*. Our second contribution is a novel model that incorporates the distributional diversity metrics, which is a slight modification of our first proposed model. The special structure of the proposed formulation allows it to be solved as a linear program. Keeping the optimization models within the confines of linear programming is vital since the efficient solution algorithms for these problems make it possible to solve large-scale instances. The last contribution of this paper is the examination of algorithms designed for real-life instances, which are too large to be handled directly by mathematical programming solvers. We propose a column generation algorithm to solve the original proposed model. Moreover, Lagrangian relaxation is applied to this model, and the subgradient method that does not necessitate a mathematical programming solver is explained. The computational experiments reveal that our proposed model is flexible to capture the trade-off between the objectives and can attain very high levels of distributional diversity without compromising dramatically on the *diversity-in-top-N* metric and the accuracy of recommendations.

## 2. Literature Review

Recommender systems can be grouped in two main categories: content-based recommendation (Lops et al. 2011) and collaborative filtering (Desrosiers and Karypis 2011, Koren and Bell 2011). In content-based approaches, the interests of the users are modeled based on the features of the items rated or consumed by the users in the past. For example, in a book recommender system, the user interest model might be built based on the analysis of the text of books a user liked and disliked in the past. Users are recommended items which are similar in content to the items they preferred in the past. One drawback of the content-based approaches is the difficulty of gathering the content information of items. Collaborative filtering approaches are not based on the content of items. In collaborative filtering, the users are recommended items based on like-minded users. There are two main approaches to collaborative filtering: neighborhood-based (item-based or user-based) and model-based. In the item-based approaches (Desrosiers and Karypis 2011), user $u$ is recommended items similar to the ones $u$ liked in the past. On the other hand, in the user-based approaches (Desrosiers and Karypis 2011), user $u$ is recommended items that are liked by similar users to $u$. Model-based approaches build a model of each user based on her past behaviour. For example, the matrix factorization models (Koren and Bell 2011, Hu et al. 2008) map each user and item to a latent factor space of dimensionality $f$. The following is a simple matrix factorization model (dubbed as SVD in Koren and Bell (2011)) for predicting the ratings of the users on individual items:

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u \tag{4}$$

where $\mu$ is the average rating over all items, $b_i$ represents the rating bias of item $i$, and $b_u$ represents the rating bias of user $u$. In this model, each user $u \in U$ is associated with a vector $p_u \in \mathbb{R}^f$, and each item $i \in I$ is associated with a vector $q_i \in \mathbb{R}^f$. The values of these vectors represent to what extent the user/item possesses the corresponding factor. The predicted rating of user $u$ on item $i$ is computed as described in (4). The model is typically learned by minimizing the regularized squared error cost function described in (5).

$$\min_{q*,p*} \sum_{(u,i)\in K} (r_{ui} - \mu - b_i - b_u - q_i^T p_u)^2 + \lambda(b_i^2 + b_u^2 + \|q_i\| + \|p_u\|) \tag{5}$$

where $K$ is the set of all known ratings and $\lambda$ is the regularization factor.

In a typical recommender system, recommendations for user $u$ are produced in two phases: in Phase I the ratings of items which are not rated by $u$ are predicted using a recommender algorithm and then, in Phase II the top-$N$ items whose predicted ratings are the highest are recommended to user $u$. Naturally, the success of a recommender system is thought to be determined by the accuracy of the predictions. Hence, most of the work in recommender systems is devoted to develop algorithms which make more accurate predictions.

However, now it has been recognized that accurate recommendations is not the only property a successful recommender system should have (Herlocker et al. 2004, McNee et al. 2006). For example, in the context of movie recommendation, a recommender system which makes recommendations to a user all from the same genre (even if it is very accurate) will not satisfy the user. The recommendations should be not only accurate but also novel and serendipitous. In this regard diversification of recommendation lists is important. As we defined in Section 1 there are two dimensions of diversity: individual and aggregate. Most of the works on diversity have focused on improving individual diversity (Smyth and McClave 2001, Ziegler et al. 2005, Bradley and Smyth 2001, Hurley and Zhang 2011, Aytekin and Karakaya 2014). However, recently Adomavicius and Kwon (2012) and Adomavicius and Kwon (2014) directly address the issue of aggregate diversity. In Adomavicius and Kwon (2012), the authors developed a ranking-based heuristic algorithm to improve aggregate diversity. In Adomavicius and Kwon (2014), the authors proposed three methods to improve aggregate diversity: an iterative approach, a max-flow-based method, and an integer programming approach. The experimental results reported in Adomavicius and Kwon (2014) show that as the methods get more complex, their diversity-accuracy performance increases with a deterioration in the running time performance.

One important point about the works described in Adomavicius and Kwon (2012, 2014) is that all of the three methods are directed to improve the *diversity-in-top-$N$* metric as defined in (1). However, as a side effect reported by the authors, these methods also have a positive impact on the distributional diversity metrics. The model we propose in this work given in Section 3 incorporates distributional diversity metrics given in (2) and (3) as an objective. We will show that the proposed model can capture diversity-accuracy objectives and reaches distributional diversity levels much higher than the best performing method (integer programming) described in Adomavicius and Kwon (2014). Moreover, the column generation method explained in Section 4 has superior running time performance which makes it highly scalable.

## 3. Mathematical Models

In this section, we consider the formulations of the recommendation problem that takes into account both accuracy and the aggregate diversity. In the domain of recommender systems one of the most commonly used measures for accuracy is *precision* which is defined as the ratio of the number of recommended items which are relevant over all recommended items. Recall the two phase process of recommendation. In Phase I all unrated items of all users are predicted. In order to maximize precision, in Phase II, users are recommended those items which have the highest predicted rating values. Since our methods enter the scene in Phase II, similar to Adomavicius and Kwon (2014), we will use the following metric as a proxy for *precision*:

$$Prediction\text{-}in\text{-}top\text{-}N = \frac{\sum_{u\in U}\sum_{i\in L(u)} R^*_{ui}}{\sum_{u\in U}|L(u)|} \tag{6}$$

where $R^*_{ui}$ is the predicted rating of user $u$ for item $i$. This metric can easily be computed at the time of recommendation and there is no need to run a cross-validation experiment which is needed for computing *precision*. In the mathematical models given below, the maximization of accuracy only contains the nominator of (6), which is the total rating, since the denominator is a constant.

Adomavicius and Kwon (2014) proposed mathematical models that take into account aggregate diversity. In this section, we explain their proposed models in our own notation, discuss the characteristics of these models from the multi-objective optimization point of view, and lastly propose our models. Recall that $U$ and $I$ denote the set of users and items, respectively. Moreover, the estimated rating of item $i \in I$ for user $u \in U$ is $R^*_{ui}$ which is a positive real number within a given scale depending on the recommender system. In some cases, the recommender system rules out $(u,i)$ pairs for which $R^*_{ui} < H$ where $H$ is a user-defined threshold. This elimination both reduces the instance size and prevents recommendation of irrelevant items. To that end, two sets are introduced; $I^u \subset I$ is the set of items that can be recommended to user $u \in U$ and $U^i \subset U$ is the set of users to whom item $i$ can be recommended. $x_{ui}$ is a binary variable that indicates whether user $u \in U$ contains item $i \in I^u$ in its recommendation list.

### 3.1. Accuracy and *Diversity-in-top-N*

The recommendation problem that takes into account only the accuracy of the recommendations is to determine the list of recommendations of size $N$ for each user $u \in U$ that maximizes the total estimated ratings, which also maximizes accuracy. The solution of the mathematical model is trivial since selecting top-$N$ items with the largest $R^*_{ui}$ for each user $u \in U$ is sufficient. When the aggregate diversity maximization is also desired, the resulting problem becomes a multi-objective optimization problem, which can be described as follows:

$$\text{maximize} \quad f_1(x) = \sum_{u\in U}\sum_{i\in I^u} R^*_{ui}x_{ui}, \tag{7}$$

$$\text{maximize} \quad f_2(x) = \sum_{i\in I} z_i, \tag{8}$$

$$\text{subject to} \quad \sum_{i\in I^u} x_{ui} = min(|I^u|, N), \qquad u \in U, \tag{9}$$

$$\sum_{u \in U^i} x_{ui} \geq z_i, \qquad i \in I, \tag{10}$$

$$x_{ui} \in \{0,1\}, \qquad u \in U, i \in I^u. \tag{11}$$

$$z_i \in \{0,1\}, \qquad i \in I, \tag{12}$$

where binary variable $z_i$ for $i \in I$ indicates whether an item is recommended to any user. Constraints (9) impose that the number of items recommended to each user is minimum of $N$ and the cardinality of the item set viable for that user. The left-hand-side of (10) represents $rec(i)$ for $i \in I$, and $rec(i) \geq 1$ induces $z_i = 1$ due to the positive coefficient of $z_i$ in objective function $f_2(x)$. The coefficient matrix formed by (9)-(10) possesses the total unimodularity property since each column associated with a variable has at most two ones each of which is in one of the constraint sets. Namely, $x_{ui}$ has one (+1) coefficient in each of (9) and (10) and the other entries are zero, and $z_i$ has only one (-1) coefficient in (10). Given that the right-hand-sides of the constraint sets are integral, this property ensures that the extreme points of the feasible region induced by these constraints are integral. Hence, solving the linear programming relaxation of this model gives an integer solution. Despite of this property, (7)-(12) is still a multi-objective integer programming problem, and the literature on these problems mainly focuses on the generation of the nondominated points (Sylva and Crema (2004), Özlen and Azizoğlu (2009), Lokman and Köksalan (2013)), which requires a considerable effort. In this paper, we analyze and propose multi-objective optimization models arising in recommender systems, which are generally very large-scale, and we do not aim to generate the complete set of nondominated points though the ability of the proposed methods will still be scrutinized for such capability.

The scalar function $f_1(x)$ calculates the total rating, which corresponds to accuracy, and is maximized in (7). $f_2(x)$ maximized in (8) is the number of items that are recommended at least once, which is equivalent to *diversity-in-top-N*, a metric focusing on the breadth of the recommendation of the items. These objectives conflict with one another, i.e., increasing one objective deteriorates the other. Hence, instead of the term *optimality*, the term *Pareto optimality* is employed in the multi-objective optimization terminology. A feasible solution $\hat{x}$ is said to be Pareto optimal if there does not exist any other solution that has no smaller values in accuracy and the aggregate diversity metrics with at least one of the objectives being strictly larger. Two variants of this term are

- A feasible solution $\hat{x}$ is weakly Pareto optimal if there is no other feasible solution $x \neq \hat{x}$ such that $f_1(x) > f_1(\hat{x})$ and $f_2(x) > f_2(\hat{x})$.
- A feasible solution $\hat{x}$ is strict Pareto optimal if there is no other feasible solution $x \neq \hat{x}$ such that $f_1(x) \geq f_1(\hat{x})$ and $f_2(x) \geq f_2(\hat{x})$.

If two feasible solutions $x_1$ and $x_2$ satisfy $f_i(x_1) \geq f_i(x_2)$ for $i = 1, 2$, then $x_1$ dominates $x_2$, and the objective vector $f(x_1) = (f_1(x_1), f_2(x_1))$ dominates $f(x_2) = (f_1(x_2), f_2(x_2))$. If a feasible solution $x$ is Pareto optimal, $f(x)$ is called a nondominated point.

The most prevalent approaches for the multi-objective optimization are $\epsilon-$constraint and weighted sum methods both of which turn the original problem with multiple objectives into a single objective one through scalarization. The weighted sum method maximizes a single objective function constructed by multiplying $f_1(x)$ and $f_2(x)$ with weights satisfying $\lambda_1, \lambda_2 \geq 0$ and $\lambda_1 + \lambda_2 = 1$. If $\lambda_1, \lambda_2 > 0$, the optimum solution of the single objective problem is Pareto optimal. On the other hand, the $\epsilon-$constraint method involves choosing one of the objectives to be maximized and restricting the value of the other objective by a lower bound in a constraint. Given an $\epsilon$ value as a lower bound on one of $f_1(x)$ or $f_2(x)$, the optimal solution obtained by this method is weakly Pareto optimal unless it is the unique optimal solution for the selected objective, in which case the solution is Pareto optimal. The interested reader is referred to Chankong and Haimes (1983) and Ehrgott (2005) for the details of the multi-objective optimization and techniques to solve these problems. In Ehrgott (2006), the authors evaluate the characteristics of the scalarization methods applied to the multi-objective integer programming problems and the drawbacks of these methods, which are explained as follows: The weighted sum method has the computational advantage of keeping the number of constraints unchanged, unlike the $\epsilon-$constraint method which adds a new constraint to put a lower bound on the value of one of the objective functions. However, the $\epsilon-$constraint method can find all Pareto optimal solutions, which is not possible using the weighted sum method. The authors also proposed a new scalarization referred to as the elastic constraint method –as given later in this section– that alleviates these drawbacks, and in Ehrgott and Ryan (2002), this method is applied to the robust crew scheduling problem. The approach that we employed in this paper is reminiscent of the elastic constraint method. Therefore, the characteristics of the solutions of our models will be defined according to the properties of this method as explained in Ehrgott (2006).

Adomavicius and Kwon (2014) proposed two integer programming models using the $\epsilon-$constraint method, each of which aims at maximizing one of the objectives while the level of the other is bounded from below by a prespecified value in a constraint. The first model that maximizes the *diversity-in-top-N* metric is as follows:

$$\textbf{(M1)} \quad \text{maximize} \quad \sum_{i \in I} z_i \tag{13}$$

$$\text{subject to} \quad \sum_{i \in I^u} x_{ui} = min(|I^u|, N), \qquad u \in U, \tag{14}$$

$$\sum_{u \in U^i} x_{ui} \geq z_i, \qquad i \in I, \tag{15}$$

$$\sum_{u \in U} \sum_{i \in I^u} (R^*_{ui} - A) x_{ui} \geq 0, \tag{16}$$

$$x_{ui} \in \{0, 1\}, \qquad u \in U, i \in I^u. \tag{17}$$

$$z_i \in \{0,1\}, \qquad i \in I, \qquad (18)$$

where $A$ is a positive scalar that represents the desired accuracy level. Summing (14) over all $u \in U$, we get $\sum_{u\in U}\sum_{i\in I^u} x_{ui} = \sum_{u\in U} min(|I^u|, N)$, which is the total number of recommendations in the system denoted by $T$. (16) imposes that the average rating of the selected items be larger than or equal to $A$, since this constraint can be written as $\frac{1}{T}\sum_{u\in U}\sum_{i\in I^u} R^*_{ui}x_{ui} \geq A$. The objective (13) maximizes the number of items that are recommended at least once, equivalently *diversity-in-top-N*. The maximum possible objective value, also the maximum *diversity-in-top-N* level, is equal to the number of items, $|I|$. For a given accuracy level $A$, there may be many alternative optimal solutions for the objective (13) so that the one found by M1 is weakly Pareto optimal unless it is the unique optimal solution, as mentioned previously. Moreover, as a consequence of applying the $\epsilon-$constraint method which adds constraint (16), the total unimodularity of the constraint matrix formed by (14)-(15) is lost, which was also discussed previously as one of the drawbacks of this method. Being an integer programming problem, M1 may induce long solution times and memory problems for large instances.

The second model presented in Adomavicius and Kwon (2014) is obtained by the application of the $\epsilon-$constraint method in which accuracy is maximized. This model is as follows:

$$\begin{aligned}
\textbf{(M2)}\quad & \text{maximize} && \sum_{u\in U}\sum_{i\in I^u} R^*_{ui}x_{ui} && && (19)\\
& \text{subject to} && \sum_{i\in I^u} x_{ui} = min(|I^u|, N), && u \in U, && (20)\\
& && \sum_{u\in U^i} x_{ui} - z_i \geq 0, && i \in I, && (21)\\
& && \sum_{i\in I} z_i \geq D, && && (22)\\
& && x_{ui} \in \{0,1\}, && u \in U, i \in I^u. && (23)\\
& && z_i \in \{0,1\}, && i \in I. && (24)
\end{aligned}$$

where $0 \leq D \leq |I|$ is a prespecified parameter that imposes a certain level of *diversity-in-top-N*. This model aims at maximizing the accuracy level with a certain aggregate diversity level. The higher the level of *diversity-in-top-N* metric prescribed by $D$ is imposed, the smaller the total rating and the accuracy level we get, if not stay the same. When $D = |I|$, M2 achieves the maximum possible *diversity-in-top-N* level. However, if the maximum *diversity-in-top-N* level, denoted by $D^*$, is lower than $|I|$, setting $D > D^*$ in (22) results in an infeasible solution. Although new constraint (22) is added to constraints (9) and (10) whose coefficient matrix is totally unimodular, this property is preserved since $z_i$ has now one (-1) coefficient in (21) and one (+1) coefficient in (22).

We now employ a different approach to the above multi-objective optimization problem. Objective function $f_2(x)$ corresponding to the *diversity-in-top-N* metric takes the maximum possible value, $|I|$, when $z_i = 1$ for all $i \in I$. Therefore, we can modify M2 by omitting constraint (22) and fixing all $z_i$ to one. This approach is called the lexicographic optimization in which the optimal solution is found for the primary objective, and its objective value is set as a lower bound in a constraint for the optimization problem with the secondary objective. In other words, among the alternative optimal solutions for the primary objective, the one that gives the best evaluation for the secondary objective is selected. In our case, through the modifications explained above, we ensure that the maximum level of the *diversity-in-top-N* metric is obtained without imposing (22). We note that the resulting problem may not have a feasible solution after fixing all $z_i$ to one if the total number of recommendations, $T$, is small. This issue can be circumvented by adding artificial variables to the left-hand-sides of (21), which renders them soft (elastic) constraints. The resulting model becomes

$$\textbf{(M3)} \quad \text{maximize} \quad \sum_{u \in U} \sum_{i \in I^u} R^*_{ui} x_{ui} - \sum_{i \in I} m_i r_i \tag{25}$$

$$\text{subject to} \quad \sum_{i \in I^u} x_{ui} = min(|I^u|, N), \qquad u \in U, \tag{26}$$

$$\sum_{u \in U^i} x_{ui} + r_i \geq 1, \qquad i \in I, \tag{27}$$

$$0 \leq x_{ui} \leq 1, \qquad u \in U, i \in I^u, \tag{28}$$

$$r_i \geq 0, \qquad i \in I, \tag{29}$$

where $r_i$, $i \in I$, is an artificial variable which gets value one if the corresponding constraint in (27) is not satisfied, i.e., item $i$ is not recommended to any user, and $m_i$ is the penalty parameter associated with this variable. Observe that in the above model, $f_2(x)$ is defined in a different way as $f_2(x) = -\sum_{i \in I} r_i$ and its level is not imposed through a constraint, unlike in M2 which uses $\epsilon-$constraint method with parameter $D$. The reason is that the maximum level of $f_2(x)$ is already imposed by constraints (27) which are softened by adding artificial variables with a penalty parameter. This method is similar to the elastic constraint method –as explained previously– that contains the upsides of both weighted sum and $\epsilon-$constraint methods. As shown in Ehrgott (2006), these methods are the special cases of the elastic constraint method. (25) contains $f_1(x)$ with weight one and $f_2(x)$ with weights $m_i$, $i \in I$ so that M3 is also a weighted sum problem. While large values of $m_i$ for all $i \in I$ cause small, if not zero, violations in (27) and large *diversity-in-top-N*, small values entail large violations and small *diversity-in-top-N*. Moreover, when $m_i$ for all $i \in I$ are set to large values in the objective function, M3 becomes the $\epsilon-$constraint problem given in M2 in which $D = D^*$. In this case, M3 generates Pareto optimal solutions since it reaches the maximum possible total rating at the maximum *diversity-in-top-N* level. When $m_i$ is set to zero for all $i \in I$, accuracy is maximized as the sole objective which can be achieved by selecting the top-$N$ items. Applying the result that was given in Ehrgott (2006), we state that a solution found by M3 is weakly Pareto optimal unless it is unique or $r_i > 0$ for some $i \in I$.

Solving M3 by varying the values of penalty parameter $m$ generates a collection of (weakly) Pareto optimal solutions. As will be shown in the results of the computational experiments, increasing the values of $m_i$, $i \in I$, has an adverse effect on the computational performance of the model since constraints (27) then turn into hard constraints. To achieve a solution for a prespecified level $D$ of *diversity-in-top-N* which was imposed by (22) in M2, the value of penalty parameter $m$ can be set to a large value for $D$ randomly selected $r-$variables and to a small value or zero for the rest of them. Instead of random selection of items, it is possible to distinguish between the items, which enables the recommender to incorporate preferences for items in the system by defining distinct $m$ values for them. Parameter $m_i$ for a given item $i \in I$ is a penalty term deteriorating the objective function if item $i$ is not recommended at all. It controls the trade off between the objective functions and can be interpreted as the largest allowable decrease in the total rating in order to recommend item $i$.

### 3.2. Maximization of Distributional Diversity

The above models are not designed to maximize the distributional diversity metrics which get large values as the items are evenly recommended across all users. Although the distributional diversity increases together with *diversity-in-top-N*, both of which are metrics used to measure the aggregate diversity, it may be low even at the maximum level of *diversity-in-top-N* which will be shown in Section 5. In the rest of the section, instead of handling distributional diversity indirectly through *diversity-in-top-N* which has only a limited effect, we divert to maximizing the distributional diversity directly by considering it as another objective in addition to the accuracy and *diversity-in-top-N* maximization.

A new objective function that is designed for the maximization of the distributional diversity metrics given in (2) and (3) may be integrated in one of the multi-objective optimization methods to generate Pareto optimal solutions for all three objectives. On the other hand, the efforts in incorporating a new objective to our multi-objective optimization problem should not render the mathematical model more complex for the sake of the scalability. Therefore, we will strive to modify model M3 in the minimal way so as to incorporate the new objective.

When the total estimated ratings is maximized in a single objective setting by selecting the top-$N$ items, a small percentage of the items classified as popular items accounts for almost all recommendations which sum up to $T$, and a large bulk of the items classified as unpopular items are not recommended at all, which was illustrated in Figure 1. We observed in the computational experiments that model M3 that is solved with large values of $m$ to maximize accuracy at the maximum *diversity-in-top-N* level produces solutions in which the unpopular items are recommended only once, if not at all. Hence, the tail of the curve in Figure 1 is leveraged to one, and the peak for the popular items smooths out slightly. As the right-hand-side of (27) is increased to a larger value, the following pattern can be observed in the distribution of the recommendations: the tail of the distribution which contains the unpopular items is leveraged to the right-hand-side value of (27) and the peak of the popular items is trimmed. Hence, the larger the right-hand-side value of (27), the more even the items are distributed in the recommendation lists of the users.

On the other hand, analyzing the formulation of the distributional diversity metrics given in (2) and (3), we can infer that the maximum level of the distributional diversity can be achieved when the distribution of the total recommendations $T$ to items is uniform, i.e., $rec(i) = \frac{T}{|I|}$ for all $i \in I$. Such a solution may not be possible since each item has only a subset of ratings available for each user $u \in U$, defined in $I^u$, and for some of the items $i \in I$, $rec(i) < \frac{T}{|I|}$. The solutions associated with the two cases, one with the uniform distribution of recommendations and the other illustrated in Figure 1, correspond to two extreme Pareto optimal solutions for the distributional diversity and accuracy maximization objectives, respectively. Moreover, if the right-hand-side value of (27) is larger than zero, the maximum *diversity-in-top-N* level is also achieved as long as $T$ and $|I^u|$, $u \in U$, are sufficiently large. To find the (weakly) Pareto optimal solutions so as to depict the trade-off between accuracy and the aggregate diversity metrics, the right-hand-side of (27) is varied between 0 and $\left\lceil \frac{T}{|I|} \right\rceil$ together with the values of $m$. However, as pointed out previously, the complete enumeration of the nondominated points is not aimed for this multi-objective problem since modeling an optimization problem that explicitly maximizes the distributional diversity eradicates the properties of the model that allows the solution of the linear programming relaxation. Our proposed model that incorporates the distributional diversity is

$$\textbf{(M4)} \quad \text{maximize} \quad \sum_{u\in U}\sum_{i\in I^u} R^*_{ui}x_{ui} - \sum_{i\in I} m_i r_i \tag{30}$$

$$\text{subject to} \quad \sum_{i\in I^u} x_{ui} = min(|I^u|, N), \qquad u \in U, \tag{31}$$

$$\sum_{u\in U^i} x_{ui} + r_i \geq \bar{z}, \qquad i \in I, \tag{32}$$

$$0 \leq x_{ui} \leq 1, \qquad u \in U, i \in I^u, \tag{33}$$

$$r_i \geq 0, \qquad i \in I, \tag{34}$$

where $\bar{z}$ is a parameter that will determine the level of the distributional diversity. M3 is a special instance of M4 in which $\bar{z}$ is set to one. The value of $m_i$ for a given item $i \in I$ now represents the penalty of falling short of the given level of recommendations $\bar{z}$. For large values of penalty parameter

$m$, (32) becomes a hard constraint set, and the solutions of M4 lean toward the distributional diversity objective with a certain decrease in accuracy depending on the value of $\bar{z}$. When $\bar{z} > 1$, it is possible even for large values of $m$ that the artificial variable $r_i$, $i \in I$, can take values larger than one. This occurs if $T = \sum_{u \in U} max(|I^u|, N) < \bar{z}|U|$, where $\bar{z}|U|$ is the required number of recommendations, or $|I^u| < \bar{z}$ for some $u \in U$, i.e. user $u$ does not have sufficient number of items to match the imposed level of $\bar{z}$. For smaller values of $m$ comparable to the estimated ratings of the items, the value of $r_i$ larger than one may inherently mean that the decrease in the total rating by imposing $\bar{z}$ recommendations for item $i$ cannot be compensated by the penalty inflicted by $m_i$. Since the constraint coefficient matrix of M4 is totally unimodular, this model is written as a linear programming problem, and experimenting over the various values of $\bar{z}$ and $m$ is not computationally intractable. As discussed previously, $\left\lceil \frac{T}{|I|} \right\rceil \geq \bar{z} \geq 0$. Varying the value of $\bar{z}$ within this interval would give the recommender the flexibility to observe the trade-off between the accuracy and diversity and to select the solution that satisfies the needs of the system. When $\bar{z} = 0$, the objective function value will be equal to that obtained by selecting the top-$N$ items. We will show through computational experiments that for a given value of $m$, increasing $\bar{z}$ monotonously increases the distributional diversity, and vice versa. For large values of both of the parameters, the distribution of the recommendations across the items approaches uniform with a certain decrease in the total rating.

## 4. Solving the Proposed Model

M4 is a linear programming problem, and the large-scale instances of this problem can be solved very efficiently by available solvers. The number of variables is $\sum\limits_{u \in U} |I^u| + |I|$, and the number of constraints is $|U| + |I|$. On the other hand, the size of the instances may be so large that the complete mathematical model may not be solved. For such large-scale instances of linear programming problems, an important observation is that only a small proportion of the whole variable set takes non-negative values at the optimal solution. This phenomenon led to column generation, a prominent algorithm to solve large-scale linear programming problems, pioneered by Dantzig and Wolfe (1960) and Gilmore and Gomory (1961). This algorithm keeps a small subset of variables and generates the promising columns that potentially improve the solution on the fly. In this section, we first propose a column generation algorithm to solve M4 to refrain from handling the large variable set of the problem completely. Moreover, the structure of M4 makes it amenable to Lagrangian relaxation (See Fisher (1981) for the details). In the second part of this section, we explain the steps of decomposition applied to our proposed model M4, and propose a Lagrangian relaxation application to this problem which results in a procedure that does not require a mathematical programming solver.

### 4.1. Column Generation

Column generation involves forming a restricted master problem (RMP) by selecting a small subset of columns, and then adding columns with positive reduced cost (for a maximization problem) by solving a pricing subproblem. When the pricing subproblem can no longer generate a positive reduced cost column, the solution of the RMP is also optimal for the original problem. In our application, we select the top-$N$ items with the highest ratings for each user to initialize the RMP, which also gives the optimal solution of M4 for $\bar{z}=0$. In order to construct the pricing subproblem, we first give the dual constraint corresponding to $x-$variables:

$$\alpha_u + \beta_i \geq R^*_{ui}, \qquad u \in U, i \in I^u, \tag{35}$$

where $\alpha_u \in \mathbb{R}$, $u \in U$ and $\beta_i \in \mathbb{R}_-$, $i \in I$ are the dual variables corresponding to (31) and (32), respectively. After solving the RMP, the values of the dual variables are obtained, and the pricing subproblem given below that finds the $x-$variable with the maximum reduced cost is solved to check whether any dual constraint in (35) is violated:

$$\tau = \max_{u \in U, i \in I^u} \left( R^*_{ui} - \alpha_u - \beta_i \right). \tag{36}$$

If $\tau > 0$, a new variable $x_{ui}$, $u \in U$ and $i \in I$ that has the maximum reduced cost is added to the RMP. Otherwise, the solution of the RMP is also optimal for M4. Instead of adding variables one-by-one, we solve (36) for each $u \in U$, and add the variable with the largest positive reduced cost to the RMP. Adding a bulk of variables simultaneously to the RMP generally enhances the performance of the column generation procedure by reducing the number of iterations. Moreover, when solving M4 for $\bar{z} > 0$, we initialize the RMP with the set of columns existing in the RMP at the optimal solution obtained for $\bar{z} - 1$, which provides a warm start and reduces the number of iterations. Although this strategy may increase the number of variables for large values of $\bar{z}$, we did not observe any performance deterioration due to the relatively small size of the RMP.

### 4.2. Lagrangian Relaxation

Constraint set (31) has a block-diagonal structure that is separable for each $u \in U$. Any solution that satisfies (31) for some $u \in U$ corresponds to $min(|I^u|, N)$ items from the set $I^u$. To exploit such a structure, the constraint set (32), referred to as the set of complicating constraints, is dualized in the objective function with a set of multipliers $\lambda \in \mathbb{R}^{|I|}_-$, referred to as the Lagrangian multipliers. The problem resulting from Lagrangian relaxation is

$$L(\lambda) = \text{maximize} \quad \sum_{u \in U} \sum_{i \in I^u} R^*_{ui} x_{ui} + \sum_{i \in I} \lambda_i (\bar{z} - \sum_{u \in U^i} x_{ui}) \tag{37}$$

$$\text{subject to} \quad \sum_{i \in I^u} x_{ui} = min(|I^u|, N), \qquad u \in U, \tag{38}$$

$$0 \le x_{ui} \le 1, \qquad u \in U, i \in I^u, \qquad (39)$$

which does not contain the artificial variable $r_i$ since the Lagrangian multipliers serve the same purpose of the penalization of the constraint violations. The objective function of the model can be rearranged as $\sum_{u \in U} \sum_{i \in I^u} (R^*_{ui} - \lambda_i) x_{ui} + \sum_{i \in I} \lambda_i \bar{z}$, which involves updating the ratings of the items with the given values of $\lambda$. For any given $\lambda \in \mathbb{R}^{|I|}_{-}$, $L(\lambda)$ gives an upper-bound to the optimal objective function value of M4. In order to find the best upper-bound, the following problem is posed

$$L_D = \min_{\lambda \in \mathbb{R}^{|I|}_{-}} L(\lambda) \qquad (40)$$

which is called the Lagrangian dual problem. (40) is a minimization problem over the Lagrangian multipliers, while (37)-(39) is a maximization problem over the original $x-$variables. $\bar{x}$, which is obtained by solving (37)-(39) for some $\lambda \in \mathbb{R}^{|I|}_{-}$, is the optimal solution of M4 if the following conditions are satisfied

- the complementary slackness condition $\sum_{i \in I} \lambda_i (\bar{z} - \sum_{u \in U^i} x_{ui}) = 0$, i.e., $\lambda_i = 0$ for $i \in I$ satisfying $(\bar{z} - \sum_{u \in U^i} x_{ui}) > 0$ and $(\bar{z} - \sum_{u \in U^i} x_{ui}) = 0$ for $i \in I$ satisfying $\lambda_i < 0$,
- $\sum_{u \in U^i} x_{ui} \ge \bar{z}$ for each $i \in I$ (possible only if $r_i = 0$, $i \in I$ in the optimal solution of M4).

The Lagrangian dual problem can be written as a linear programming problem by enumerating the extreme point solutions, indexed by $Q$, of the polyhedron induced by (38). For each $u \in U$, extreme point solution $q \in Q^u$ corresponds to a selection of $min(|I^u|, N)$ items from the set $I^u$, and $\mathbf{x}^u_q$ is the corresponding solution vector consisting of $x^{ui}_q$ for $i \in I$. Parameter $x^{ui}_q = 1$ if item $i$ is recommended to user $u$ in solution $q$ and 0, otherwise. We can write the Lagrangian dual problem as

$$L_D = \min_{\lambda \in \mathbb{R}^{|I|}_{-}} (\sum_{i \in I} \lambda_i \bar{z} + \sum_{u \in U} \max_{q \in Q^u} (\sum_{i \in I^u} R^*_{ui} x^{ui}_q - \sum_{i \in I^u} \lambda_i x^{ui}_q)), \qquad (41)$$

and the linearization of this problem is

$$L_D = \text{minimize} \quad \sum_{i \in I} \bar{z} \lambda_i + \sum_{u \in U} Z^u \qquad (42)$$

$$\text{subject to} \quad Z^u + \sum_{i \in I^u} x^{ui}_q \lambda_i \ge \sum_{i \in I^u} R^*_{ui} x^{ui}_q, \qquad u \in U, q \in Q^u, \qquad (43)$$

$$\lambda_i \le 0, \qquad i \in I. \qquad (44)$$

This model has many constraints corresponding to the extreme points of the polyhedron induced by (38), which necessitates constraint generation algorithm for its solution.

Letting $p^u_q$ be the dual variable associated with (43), the dual of (42)-(44) can be written as

$$\text{maximize} \quad \sum_{u \in U} \sum_{q \in Q^u} (\sum_{i \in I^u} R^*_{ui} x^{ui}_q) p^u_q \qquad (45)$$

$$\text{subject to} \quad \sum_{u\in U^i}\sum_{q\in Q^u} x_q^{ui} p_q^u \geq \bar{z}, \qquad i\in I, \tag{46}$$

$$\sum_{q\in Q^u} p_q^u = 1, \qquad u\in U, \tag{47}$$

$$p_q^u \geq 0, \qquad u\in U, q\in Q^u. \tag{48}$$

The above model is defined over a set of variables $p_q^u$, $q\in Q^u$ and can also be obtained by the application of the Dantzig-Wolfe decomposition to M4 (Lübbecke and Desrosiers 2005). This problem is referred to as the master problem, and (47) is the convexity constraint set. (45)-(48) aims to find the convex combination of the extreme point solutions for each user such that each item is recommended at least $\bar{z}$ times and the total rating is maximized. The large column set of this model makes column generation a viable method, which generates the variables in the extreme point set $Q^u$ by solving a pricing subproblem for each $u\in U$ which can be written as

$$\text{maximize} \quad \sum_{i\in I^u}(R^*_{ui}-\beta_i)x_{ui} - \gamma_u \tag{49}$$

$$\text{subject to} \quad \sum_{i\in I^u} x_{ui} = min(|I^u|, N), \tag{50}$$

$$0 \leq x_{ui} \leq 1, \qquad i\in I^u, \tag{51}$$

where $\beta_i \in \mathbb{R}_-$, $i\in I$ is the dual variable associated with (46), as defined earlier in the pricing subproblem of M4, which coincide with $\lambda_i$ at the optimal solution, and $\gamma_u \in \mathbb{R}$, $u\in U$ is the dual variable associated with the convexity constraint set. The aggregation of (49)-(51) for $u\in U$ is equivalent to (37)-(39) except for the constant terms in the objective functions (37) and (49).

The Dantzig-Wolfe decomposition is generally utilized to obtain more compact models with smaller number of constraints, and the extended column size is handled by column generation. In this case, decomposition transformed the original model M4 into a formulation with equal number of constraints and with a larger set of columns. Additionally, the column vectors in the decomposed model are denser than those in M4, which renders the solution of the linear programs relatively harder in the column generation iterations. Finally, in the master problem resulting from the decomposition, the coefficient matrix formed by (46) and (47) is not totally unimodular so that the optimal solution reached by column generation is fractional. Hence, column generation must be embedded in a branch-and-bound procedure to reach the integer optimal solution. The resulting method is known as branch-and-price, whose solution time can be prohibitive for large instances. Due to these issues, instead of solving the dual of the Lagrangian dual given in (45)-(48), we handle the Lagrangian dual problem (40).

### 4.3. The Subgradient Method

There are various methods to solve the Lagrangian dual problem, such as the bundle method (Lemaréchal et al. (1981)), the analytic center cutting plane method (Goffin and Vial (2002)) and the subgradient method (Held and Karp (1970), Held and Karp (1971), Held et al. (1974)). The subgradient method is one of the most popular and practical one, and it provides an approximation to the optimal Lagrangian multipliers in an efficient way by updating them at each iteration in the subgradient direction. Even though the computational performance of this algorithm could not match the efficiency of the method given in Section 4.1 in the computational experiments, as well as its accuracy in reaching the primal optimal solution, we explain this methodology as a heuristic, which does not require any usage of a solver, for the sake of completeness.

The subgradient method is the counterpart of the gradient method in nondifferentiable optimization. Instead of the gradient, this method uses the subgradient vector at each point, which is the difference between two sides of the relaxed constraints, as the direction and it moves in this direction with a step size determined at each iteration. As defined in (40), the optimal value of $\lambda$ is the one that minimizes $L(\lambda)$, and at the minimum point, it is equal to the optimal value of the original model, as mentioned previously. The subgradient method starts with the initial values of the multiplier set $\lambda^0$, which is generally set to zero, and at each iteration, the values of the multipliers are updated as

$$\lambda^{k+1} = \lambda^k + t_k(\bar{\mathbf{z}} - \mathbf{x}^k), \tag{52}$$

where $\mathbf{x}^k$, which is the optimal solution of $L(\lambda^k)$, and $\bar{\mathbf{z}}$ are vectors of length $|I|$ consisting of $\sum_{u \in U^i} x_{ui}$ and $\bar{z}$, respectively, at each $i \in I$. $t_k > 0$ is the step size at iteration $k$, which is generally calculated as

$$t_k = \frac{\theta_k(L(\lambda^k) - LB)}{\|\bar{\mathbf{z}} - \mathbf{x}^k\|^2}, \tag{53}$$

where $2 \geq \theta_k > 0$ is a coefficient that is initialized with value two, and is halved when $L(\lambda^k)$ does not decrease in a prespecified number of iterations. $LB$ is a lower bound to the optimal objective function value of M4 which can be obtained by a feasible solution to this problem. The step size gets smaller as $L(\lambda^k)$ converges to the optimal value in the later iterations in which $\theta_k$ also decreases. The algorithm works as follows: Given the values of $\lambda^k$, (37)-(39) is solved by updating the rating of each item $i \in I^u$ for a user $u \in U$ as $(R^*_{ui} - \lambda^k_i)$ and then selecting the top-$N$ items. We update the best upper bound and $\theta_k$, if necessary, and then, with the new step size $t_k$ found in (53), the multipliers are updated in (52). This method is generally terminated after a number of iterations or $\theta_k$ is smaller than a prespecified value. Moreover, the best upper-bound obtained when the algorithm terminates is not necessarily a feasible solution since it may violate the relaxed constraint set of M4, namely (32). Therefore, further efforts are needed to obtain primal feasible solutions, which is outside the scope of this work.

## 5. Computational Results

In this section, we demonstrate and explain the results of the experiments conducted on the mathematical models given in Section 3 except M1 and M3 since the former is an integer programming model, which is computationally expensive to solve and has no control on the aggregate diversity, and the latter is a special instance of M4, as explained previously. First, we explain the data sets that we use and our test environment. Then, for the objective pairs consisting of accuracy and each aggregate diversity metric considered in this paper, the comparative results are given. The models presented in this paper cannot be solved by the available solvers for large-scale instances so that the column generation algorithm given in Section 4.1 is employed, and its results are reported.

We use three data sets for evaluation of the models: MovieLens, Amazon Movies[2], and Amazon Books[3]. MovieLens data set contains 6,040 users, 3,900 movies and 1,000,209 ratings. Amazon Movies contains 16,315 users, 20,928 items and 1,926,354 ratings and Amazon Books contains 21,930 users, 14,906 items and 1,622,105 ratings. All three data sets contain integer ratings from 1 to 5. Original Amazon data sets are very sparse. In order to reduce sparsity, for both data sets, we keep users which rated more than 50 items and keep items which are rated by more than 100 users. In Phase I, we employ the SVD method that was described in Section 2 to predict the unknown ratings, and we keep only $(u,i)$ pairs whose predicted rating $R^*_{ui}$ is larger than $H = 3.5$. Consequently, the number of predicted ratings, which is equal to the number of $x-$variables in the models, are 9,427,051 for MovieLens, 265,994,132 for Amazon Movies and 274,720,816 for Amazon Books instances.

In each experiment, we choose three values for $N$, $N = \{1,5,10\}$. We run the experiments on a computer with a 3.6 GHz Intel Xeon E5-1620 processor and 16 GB of RAM. For the solution of the mathematical models, CPLEX 12.5 was used, and the column generation algorithm presented in Section 4.1 was implemented in C++ using the same solver and Concert 2.5.

First, we give the results on the MovieLens data set. For M2, the application of the $\epsilon$-constraint method to solve the multi-objective optimization problem requires the determination of an interval and increment for the level of *diversity-in-top-N* which is controlled by $D$. In our proposed model M4, the vital parameters are $\bar{z}$ and $m_i$, $i \in I$. The values of these parameters are given in Table 1 except for $m$ which is set to four values, $m = \{0.5,1,2,5\}$. In each experiment, $m_i$ for all $i \in I$ are set to the same value although it is possible to assign distinct values according to the preferences of the items. In the figures that follow, the optimal solutions of M4, and those of both M2 and M4 in the comparative figures, are depicted in a given parameter interval for various pairs of objectives. The number of points for M2 differs in the respective figures since they may be infeasible for some

[2] http://snap.stanford.edu/data/web-Movies.html

[3] http://snap.stanford.edu/data/web-Amazon-links.html

Table 1: Parameter values used for the MovieLens data set.

| | D | $\bar{z}$ |
|---|---|---|
| Min | 1,000 | 0 |
| Max | 3,750 | $\left\lceil \frac{T}{\lvert I\rvert} \right\rceil$ |
| Increment | 250 | 1 |

values of $D$. Moreover, for M4, the maximum value of $\bar{z}$ is determined by $\frac{T}{|I|}$ which depends on $N$ so that there are only few points for small values of $N$. The trade-off between accuracy and one of the diversity metrics achieved by M4 and M2 can be observed. Since accuracy is inversely proportional to the diversity objective, as the values of the parameters that control diversity, namely $D$ and $\bar{z}$, increase, the curves descend. We point out that the uniform distribution of the recommendations to items is achieved when $\bar{z} = \frac{T}{|I|}$. However, if $\frac{T}{|I|}$ is fractional, imposing the largest value of $\bar{z}$ yielded by the ceiling operator may impair the balance in the distribution of recommendations, and may result in a peculiar pattern in the curves depending on the data set. This can be observed at the right tails of some of the upcoming figures, such as Figures 5, 6, 7 and 9. Moreover, the left-most points in the figures correspond to the maximum possible accuracy values resulting from the solution of M4 with $\bar{z} = 0$. Such solutions induce small values for the diversity metrics. Models M2 and M4 are equivalent when $D = 0$ and $\bar{z} = 0$, respectively, since without imposing any lower-bound on the *diversity-in-top-N* level, the solution is equivalent to selecting the top-$N$ items.

In the first experiments, we demonstrate the results of M4 with various values of $\bar{z}$ and $m$ in order to analyze the effect of changing one of them when the other is fixed. In Figure 2, the solutions of M4 are depicted for the accuracy and *diversity-in-top-N* metrics for values of $\bar{z}$ in the given interval with each value of $m = \{0.5, 1, 2\}$. As $m$ increases, the curves shift towards the larger values of *diversity-in-top-N* and smaller values of accuracy. For large values of $m$ and $N$, the *diversity-in-top-N* level obtained by M4 is close to the maximum value $D^*$. On the other hand, for small values of $m$, M4 compromise on *diversity-in-top-N* for accuracy. The accuracy and the distributional diversity values obtained by M4 are depicted in Figures 3 and 4. For a given value of $m$, the level of the distributional diversity rises together with the value of $\bar{z}$. However, this increase in the level of the distributional diversity with $\bar{z}$ is even more striking for larger values of $m$. Thus, the value of $m$ sets a window in which the trade-off between accuracy and diversity can be adjusted through $\bar{z}$, and as $m$ increases, this window enlarges. In the subsequent experiments, $m$ is set to five, which is a sufficiently large value, to show the potential of M4 in increasing the distributional diversity in comparison with M2. Therefore, the primary objective leans to aggregate diversity through satisfying the elastic constraint in M4 as much as possible for the given value of $\bar{z}$.

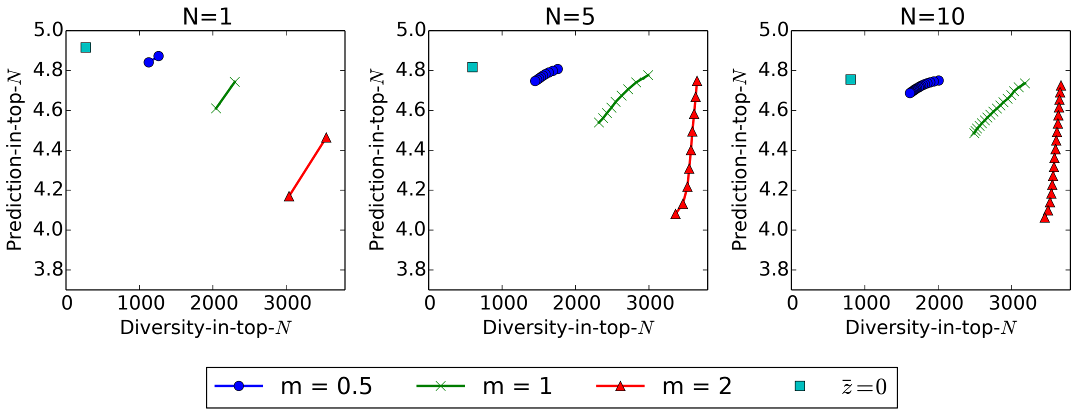

Figure 2: Prediction-in-top-$N$ vs. *Diversity-in-top-N* results of M4 on MovieLens data set for given values of $m$ and $0 \leq \bar{z} \leq \left\lceil \frac{T}{|I|} \right\rceil$.

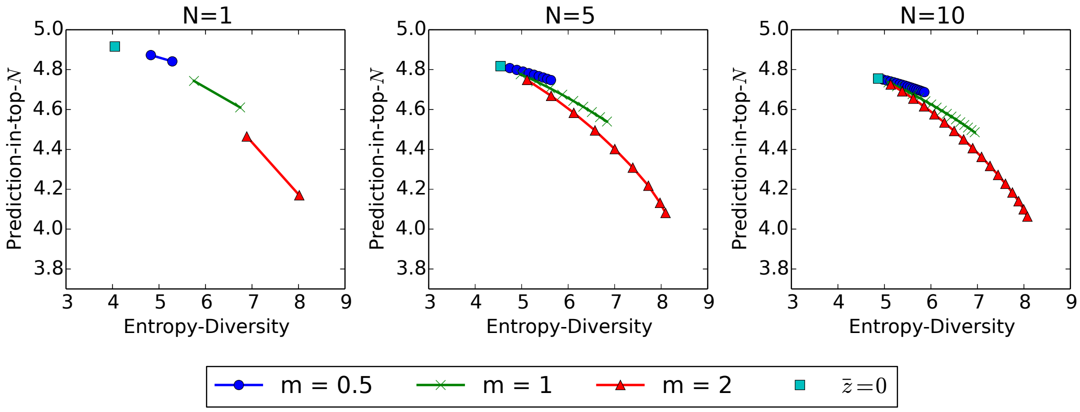

Figure 3: Prediction-in-top-$N$ vs. entropy-diversity results of M4 on MovieLens data set for given values of $m$ and $0 \leq \bar{z} \leq \left\lceil \frac{T}{|I|} \right\rceil$.

In Figure 5, the accuracy and *diversity-in-top-N* curves for M2 and M4 are given for $N = \{1, 5, 10\}$. It can be seen that the point generated by M4 for $\bar{z} = 1$ corresponds to a Pareto optimal solution that reaches the maximum possible *diversity-in-top-N* level while maximizing accuracy at this level. For $N = 5$ and $N = 10$, the *diversity-in-top-N* level achieved by M4 deviates slightly from the maximum possible value $|I|$ as $\bar{z}$ exceeds one. If the primary objective of the recommender system is to maximize *diversity-in-top-N*, solving M4 for $\bar{z} = 1$ is sufficient. On the other hand, as we will show in Figures 6 and 7, larger values of $\bar{z}$ are required to increase the distributional diversity metrics. For all three values of $N$, the points obtained from M2 reside between those generated by M4 with $\bar{z} = 0$ and $\bar{z} = 1$.

Figures 6 and 7 show the results of M2 and M4 for the accuracy and the distributional diversity metrics. Both figures convey similar structures for each value of $N$. As $D$ increases, M2 yields larger

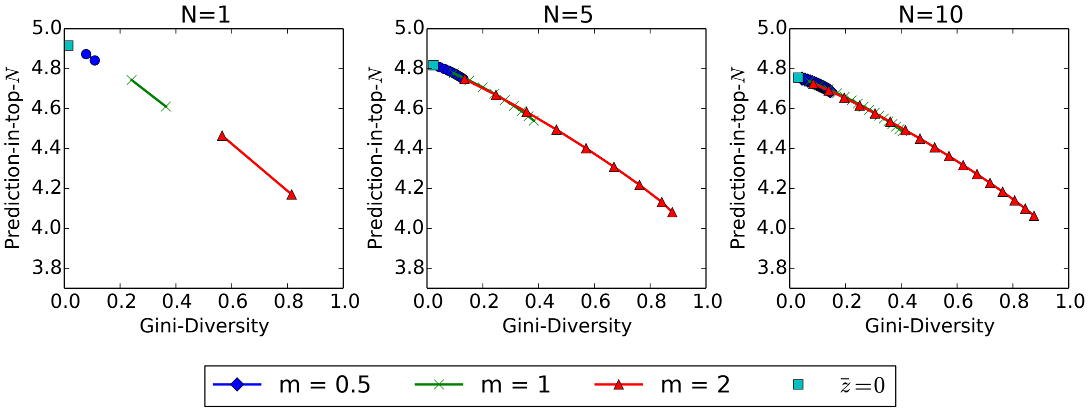

Figure 4: Prediction-in-top-$N$ vs. Gini-diversity results of M4 on MovieLens data set for given values of $m$ and $0 \leq \bar{z} \leq \left\lceil \frac{T}{|I|} \right\rceil$.

distributional diversity values, which indicates that the *diversity-in-top-N* and distributional diversity metrics are correlated. On the other hand, as discussed earlier, the values of both Gini-diversity and entropy-diversity reached by M2 are bounded from above by those achieved by M4 for $\bar{z} = 1$. For large values of $\bar{z}$, M4 reaches very high distributional diversity values with a certain compromise in accuracy. Although there is a certain decrease in accuracy, *prediction-in-top-N* decreases down to at most around 4.0 even at the highest distributional diversity levels. Given that items which get ratings above 3.5 are considered to be highly ranked (or relevant) (Adomavicius and Kwon 2012), the generated accuracy vs. distributional diversity levels by our method can satisfy the needs of many real life applications where distributional diversity is important. Returning back to our job recommendation example, using the proposed method an e-recruitment company can give almost an equal share to the job ads in the recommendation lists while still providing relevant job recommendations to the users. Moreover, the trade-off between accuracy vs. distributional diversity can be adjusted according to the needs of the company by selecting an appropriate value of $\bar{z}$.

Overall, M2 generates Pareto optimal solutions (some of which can be weakly Pareto optimal) between the solutions generated by M4 with $\bar{z} = 0$ and $\bar{z} = 1$ for each pair of objectives. We can observe in the figures that for small $D$ values, M2 yields points that are at the same level of the leftmost point which corresponds to $\bar{z} = 0$. Additionally, the points of M2 and M4 coincide when $D = D^*$ and $\bar{z} = 1$, respectively, since they both impose the maximum *diversity-in-top-N* while maximizing accuracy. The level of distributional diversity M2 attains is very limited compared to M4 especially when $N$ is 5 and 10. At these values of $N$, M4 yields *diversity-in-top-N* levels close to its maximum possible value for the nonzero values of $\bar{z}$. In both Figure 6 and Figure 7, entropy-diversity and Gini-diversity, respectively, increase monotonically with $\bar{z}$. Therefore, we can state that $\bar{z}$ controls the distributional diversity as in the way $D$ and $A$ control *diversity-in-top-N* and accuracy, respectively.

Although $\bar{z}$ drives the levels of accuracy and the distributional diversity, the trade-off between these objectives can be further controlled by $m$. For example, if accuracy has greater importance than diversity in a recommender system, a small value of $m$ ensures that the accuracy level is affected minimally regardless of the value of $\bar{z}$, as shown in Figures 2, 3 and 4. The value of $m_i$ can be further scrutinized given its interpretation as the decrease in the total rating resulting from falling short of $\bar{z}$ for any item $i$. Another way to control the decrease in accuracy is adding a bounding constraint which enforces that the total rating maximized in M4 is no smaller than a prespecified percentage of the top-$N$ accuracy, which is obtained when $\bar{z} = 0$. Even though such an approach restrains solutions having total rating smaller than a tolerance limit, the bounding constraint destroys the total unimodularity of the constraint matrix and renders the solution of the model substantially harder than M4.

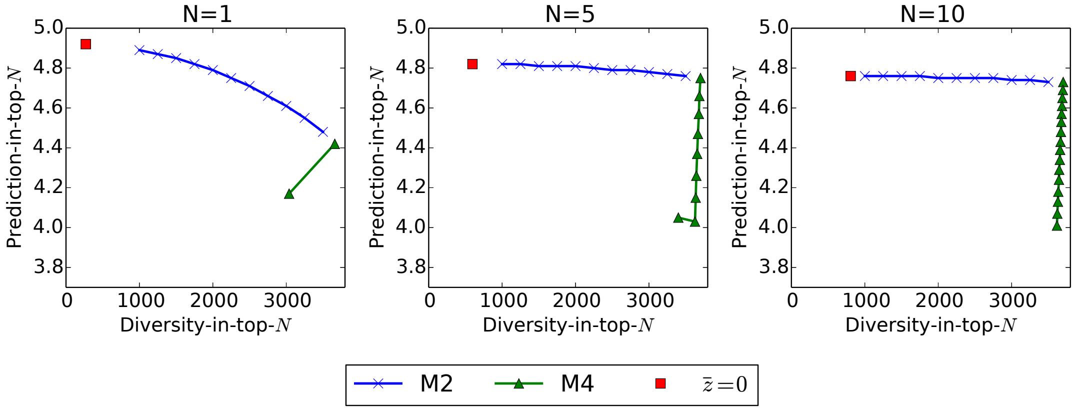

Figure 5: Prediction-in-top-$N$ vs. *Diversity-in-top-$N$* results of M2 and M4 on MovieLens data set for $1,000 \leq D \leq 3,750$ with increment of 250 and $0 \leq \bar{z} \leq \left\lceil \frac{T}{|I|} \right\rceil$, respectively.

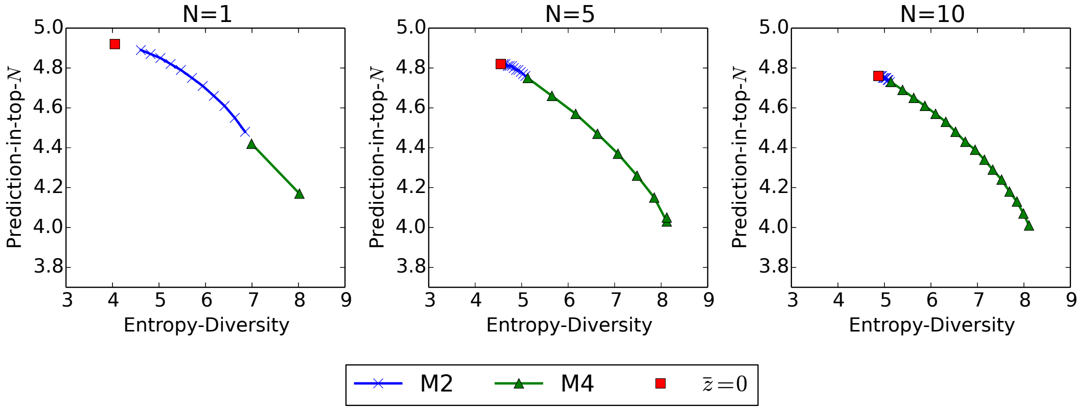

Figure 6: Prediction-in-top-$N$ vs. entropy-diversity results of M2 and M4 on MovieLens data set for $1,000 \leq D \leq 3,750$ with increment of 250 and $0 \leq \bar{z} \leq \left\lceil \frac{T}{|I|} \right\rceil$, respectively.

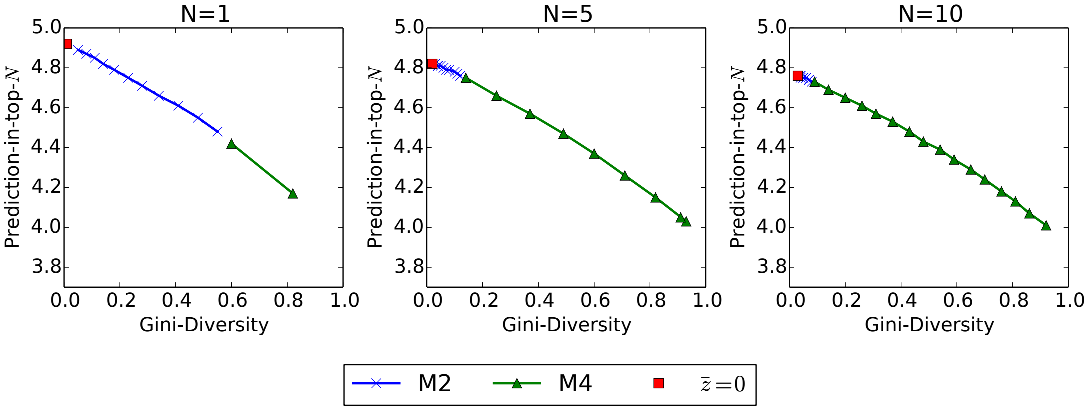

Figure 7: Prediction-in-top-$N$ vs. Gini-diversity results of M2 and M4 on MovieLens data set for $1,000 \leq D \leq 3,750$ with increment of 250 and $0 \leq \bar{z} \leq \left\lceil \frac{T}{|I|} \right\rceil$, respectively.

Figure 8 illustrates the distribution of $rec(i)$, $i \in I$ when M4 is solved for a selection of $\bar{z}$ values. These results are obtained by solving M4 with the MovieLens data set for $N = 10$. This figure is similar to Figure 1, which contains only the case $\bar{z} = 0$, except that in Figure 8, log transformation of $rec(i)$ is given in the vertical axis and the horizontal axis shows only a subset of the items for the sake of compactness. As $\bar{z}$ increases, we observe the following:

- the distribution of $rec(i)$ smooths out,
- the peak of the distribution descends,
- the tail of the distribution rises,
- the number of items that falls in the tail of the distribution increases.

These results demonstrate the impact of changing $\bar{z}$ on the graphical distribution of $rec(i)$, $i \in I$. A more even distribution is obtained with large values of $\bar{z}$, which supports our claim on $\bar{z}$ being the control parameter of the distributional diversity.

In Table 2, the solution times of the models which are obtained by CPLEX 12.5 are reported in seconds. These durations are averages over all values of $D$ and $\bar{z}$ for models M2 and M4, respectively. The last line gives the overall averages, and the solution times of M2 which can be solved as a linear programming problem appears to be comparable to those of M4 solved with $m = 1$. M4 is also a linear programming problem and is solved very efficiently for small values of $m$. However, the solution time of this model increases together with $\bar{z}$ and especially with $m$ as demonstrated in the table for its values used in the experiments. On the average, M2 is more efficiently solved than M4, which is more tightly constrained than the former for larger $m$ values.

Finally, we explain our results on two larger data sets, Amazon Books and Amazon Movies. The number of variables and constraints in M1, M2 and M4 for these data sets causes the solver to exceed the memory limit of the configuration of our computer. Therefore, we could not solve these instances

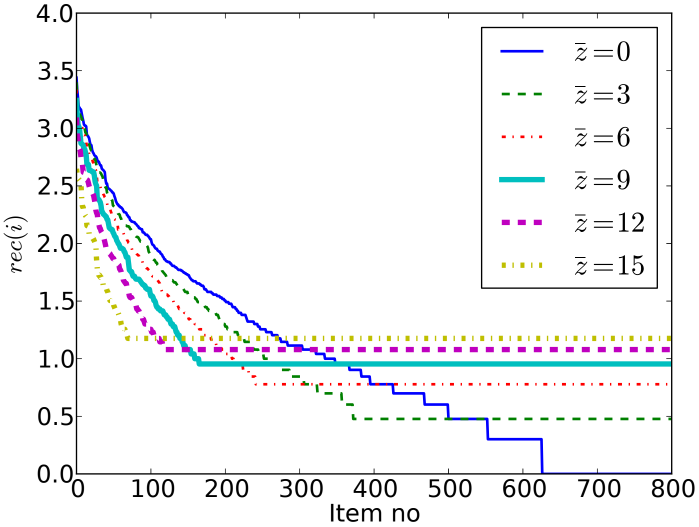

Figure 8: The effect of the parameter $\bar{z}$ on the distribution of items.

Table 2: Average solution times of the models (MovieLens) in seconds

| N | M2 | M4 (m=0.5) | M4 (m=1) | M4 (m=2) | M4 (m=5) |
|---|---|---|---|---|---|
| 1 | 51 | 13 | 41 | 105 | 110 |
| 5 | 55 | 16 | 58 | 134 | 143 |
| 10 | 54 | 16 | 61 | 119 | 114 |
| Average | 53 | 15 | 53 | 119 | 122 |

by handling the complete problem using a solver. Instead, the column generation method presented in Section 4.1 is employed, and the results are given in Table 3. We have discovered in our experiments presented above that the solutions generated by M2 lie between those generated by M4 with $\bar{z}=0$ and $\bar{z}=1$. Therefore, the column generation algorithm is applied only to our proposed model M4. Since the number of variables is independent of the parameters, its value is given on the top of Table 3 as “Totvar”. The solution time in seconds and the number of variables existing in the RMP at the optimal solution are reported under “time” and “col-var”, respectively. As discussed previously, the RMP is initialized with the top-$N$ items having the largest ratings for each user. Hence, the optimal solution of M4 for $\bar{z}=0$ is readily available, and its solution time is therefore omitted in Table 3. Moreover, the set of columns in the RMP at the optimal solution obtained for $\bar{z}-1$ is used to warm-start the RMP of M4 for $\bar{z}$. This is the reason that the number of columns is non-decreasing in $\bar{z}$ in Table 3. For each value of $N$, the solution time decreases until a threshold and then increases. The decreasing pattern can be attributed to the warm-starting of the RMP for $\bar{z}$ with the columns in the RMP for $\bar{z}-1$, while the increasing pattern after a threshold is due to the enlarged size of the RMP, which inflicts longer solution times at the iterations of the column generation algorithm. It

Table 3: Column generation results for the Amazon data sets

| | | Amazon Books | | Amazon Movies | |
|---|---|---|---|---|---|
| | | Totvar: 274,720,816 | | Totvar: 265,994,132 | |
| $N$ | $\bar{z}$ | col-var | time | col-var | time |
| 1 | 1 | 908,822 | 2,115 | 316,119 | 1,247 |
| | 2 | 1,017,435 | 837 | - | - |
| 5 | 1 | 795,100 | 1,225 | 594,351 | 924 |
| | 2 | 1,081,978 | 558 | 989,583 | 815 |
| | 3 | 1,320,125 | 545 | 1,229,254 | 688 |
| | 4 | 1,497,748 | 512 | 1,326,516 | 1,300 |
| | 5 | 1,658,687 | 589 | - | - |
| | 6 | 1,807,354 | 761 | - | - |
| | 7 | 1,927,457 | 1,307 | - | - |
| | 8 | 1,954,854 | 737 | - | - |
| 10 | 1 | 832,693 | 1,177 | 595,999 | 791 |
| | 2 | 1,117,200 | 596 | 909,657 | 711 |
| | 3 | 1,303,604 | 465 | 1,146,208 | 642 |
| | 4 | 1,471,047 | 465 | 1,374,863 | 592 |
| | 5 | 1,617,521 | 451 | 1,537,083 | 560 |
| | 6 | 1,777,617 | 467 | 1,662,868 | 563 |
| | 7 | 1,900,910 | 443 | 1,769,766 | 809 |
| | 8 | 2,037,086 | 508 | 1,829,162 | 996 |
| | 9 | 2,148,821 | 531 | - | - |
| | 10 | 2,304,628 | 698 | - | - |
| | 11 | 2,412,238 | 620 | - | - |
| | 12 | 2,497,919 | 704 | - | - |
| | 13 | 2,613,401 | 883 | - | - |
| | 14 | 2,693,430 | 998 | - | - |
| | 15 | 2,735,027 | 860 | - | - |

can be concluded that the warm-starting strategy together with adding multiple columns (a positive reduced cost column for each user) at each iteration lead to a superior performance in the tests of the method which generates only a fraction of all variables and terminates in a short time. When this method is applied to solve M4 (m=5) with the MovieLens data set, the average solution times are 25, 27 and 25 for $N = \{1, 5, 10\}$, respectively, which improves the average times obtained by solving M4 directly by solver.

In Figures 9 and 10, the accuracy and the aggregate diversity metrics obtained by solving M4 with Amazon Books and Amazon Movies data sets, respectively, are illustrated. The results are in line with those obtained from the MovieLens data set. For $N = 5$ and $N = 10$, the *diversity-in-top-N* levels attained by M4 are equal to the maximum possible value $|I|$ for almost all values of $\bar{z}$ larger than zero. While the distributional diversity metrics, namely Gini-diversity and entropy-diversity, increase monotonously with $\bar{z}$, the accuracy decreases. In contrast to Amazon Books data set, Amazon Movies data set contains more items than users. Therefore, $\bar{z}$ takes on smaller values in Amazon Movies than it takes in Amazon Books. However, as can be seen in the figures, the distributional diversity values reached with Amazon Movies are no smaller than those reached with Amazon Books since $rec(i)$ for $i \in I$ in both data sets approaches uniform distribution. Also note that for the Amazon data sets, the decrease in accuracy, as diversity increases, is less than that for the Movielens data set. Moreover, if we compare the Amazon data sets with each other, the decrease in accuracy in Amazon Books is less than that in Amazon Movies. The explanation of these differences in the accuracy levels lies in the number of users the data sets contain. Among these three data sets, as explained previously, Amazon Books contains the largest number of users whereas Movielens contains the smallest number of users. The larger the number of users is, the higher the likelihood that the model finds $\bar{z}$ users who have high predicted ratings for each item. In general, we can argue that for larger data sets, given that the user / item ratios are similar, our proposed model tends to achieve higher diversity values with smaller decrease in accuracy.

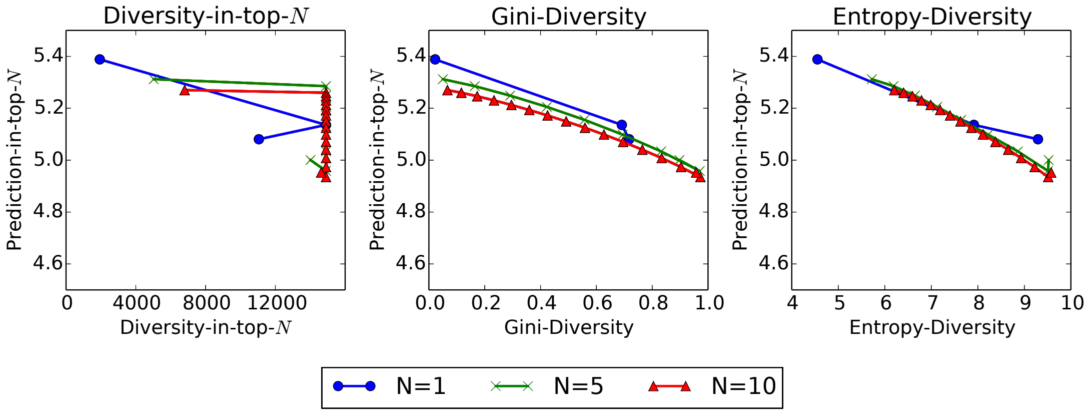

Figure 9: Prediction-in-top-$N$ vs. *Diversity-in-top-$N$* results of M2 and M4 on Amazon Books data set for $0 \leq \bar{z} \leq \left\lceil \frac{T}{|I|} \right\rceil$.

## 6. Conclusions and Future Work

Accuracy of recommendations is generally considered the single most important aspect of a recommender system. However, recently researchers have recognized the importance of other aspects of recommender systems which are also important such as the diversity of recommendations. In this paper, we have proposed a linear programming model which has been specifically designed for improving distributional diversity metrics. We have shown that the performance of our method in improving distributional diversity metrics is excellent. Such an increase in distributional diversity comes at the expense of decreased accuracy, which is controlled through model parameters. The multi-objective nature of the problem mandates that the desired level of these parameters is determined by solving the resulting model for various values of these parameters. We have provided guidelines on how to

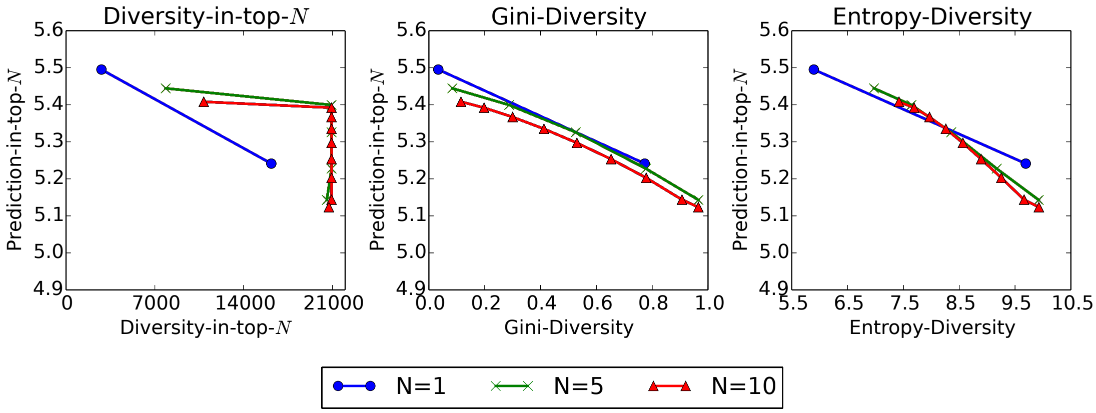

Figure 10: Prediction-in-top-$N$ vs. *Diversity-in-top-$N$* results of M2 and M4 on Amazon Movies data set for $0 \leq \bar{z} \leq \left\lceil \frac{T}{|I|} \right\rceil$.

select the appropriate values for these parameters, which will be different depending on the needs of the application and can be found with experimental methodologies such as A/B testing. We have also devised a column generation algorithm for our proposed model, and have managed to solve large-scale instances efficiently.

Although we apply our method for improving aggregate diversity in recommender systems, the proposed method is not specific to recommender systems domain. Our method can be applied in a multi-objective optimization framework where it is desired, as one of the objectives, that the distribution of a set of entities is close to uniform as much as possible. As a future work we plan to apply the proposed method in other problem domains.

## Acknowledgement

This work has been partially completed while the author was a member in the Faculty of Engineering at Bahçeşehir University. The research is supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under grant 113M480.

## References

Adomavicius, G., Y. Kwon. 2012. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Trans. Knowl. Data Eng.* **24** 896–911.

Adomavicius, G., Y. Kwon. 2014. Optimization-based approaches for maximizing aggregate recommendation diversity. *INFORMS Journal on Computing* **26** 351–369.

Adomavicius, G., A. Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **17** 734–749.

Aytekin, T., M. Ö. Karakaya. 2014. Clustering-based diversity improvement in top-n recommendation. *J. Intell. Inf. Syst.* **42** 1–18.

Bradley, K., B. Smyth. 2001. Improving recommendation diversity. *Proceedings of the 12th Irish Conference on Artificial Intelligence and Cognitive Science.*

Chankong, V., Y.Y. Haimes. 1983. *Multiobjective Decision Making Theory and Methodology*. Elsevier Science, New York.

Dantzig, G.B., P. Wolfe. 1960. Decomposition principle for linear programs. *Operations Research* **8** 101–111.

Desrosiers, C., G. Karypis. 2011. A comprehensive survey of neighborhood-based recommendation methods. Ricci et al. (2011), 107–144.

Ehrgott, M. 2005. *Multicriteria optimization*. Lecture Notes in Economics and Mathematical Systems, Springer-Verlag.

Ehrgott, M. 2006. A discussion of scalarization techniques for multiple objective integer programming. *Annals of Operations Research* **147** 343–360.

Ehrgott, M., D. M. Ryan. 2002. Constructing robust crew schedules with bicriteria optimization. *Journal of Multi-Criteria Decision Analysis* **11** 139–150.

Fisher, L.M. 1981. The lagrangean relaxation method for solving integer programming problems. *Management Science* **27** 1–18.

Fleder, D. M., K. Hosanagar. 2009. Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. *Management Science* **55** 697–712.

Gilmore, P. C., R. E. Gomory. 1961. A linear programming approach to the cutting-stock problem. *Operations Research* **9** 849–859.

Goffin, J.-L., J.-P. Vial. 2002. Convex nondifferentiable optimization: A survey focused on the analytic center cutting plane method. *Optimization Methods and Software* **17** 805–867.

Goldstein, D. G., D. C. Goldstein. 2006. Profiting from the long tail. *Harvard Business Review* **84** 24–28.

Held, M., R.M. Karp. 1970. The travelling salesman problem and minimum spanning trees:part i. *Operations Research* **18** 1138–1162.

Held, M., R.M. Karp. 1971. The travelling salesman problem and minimum spanning trees:part ii. *Mathematical Programming* **1** 6–25.

Held, M., P. Wolfe, H.P. Crowder. 1974. Validation of subgradient optimization. *Mathematical Programming* **6** 62–88.

Herlocker, J. L., J. A. Konstan, L. G. Terveen, J. Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* **22** 5–53.

Hu, Y., Y. Koren, C. Volinsky. 2008. Collaborative filtering for implicit feedback datasets. *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*. 263–272.

Hurley, N., M. Zhang. 2011. Novelty and diversity in top-n recommendation - analysis and evaluation. *ACM Trans. Internet Techn.* **10** 14.

Koren, Y., R. M. Bell. 2011. Advances in collaborative filtering. Ricci et al. (2011), 145–186.

Lemaréchal, C., J.J. Strodiot, A. Bihain. 1981. On a bundle algorithm for nonsmooth optimization. O.L. Mangasarian, R.R. Meyer, S.M. Robinson, eds., *Nonlinear Programming*, chap. 4. Academic Press, New York, 331–358.

Lokman, B., M. Köksalan. 2013. Finding all nondominated points of multi-objective integer programs. *Journal of Global Optimization* **57** 347–365.

Lops, P., M. de Gemmis, G. Semeraro. 2011. Content-based recommender systems: State of the art and trends. Ricci et al. (2011), 73–105.

Lübbecke, M. E., J. Desrosiers. 2005. Selected topics in column generation. *Operations Research* **53** 1007–1023.

McNee, S. M., J. Riedl, J. A. Konstan. 2006. Being accurate is not enough: how accuracy metrics have hurt recommender systems. G. M. Olson, R. Jeffries, eds., *CHI Extended Abstracts*. ACM, 1097–1101.

Özlen, M., M. Azizoğlu. 2009. Multi-objective integer programming: a general approach for generating all non-dominated solutions. *European Journal of Operational Research* **199** 25–35.

Ricci, F., L. Rokach, B. Shapira, P. B. Kantor, eds. 2011. *Recommender Systems Handbook*. Springer.

Smyth, B., P. McClave. 2001. Similarity vs. diversity. D. W. Aha, I. Watson, eds., *Proceedings of the 4th International Conference on Case-Based Reasoning, Vancouver, Canada*, *Lecture Notes in Computer Science*, vol. 2080. Springer, 347–361.

Sylva, J., A. Crema. 2004. A method for finding the set of non-dominated vectors for multiple objective integer linear programs. *European Journal of Operational Research* **158** 46–55.

Wu, L., S. Shah, S. Choi, M. Tiwari, C. Posse. 2014. The browsemaps: Collaborative filtering at linkedin. D. Jannach, J. Freyne, W. Geyer, I. Guy, A. Hotho, B. Mobasher, eds., *Proceedings of the 6th Workshop on Recommender Systems and the Social Web (RSWeb 2014)*, vol. 1271.

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