The Thirty-Second International Florida
Artificial Intelligence Research Society Conference (FLAIRS-32)

# Managing Popularity Bias in Recommender Systems with Personalized Re-Ranking

**Himan Abdollahpouri,[2] Robin Burke,[2] Bamshad Mobasher[3]**
[1,2]That Recommender Systems Lab, Department of Information Science, University of Colorado Boulder
[3] Web Intelligence Lab, DePaul University
[1]himan.abdollahpouri@colorado.edu, [2] robin.burke@colorado.edu, [3]mobasher@cs.depaul.edu

## Abstract

Many recommender systems suffer from popularity bias: popular items are recommended frequently while less popular, niche products, are recommended rarely or not at all. However, recommending the ignored products in the "long tail" is critical for businesses as they are less likely to be discovered. In this paper, we introduce a personalized diversification re-ranking approach to increase the representation of less popular items in recommendations while maintaining acceptable recommendation accuracy. Our approach is a post-processing step that can be applied to the output of any recommender system. We show that our approach is capable of managing popularity bias more effectively, compared with an existing method based on regularization. We also examine both new and existing metrics to measure the coverage of long-tail items in the recommendation.

## Introduction

Recommender systems have an important role in e-commerce and information sites, helping users find new items. One obstacle to the effectiveness of recommenders is the problem of popularity bias (Bellogín, Castells, and Cantador 2017): collaborative filtering recommenders typically emphasize popular items (those with more ratings) over other "long-tail" items (Park and Tuzhilin 2008) that may only be popular among small groups of users. Although popular items are often good recommendations, they are also likely to be well-known. So delivering only popular items will not enhance new item discovery and will ignore the interests of users with niche tastes. It also may be unfair to the producers of less popular or newer items since they are rated by fewer users.

Figure 1 illustrates the long-tail phenomenon in recommender systems. The $y$ axis represents the number of ratings per item and the $x$ axis shows the product rank. The first vertical line separates the top 20% of items by popularity – these items cumulatively have many more ratings than the 80% tail items to the right. These "short head" items are the very popular items, such as blockbuster movies in a movie recommender system, that garner much more viewer attention. Similar distributions can be found in other consumer domains.

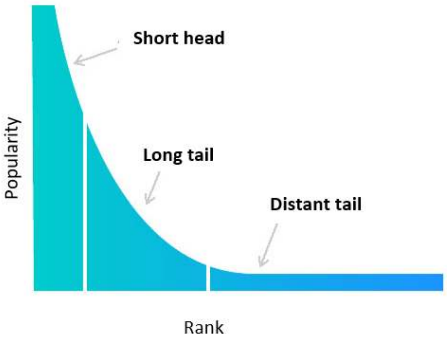

Figure 1: The long-tail of item popularity.

The second vertical line divides the tail of the distribution into two parts. We call the first part the *long tail*: these items are amenable to collaborative recommendation, even though many algorithms fail to include them in recommendation lists. The second part, the *distant tail*, are items that receive so few ratings that meaningful cross-user comparison of their ratings becomes unreliable. For these cold-start items, content-based and hybrid recommendation techniques must be employed. Our work in this paper is concerned with collaborative recommendation and therefore focuses on the long-tail segment.

We present a general and flexible approach for controlling the balance of item exposure in different portions of the item catalog as a post-processing phase for standard recommendation algorithms. Our work is inspired by (Santos, Macdonald, and Ounis 2010) where authors introduced a novel probabilistic framework called *xQuAD* for Web search result diversification which aims to generate search results that explicitly account for various aspects associated with an under-specified query. We adapt the xQuAD approach to the popularity bias problem. Our approach enables the system designer to tune the system to achieve the desired trade-off between accuracy and better coverage of long-tail, less popular items.

### Related Work

Recommending serendipitous items from the long tail is generally considered to be a key function of recommendation (Anderson 2006), as these are items that users are less likely to know about. Authors in (Brynjolfsson, Hu, and Smith 2006) showed that 30-40% of Amazon book sales are represented by titles that would not normally be found in brick-and-mortar stores.

Long-tail items are also important for generating a fuller understanding of users' preferences. Systems that use active learning to explore each user's profile will typically need to present more long tail items because these are the ones that the user is less likely to know about, and where user's preferences are more likely to be diverse (Resnick et al. 2013).

Finally, long-tail recommendation can also be understood as a social good. A market that suffers from popularity bias will lack opportunities to discover more obscure products and will be, by definition, dominated by a few large brands or well-known artists (Celma and Cano 2008). Such a market will be more homogeneous and offer fewer opportunities for innovation and creativity.

The idea of the long-tail of item popularity and its impact on recommendation quality has been explored by some researchers (Brynjolfsson, Hu, and Smith 2006; Park and Tuzhilin 2008). In those works, authors tried to improve the performance of the recommender system in terms of accuracy and precision, given the long-tail in the ratings. Our work, instead, focuses on reducing popularity bias and balancing the representation of items across the popularity distribution.

A regularization-based approach to improving long tail recommendations is found in (Abdollahpouri, Burke, and Mobasher 2017a). One limitation with that work is that this work is restricted to factorization models where the long-tail preference can be encoded in terms of the latent factors. In contrast, a re-ranking approach can be applied to the output of any algorithm. Another limitation of that work is that it does not account for user tolerance towards long-tail items: the fact that there may be some users only interested in popular items. In our model, we take personalization of long-tail promotion into account as well.

And finally, there is substantial research in recommendation diversity, where the goal is to avoid recommending too many similar items (Zhou et al. 2010; Castells, Vargas, and Wang 2011; Zhang and Hurley 2008). Personalized diversity is also another related area of research where the amount of diversification is dependent on the user's tolerance for diversity (Eskandanian, Mobasher, and Burke 2017; Wasilewski and Hurley 2018). Another similar work to ours is (Vargas, Castells, and Vallet 2012) where authors used a modified version of xQuAD called relevance based xQuAD for intent-oriented diversification of search results and recommendations. Another work used a similar approach but for fairness-aware recommendation (Liu and Burke 2018) where xQuAD was used to make a fair representation of items from different item providers. Our work is different from all these previous diversification approaches in that it is not dependent on the characteristics of items, but rather on the relative popularity of items.

# Controlling Popularity Bias

## xQuAD

Result diversification has been studied in the context of information retrieval, especially for web search engines, which have a similar goal to find a ranking of documents that together provide a complete coverage of the aspects underlying a query (Santos et al. 2015). EXplicit Query Aspect Diversification (xQuAD) (Santos, Macdonald, and Ounis 2010) explicitly accounts for the various aspects associated with an under-specified query. Items are selected iteratively by estimating how well a given document satisfies an uncovered aspect.

In adapting this approach, we seek to recognize the difference among users in their interest in long-tail items. Uniformly-increasing diversity of items with different popularity levels in the recommendation lists may work poorly for some users. We propose a variant that adds a personalized bonus to the items that belong to the under-represented group (i.e. the long-tail items). The personalization factor is determined based on each user's historical interest in long-tail items.

# Methodology

We build on the *xQuAD* model to control popularity bias in recommendation algorithms. We assume that for a given user $u$, a ranked recommendation list $R$ has already been generated by a base recommendation algorithm. The task of the modified xQuAD method is to produce a new re-ranked list $S$ ($|S| < |R|$) that manages popularity bias while still being accurate.

The new list is built iteratively according to the following criterion:

$$P(v|u) + \lambda P(v, S'|u) \tag{1}$$

where $P(v|u)$ is the likelihood of user $u \in U$ being interested in item $v \in V$, independent of the items on the list so far as, predicted by the base recommender. The second term $P(v, S'|u)$ denotes the likelihood of user u being interested in an item $v$ as an item not in the currently generated list $S$.

Intuitively, the first term incorporates ranking accuracy while the second term promotes diversity between two different categories of items (i.e. short head and long tail). The parameter $\lambda$ controls how strongly controlling popularity bias is weighted in general. The item that scores most highly under the equation 1 is added to the output list $S$ and the process is repeated until $S$ has achieved the desired length.

To achieve more diverse recommendation containing items from both short head and long tail items, the marginal likelihood $P(v, S'|u)$ over both item categories long-tail head (Γ) and short head (Γ') is computed by:

$$P(v, S'|u) = \sum_{d \in \{\Gamma, \Gamma'\}} P(v, S'|d)P(d|u) \tag{2}$$

Following the approach of (Santos, Macdonald, and Ounis 2010), we assume that the remaining items are independent of the current contents of $S$ and that the items are independent of each other given the short head and long tail categories. Under these assumptions, we can compute $P(v, S'|d)$ in Eq.2 as

$$P(v, S'|d) = P(v|d)P(S'|d) = P(v|d)\prod_{i\in S}(1 - P(i|d, S)) \tag{3}$$

By substituting equation 3 into equation 2, we can obtain

$$P(v, S'|u) = \sum_{d\in\{\Gamma,\Gamma'\}} P(d|u)P(v|d)\prod_{i\in S}(1 - P(i|d, S)) \tag{4}$$

where $P(v|d)$ is equal to 1 if $v \in d$ and 0 otherwise.

We measure $P(i|d, S)$ in two different ways to produce two different algorithms. The first way is to use the same function as $P(v|d)$, an indicator function where it equals to 1 when item $i$ in list $S$ already covers category $d$ and 0 otherwise. We call this method *Binary xQuAD* and it is how original xQuAD was introduced. Another method that we present in this paper is to find the ratio of items in list $S$ that covers category $d$. We call this method *Smooth xQuAD*.

The likelihood $P(d|u)$ is the measure of user preference over different item categories. In other words, it measures how much each user is interested in short head items versus long tail items. We calculate this likelihood by the ratio of items in the user profile which belong to category $d$.

In order to select the next item to add to $S$, we compute a re-ranking score for each item in $R\backslash S$ according to Eq. 4. For an item $v' \in d$, if $S$ does not cover $d$, then an additional positive term will be added to the estimated user preference $P(v'|u)$. Therefore, the chance that it will be selected is larger, balancing accuracy and popularity bias.

In Binary xQuAD, the product term $\prod_{i\in S}(1 - P(i|d, S))$ is only equal to 1 if the current items in $S$ have not covered the category $d$ yet. Binary xQuAD is, therefore, optimizing for a *minimal* re-ranking of the original list by including the best long-tail item it can, but not seeking diversity beyond that.

## Experiment

In this section, we test our proposed algorithm on two public datasets. The first is the well-known Movielens 1M dataset that contains 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 MovieLens users (Harper and Konstan 2015). The second dataset is the Epinions dataset, which is gathered from a consumers opinion site where users can review items (Massa and Avesani 2007). This dataset has the total number of 664,824 ratings given by 40,163 users to 139,736 items. In Movielens, each user has a minimum of 20 ratings but in Epinions, there are many users with only a single rated item.

Following the data reduction procedure in (Abdollahpouri, Burke, and Mobasher 2017a), we removed users who had fewer than 20 ratings from the Epinion dataset, as users with longer profiles are much more likely to have long-tail items in their profiles. MovieLens dataset already consists of only users with more than 20 ratings. The retained users were those likely to have rated enough long-tail items so that our objective could be evaluated in a train / test scenario. We also removed distant long-tail items from each dataset using a limit of 20 ratings, a number 20 is chosen to be consistent with the cut-off for users.

After filtering, the MovieLens dataset has 6,040 users who rated 3043 movies with a total number of 995,492 ratings, a reduction of about 0.4%. Applying the same criteria to the Epinions dataset decreases the data to 220,117 ratings given by 8,144 users to 5,195 items, a reduction of around 66%. We split the items in both datasets into two categories: long-tail (Γ) and short head (Γ')in a way that short head items correspond to %80 of the ratings while long-tail items have the rest of the %20 of the ratings. We plan to consider other divisions of the popularity distribution in future work. For MovieLens, the short-head items were those with more than 506 ratings. In Epinions, a short-head item needed only to have more than 73 ratings.

## Evaluation

The experiments compare four algorithms. Since we are concerned with ranking performance, we chose as our baseline algorithm RankALS, a pair-wise learning-to-rank algorithm. We also include the regularized long-tail diversification algorithm in (Abdollahpouri, Burke, and Mobasher 2017a) (indicated as *LT-Reg* in the figures.) We used the output from RankALS as input for the two re-ranking variants described above: Binary xQuAD and Smooth xQuAD, marked *Binary* and *Smooth* in the figures. We compute lists of length 100 from RankALS and pass these to the re-ranking algorithms to compute the final list of 10 recommendations for each user. We used the implementation of RankALS in LibRec 2.0[1] for all experiments.

In order to evaluate the effectiveness of algorithms in mitigating popularity bias we use four different metrics:

**Average Recommendation Popularity (ARP)**: This measure from (Yin et al. 2012) calculates the average popularity of the recommended items in each list. For any given recommended item in the list, we measure the average number of ratings for those items. More formally:

$$ARP = \frac{1}{|U_t|}\sum_{u\in U_t}\frac{\sum_{i\in L_u}\phi(i)}{|L_u|} \tag{5}$$

where $\phi(i)$ is the number of times item $i$ has been rated in the training set. $L_u$ is the recommended list of items for user $u$ and $|U_t|$ is the number of users in the test set.

**Average Percentage of Long Tail Items (APLT)**: As used in (Abdollahpouri, Burke, and Mobasher 2017a), this metric measures the average percentage of long tail items in the recommended lists and it is defined as follows:

$$APLT = \frac{1}{|U_t|}\sum_{u\in U_t}\frac{|\{i, i \in (L_u \cap \Gamma)\}|}{|L_u|} \tag{6}$$

This measure gives us the average percentage of items in users' recommendation lists that belong to the long tail set.

**Average Coverage of Long Tail items (ACLT)**: We introduce another metric to evaluate how much exposure long-tail items get in the entire recommendation. One problem

[1] www.librec.net

with $APLT$ is that it could be high even if all users get the same set of long tail items. $ACLT$ measures what fraction of the long-tail items the recommender has covered:

$$ACLT = \frac{1}{|U_t|} \sum_{u \in U_t} \sum_{i \in L_u} \mathbb{1}(i \in \Gamma) \quad (7)$$

where $\mathbb{1}(i \in \Gamma)$ is an indicator function and it equals to 1 when i is in $\Gamma$. This function is related to the *Aggregate Diversity* metric of (Adomavicius and Kwon 2012) but it looks only at the long-tail part of the item catalog.

In addition to the aforementioned long tail diversity metrics, we also evaluate the accuracy of the ranking algorithms in order to examine the diversity-accuracy trade-offs. For this purpose we use the standard *Normalized Discounted cumulative Gain* (*NDCG*) measure of ranking accuracy.

## Results

Figure 2 shows the results for the Epinions dataset across the different algorithms using a range of values for $\lambda$. (Note that the LT-Reg algorithm uses the parameter $\lambda$ to control the weight placed on the long-tail regularization term.) All results are averages from five-fold cross-validation using a $\%80$ -$\%20$ split for train and test, respectively. As expected, the diversity scores improve for all algorithms, with some loss of ranking accuracy. Differences between the algorithms are evident, however. The exposure metric (ACLT) plot shows that the two re-ranking algorithms, and especially the Smooth version, are doing a much better job of exposing items across the long-tail inventory than the regularization method. The ranking accuracy shows that, as expected, the Binary version does slightly better as it performs minimal adjustment to the ranked lists. LT-Reg is not as effective at promoting long-tail items, either by the list-wise APLT measure or by the catalog-wise ACLT.

Another view of the same results is provided in Figure 3. Here we look at the long-tail diversity metrics relative to NDCG loss, which clarifies the patterns seen in Figure 2. We see that the Binary and Smooth algorithms are fairly similar in terms of diversity-accuracy trade-off, while LT-Reg has a distinctly lower and flatter improvement curve with increased loss of ranking accuracy. ARP metric is the only one where the algorithms are fairly similar, especially at lower values of NDCG loss.

The MovieLens dataset shows different relative performance across the algorithms as seen in Figure 4. The Smooth re-ranking method shows a more distinct benefit and LT-Reg is somewhat more effective. This finding is confirmed in the relative results shows in Figure 5, which also shows the algorithms having quite similar values for the ARP metric, in spite of the differences on the other metrics.

Comparing the two datasets, we see that long-tail diversification is more of a challenge in the sparser Epinions dataset. With 10% NDCG loss, it is possible to bring exposure to around 15% of the long-tail catalog in Epinions; whereas for MovieLens, 0.2% loss yields an equivalent or greater benefit. LT-Reg is much less effective. (In both datasets, the baseline value is very close to zero.) The average number of long-tail items in each recommendation list shows a similar pattern.

In the sparser dataset, the Binary and Smooth measures are similar in performance, but differences appear in MovieLens, where the Smooth algorithm shows stronger improvement in the ACLT measure, particularly. This effect is most likely due to the fact that in the sparser data, it is more difficult to find a single long-tail item to promote into a recommendation list, with greater accuracy cost in doing so. In MovieLens, these higher-quality items appear more often and the Smooth objective values the promotion of multiple such items into the recommendation lists.

Another conclusion we can draw is that the ARP measure is not a good measure of long-tail diversity when it is used only on its own. It has the benefit of not requiring the experimenter to set a threshold distinguishing long-tail and short-head items. However, as we see here, algorithms can have very similar ARP performance and be quite different in terms of their ability to cover the long-tail catalog and to promote long-tail items to users. So it is important to look at all these metrics together.

## Conclusion and future work

Adequate coverage of long-tail items is an important factor in the practical success of business-to-consumer organizations and information providers. Since short-head items are likely to be well known to many users, the ability to recommend items outside of this band of popularity will determine if a recommender system can introduce users to new products and experiences. Yet, it is well-known that recommendation algorithms have biases towards popular items.

In this paper, we presented re-ranking approaches for long-tail promotion and compared them to a state-of-the-art model-based approach. On two datasets, we were able to show that the re-ranking methods boost long-tail items while keeping the accuracy loss small, compared to the model-based technique. We also showed that the average recommendation popularity (ARP) measure from (Yin et al. 2012) is not a good metric on its own for evaluating long-tail promotion, as algorithms might have similar ARP performance but quite different performance on other measures of popularity bias. So it is better to use it along with other metrics such as APLT and ACLT to get the right picture of the effectiveness of the algorithms.

One interesting area for future work would be using this model for multistakeholder recommendation where the system needs to make recommendations in the presence of different stakeholders providing the products (Abdollahpouri, Burke, and Mobasher 2017b; Burke and Abdollahpouri 2016; Burke et al. 2016). In those cases, another parameter could be used to control the priority of each stakeholder in the system.

## References

Abdollahpouri, H.; Burke, R.; and Mobasher, B. 2017a. Controlling popularity bias in learning-to-rank recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, 42–46. ACM.

Abdollahpouri, H.; Burke, R.; and Mobasher, B. 2017b. Recommender systems as multi-stakeholder environments.

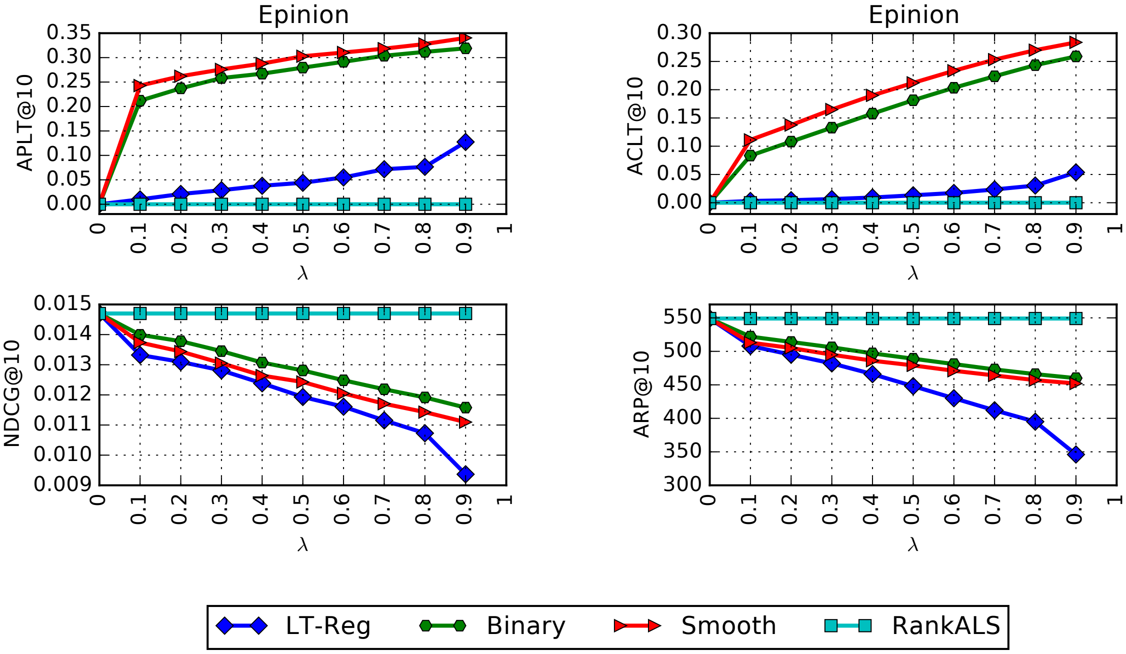

Figure 2: Results for the Epinions dataset

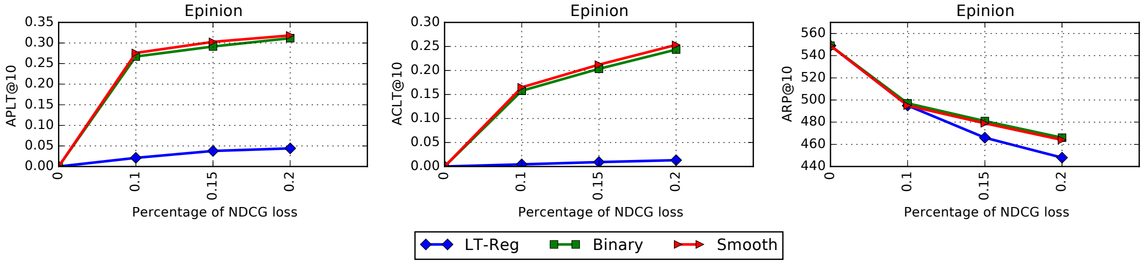

Figure 3: Comparison of popularity bias control for different algorithms at different levels of NDCG loss (Epinions)

In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization (UMAP2017)*. ACM.

Adomavicius, G., and Kwon, Y. 2012. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Transactions on Knowledge and Data Engineering* 24(5):896–911.

Anderson, C. 2006. *The long tail: Why the future of business is selling more for less*. Hyperion.

Bellogín, A.; Castells, P.; and Cantador, I. 2017. Statistical biases in information retrieval metrics for recommender systems. *Information Retrieval Journal* 20(6):606–634.

Brynjolfsson, E.; Hu, Y. J.; and Smith, M. D. 2006. From niches to riches: Anatomy of the long tail. *Sloan Management Review* 67–71.

Burke, R., and Abdollahpouri, H. 2016. Educational Recommendation with Multiple Stakeholders. In *Third International Workshop on Educational Recommender Systems*.

Burke, R. D.; Abdollahpouri, H.; Mobasher, B.; and Gupta, T. 2016. Towards multi-stakeholder utility evaluation of recommender systems. In *Workshop on Surprise, Opposition, and Obstruction in Adaptive and Personalized Systems, UMAP 2016*.

Castells, P.; Vargas, S.; and Wang, J. 2011. Novelty and diversity metrics for recommender systems: choice, discovery and relevance. In *Proceedings of International Workshop on Diversity in Document Retrieval (DDR)*, 29–37. ACM Press.

Celma, Ò., and Cano, P. 2008. From hits to niches?: or how popular artists can bias music recommendation and discovery. In *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*, 5. ACM.

Eskandanian, F.; Mobasher, B.; and Burke, R. 2017. A clustering approach for personalizing diversity in collaborative recommender systems. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, 280–284. ACM.

Harper, F. M., and Konstan, J. A. 2015. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5(4):19.

Liu, W., and Burke, R. 2018. Personalizing fairness-aware

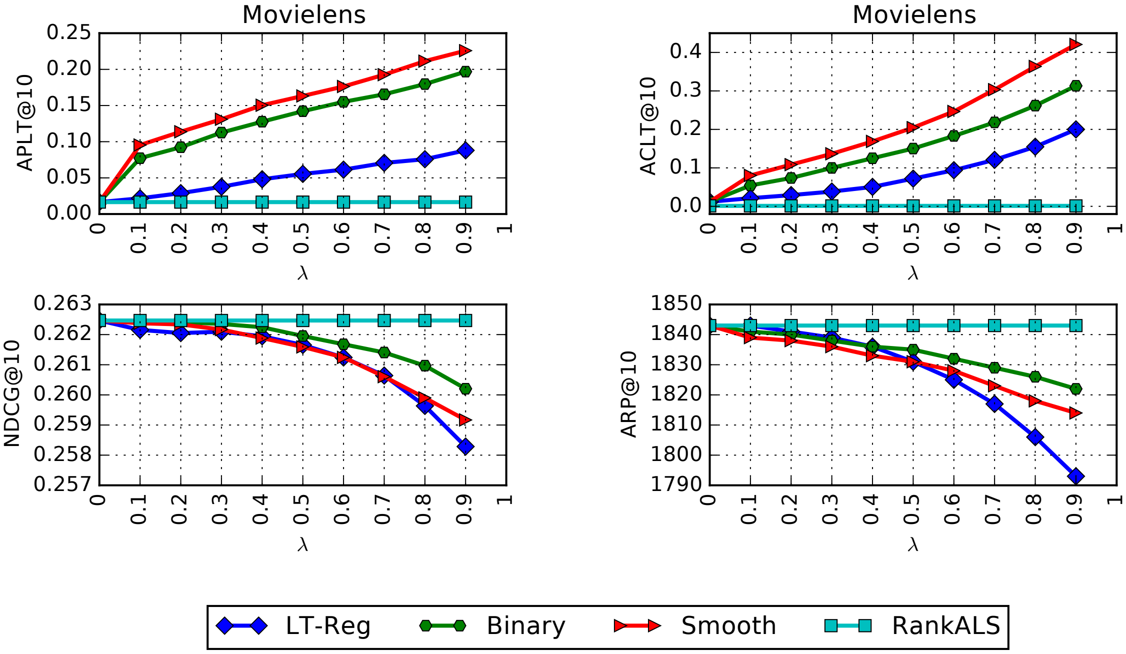

Figure 4: Results for the MovieLens dataset

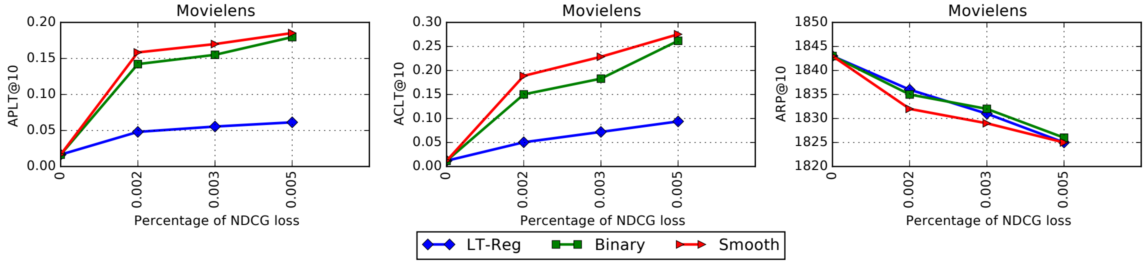

Figure 5: Comparison of popularity bias control for different algorithms at different levels of NDCG loss (MovieLens)

re-ranking. *arXiv preprint arXiv:1809.02921*.

Massa, P., and Avesani, P. 2007. Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems*, 17–24. ACM.

Park, Y.-J., and Tuzhilin, A. 2008. The long tail of recommender systems and how to leverage it. In *Proceedings of the 2008 ACM conference on Recommender systems*, 11–18. ACM.

Resnick, P.; Garrett, R. K.; Kriplean, T.; Munson, S. A.; and Stroud, N. J. 2013. Bursting your (filter) bubble: strategies for promoting diverse exposure. In *Proceedings of the 2013 conference on Computer supported cooperative work companion*, 95–100. ACM.

Santos, R. L.; Macdonald, C.; Ounis, I.; et al. 2015. Search result diversification. *Foundations and Trends® in Information Retrieval* 9(1):1–90.

Santos, R. L.; Macdonald, C.; and Ounis, I. 2010. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th international conference on World wide web*, 881–890. ACM.

Vargas, S.; Castells, P.; and Vallet, D. 2012. Explicit relevance models in intent-oriented information retrieval diversification. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, 75–84. ACM.

Wasilewski, J., and Hurley, N. 2018. Intent-aware item-based collaborative filtering for personalised diversification. In *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*, 81–89. ACM.

Yin, H.; Cui, B.; Li, J.; Yao, J.; and Chen, C. 2012. Challenging the long tail recommendation. *Proceedings of the VLDB Endowment* 5(9):896–907.

Zhang, M., and Hurley, N. 2008. Avoiding monotony: improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM conference on Recommender systems*, 123–130. ACM.

Zhou, T.; Kuscsik, Z.; Liu, J.-G.; Medo, M.; Wakeling, J. R.; and Zhang, Y.-C. 2010. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences* 107(10):4511–4515.