RECOMMENDER RESPONSE TO USER PROFILE DIVERSITY AND POPULARITY BIAS

by

Sushma Channamsetty, B.E

A thesis submitted to the Graduate Council of
Texas State University in partial fulfillment
of the requirements for the degree of
Master of Science
with a Major in Computer Science
August 2016

Committee Members:

Michael Ekstrand

Vangelis Metsis

Anne Hee Hiong Ngu

FAIR USE AND AUTHOR'S PERMISSION STATEMENT

Fair Use

Duplication Permission

ACKNOWLEDGEMENTS

I wish to express my gratitude to Dr Michael Ekstrand for supporting me and guiding me throughout the research. I would also like to appreciate Tajinder Pal Singh and Mohammed Imran Rukmoddin Kazi for demonstrating and explaining me about the Lenskit toolkit that I have used in my research. In addition, I would like to thank to the whole research team for the discussions in the research meetings.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

ABSTRACT

Recommender systems are commonly evaluated to understand the effectiveness of their algorithms. Diversity and novelty of the recommender systems have been in consideration while evaluating the systems in addition to accuracy and prediction metrics in order to provide better recommendations. Different evaluation metrics that are related to diversity and novelty have been discussed in some of the previous works. This work provides a comprehensive study and analysis of the recommender algorithms and its relationship to the user's bias in terms of popularity and diversity. This kind of analysis helps us to understand if the core algorithms personalize the recommendations based on the users' bias. We performed offline experiments using the MovieLens data and analyzed the correlation between the user profile and the recommender profile for both diversity and popularity bias using different metrics. Finally, we report the analysis observations and study how it complements the previous work done.

1. INTRODUCTION

The recommender systems are most popularly known for assisting users in selecting or purchasing items of interest in various domains like music, movies, books, recipes and many other items. They are frequently embedded in online services to provide personalized recommendations based on user's interests and expose the user to a large collection of items. In recent years, recommender systems are popular in commercial business as they play a significant role in companies' sales margin (Resnick and Varian 1997). Companies like Amazon, Netflix, Spotify and many more use recommender systems to recommend products to users based on their interests and preferences. The research community in recommender system field has been focusing a lot in finding new algorithms and to evaluate them in order to provide better recommendations to the user.

The evaluation of recommender system has been an important field of study from the beginning and is still an ongoing topic of research. Initially, the evaluation is done mostly using a prediction accuracy metric to determine the system's ability to accurately predict user's choices (Herlocker et al. 2004; Breese, Heckerman, and Kadie 1998). Evaluations taken into consideration are three types of experiments namely offline experiments, user study (Pu, Chen, and Hu 2011) and online experiments (Shani and Gunawardana 2011). Offline experiments are used to estimate the prediction accuracy of the recommender from an existing data, user study is generally conducted to gain insights about the user's experience of using the recommender system and the online experiments are conducted on a deployed system.

In recent years, studies have shown that even though the recommendations that are most accurate according to the error metrics might be of no use to the user. Tuning the recommender to produce most accurate recommendations might restrict the user from having useful recommendations (McNee, Riedl, and Konstan 2006; Bradley and Smyth 2001). Some of the non-accuracy measurements that have been proposed for recommender systems are as follows:

- Diversity (M. Zhang and Hurley 2008)
- Novelty (M. Zhang and Hurley 2008; Hurley and Zhang 2011)
- Temporal diversity (Lathia et al. 2010)
- Temporal Stability
- Attack resistance (Resnick and Sami 2007)

Clearly, there has been an increase in attention to other dimensions like diversity and novelty beyond accuracy in evaluation of recommendation systems but previous work has focused primarily on aggregate recommender behavior. In our work, we are trying to analyze the behavior of individual users in relation to the usefulness of recommendations. We are also focusing on the impact of user's profile characteristics on the recommender's output profile. The general idea of this thesis is to characterize the performance and response of the recommender profile as the users' input change. In particular, we are interested in the study of attributes such as diversity and popularity. This kind of analysis helps to see and understand the impact of the users' profile on the recommender system for different algorithms. In other words, we consider the recommender as the function of the users' input. We used the following five different recommender algorithms to do this analysis:

- Item-Item collaborative filtering
- User-User collaborative filtering
- Funk-SVD
- Content Based Filtering
- Popularity

In this work, we seek to answer the following research questions,

- Does the users' input profile change the recommender response profile?
- Do different recommender algorithms propagate the change in users' input profile differently?
- How does the accuracy of the recommender correlate with diversity or popularity bias of the user?

The remainder of this report is structured as follows:

- Chapter 2 gives the literature survey related to our work and presents the background information that is required to follow the rest of this document. It mainly deals with the basics of recommender systems and the evaluation techniques that have been in use. It also shows the importance of interests like diversity and popularity in the recommendation lists.
- Chapter 3 illustrates about the methodology we used to answer our research questions. In this chapter, we also describe about the tools we used in our research.
- Chapter 4 presents the results we obtained in our research. We included the graphs that help us answered the research questions and explain them in detail.

- Chapter 5 discusses our results in a high level. We also discuss how our results complement other peoples’ work.
- Chapter 6 gives the conclusions and future work where we summarize our work and talk about what we did and how successful we were to obtain the answers of our research questions.

2. BACKGROUND AND RELATED WORK

This chapter describes the background information and existing work in this area.

2.1 RECOMMENDER SYSTEMS

Recommender systems are systems that generate a list of unseen items that are predicted to be the most suitable to the users based on their personal profile. In (Resnick and Varian 1997), different dimensions are identified for the technical design of recommender systems. These dimensions are domain, purpose, recommendation context, neighborhood, personalization level, privacy and trustworthiness, interfaces and recommendation algorithms.

2.2 ALGORITHMS

This section describes some of the algorithms that we used in our research:

- User-User Collaborative (UUC) filtering: This algorithm makes recommendations based on opinion of other people. It finds a set of customers who are interested in the items that overlap with the items that the user's rated. It aggregates items from these similar customers (also called neighborhood users), eliminates the items that are already consumed by the users and recommends the remaining items (Resnick et al. 1994).
- Item-Item Collaborative (IIC) filtering: This algorithm, instead of looking for similar users, it finds similar items to the items that are purchased or rated by the users. Most similar items that are found are recommended to the users (Linden, Smith, and York 2003).
- Funk SVD: SVD (Singular Value Decomposition) is matrix decomposition of the ratings matrix into three matrices.

$$M = U\Sigma\mathrm{VT}$$

M is the ratings matrix, U and V are the eigenvectors of MM^T and M^TM respectively, Σ is the diagonal matrix of singular values. Singular values can be considered as the dimensions or the features of the items that the users prefer. Hence, most important singular values are considered and the rest are truncated. Funk SVD deals with the problems of SVD like being slow and also deals with the missing data. It initializes matrix to some arbitrary value and it checks for convergence for every latent feature. It actually trains every feature till it reaches convergence.

- Content-Based filtering (CBF): Content based filtering technique models items to attributes. It's based on the vector of keywords or features. It tries to recommend similar items that the user has rated in the past based on content description of the items (Kamba, Bharat, and Albers 1995).
- Popularity: Popularity algorithms recommends the most popular items to the users. It eliminates the items that are already consumed by the users and recommends the remaining popular items.

2.3 RELATED WORK

In recent times, researchers have acknowledged that just considering the prediction accuracy would not make a good recommender. For example, in (McNee, Riedl, and Konstan 2006), the authors show how accuracy metrics can actually hurt the recommender systems. They give an example of a travel recommender where the recommender is penalized for recommending new places and ended up recommending the places that the user has already visited. They claim that there are other aspects like similarity of recommendation lists, recommendation serendipity, and the importance of user needs and expectations which the accuracy metrics does not measure. This has drawn interest in evaluation of the system in terms of new characteristics like diversity, novelty, serendipity and many more depending on the domain of the recommender system.

User centric evaluation of recommenders have grown popular as users' experience is the most important aspect when deciding the fate of the recommender system. In (Pu, Chen, and Hu 2011), the authors conducted a large user survey to measure some of the aspects from the users' experience like users' satisfaction with the system, system's usability, usefulness which means the intention to purchase a product and return to the system. Commercial businesses conduct user evaluations like users' click rate and users' browsing patterns to understand their preferences and interests. (Knijnenburg et al. 2012; Xiao and Benbasat 2007; Ozok, Fan, and Norcio 2010) proposed different models that support hypothesis relating to users' perception and correlation with their choice satisfaction.

Recent focus has also shifted to the recommendation lists as a whole instead of focusing on the quality of each recommended item. This deals with the concern of 'pigeonholing' the users. Hence, there have been research done in interests like diversity, novelty and serendipity of the recommendation lists (Y. C. Zhang et al. 2012; Yu, Lakshmanan, and Amer-Yahia 2009; M. Zhang and Hurley 2008). There has also been some research done to evaluate these kinds of interests in the recommendation list. In (Vargas and Castells 2011), some of the hybrid approaches were proposed to understand the non-performance recommendation characteristics like diversity, novelty and coverage.

Our work is related to the understanding of the diversity and novelty that some of the recommender algorithms can produce and we try to evaluate them using the user profile. Some studies have been done for understanding the user perceptions and expectations at the algorithmic level. In (Konstan and Riedl 2012), the authors concentrated on different algorithms and their hybrid forms to understand the user experience with each recommender. In (Ekstrand et al. 2014), a user study was done to evaluate the common collaborative filtering algorithms in dimensions like novelty, diversity, accuracy, satisfaction and degree of personalization based on their experience with each of the recommender. Our work mainly deals with the recommender's output profile changes with the users' profile for five different recommender algorithms and see how accuracy correlate with the diversity and popularity bias of the users' input.

3. METHODOLOGY

This chapter mainly deals with methodology we adopted to answer our research questions.

3.1 DATA

We are using a stable benchmark dataset which is MovieLens 10M dataset (Harper and Konstan 2015). It consists of approximately 10M ratings and 100,000 tag applications applied to 10,000 movies by 72,000 users[1]. The dataset was collected by GroupLens research team through an online recommendation service called MovieLens[2]. The users included in the dataset have rated at least 20 movies and each user is represented by a unique Id. It consists of three files called movie.dat, tags.dat and ratings.dat.

We also use an additional dataset called MovieLens Tag Genome dataset (Vig, Sen, and Riedl 2009). This dataset consists of approximately 11 million computed tag-movie relevance scores from a pool of 1,100 tags applied to 10,000 movies. This dataset is used to find the similarity between the movies in our analysis. This dataset consists of three files called tag-relevance.dat, movies.dat, tags.dat. The tag relevance scores in the file tag-relevance.dat represents the relevance of the tag to the movie on a scale of 0-1. We use this information to calculate how similar a movie is from one another.

3.2 LENSKIT EXPERIMENT

We conducted an experiment using LensKit Recommender toolkit (Ekstrand et al. 2011) on the MovieLens 10M data set. We are using the LensKit evaluator which is used

---

[1] The sparsity of the MovieLens 10M data is 76.4%.

[2] https://movielens.org

for conducting offline evaluations of recommenders and it uses the train-test evaluation method to find the accuracy of the predictions done by the recommenders. The different settings we used in our experiment are described as follows:

- Partition count is 5 and hence, 5-fold cross validation in LensKit is used to evaluate the recommendations given by each algorithm.
- LensKit supports three cross folding methods: partition by ratings, partition by users and partition by users with a sample. In this experiment, we used partition by users method. This method splits the data into five partitions and it chooses any four partitions as train data. In the remaining one partition, random one-fifth of each users' ratings are considered as the test data and the rest four-fifth is combined with the train data.
- No candidate set restriction is set in the experiment. Hence, we generated the recommendation list of size 100. We then filtered the lists up to rank 10 and 25 for recommendation list sizes 10 and 25.

We are using five common algorithms for our analysis: Funk SVD, Item-Item collaborative filtering, User-User collaborative filtering, and content based filtering and popularity based recommender. Each of these algorithms is configured as follows:

- Funk SVD: Feature count is set to 20 and iteration count is set to 125.
- Item-Item collaborative filtering: Neighborhood size is set to 20.
- User-User collaborative filtering: Neighborhood size is set to 30 and damping factor is set to 25.
- Content based filtering: Neighborhood size is set to 20 and model size is set to 100.

The above configurations are considered to be the best fit for the MovieLens data and the algorithms will work their best (Ekstrand et al. 2011). In this experiment, we set up two metrics called: RMSE and MAP for each of the recommendations that every algorithm predicted. Each users' MAP and RMSE are generated in a csv file called eval-user.csv. We are also using the partitioned test data that are generated from the 5-fold cross evaluation. This is to get the diversity metric and popularity metric for test data and train data so that we can profile recommenders output and users' input respectively. All the configuration files are included in the Appendix section at the end of the document.

3.3 ANALYSIS

R (R. Core Team 2012) is a powerful open source software used for statistical computing. We use R as a tool for data analysis in our research. We are also using a web based application called Jupyter notebook (Pérez and Granger 2007). We used the R packages like dplyr (Wickham, Francois, and RStudio 2016), reshape2 (Wickham 2014), plyr (Wickham 2016), ggplot2 (Wickham, Chang, and RStudio 2016) and gridExtra (Auguie and Antonov 2016).

We analyzed the diversity and popularity bias in the users' profile separately. We obtained train and test data from the LensKit experiment we conducted and considered the train data as the users' input profile and test data as the recommender's output profile. For diversity bias, we used four metrics that are described as follows:

- Intra List Similarity (ILS) metric: This is obtained by measuring intra-list similarity (Ziegler et al. 2005) using Pearson correlation over tag genome vectors (Vig, Sen, and Riedl 2009) as the similarity metric.

- Average Intra List Similarity metric: Average ILS metric is obtained by first measuring ILS for every item in the recommendation list and by taking average of every prefix in the list. This metric helps us to give a weightage to the position of the item in the recommendation list.
- Entropy: Entropy is obtained by calculating the probability of the tag that is tagged to that particular movie in the list and every item in the recommendation list is given equal weightage.
- Discounted Entropy: Discounted entropy is the metric that discounts the item if it is recommended on the later part of the recommendation list. In this case, a weightage is given to each item based on the position of the item in the recommendation list.

For popularity bias, we used the metric described below:

- Mean popularity rank: Mean popularity rank is obtained by first obtaining the popularity rank of each item and then taking the mean of the items per user. Popularity of the items is obtained by counting the number of ratings per each item. An item that has the highest number of ratings is considered to have a popularity rank of one.

For analyzing the popularity bias of the user, we obtained the popularity rank metric for each movie based on the number of ratings for that movie. The higher the number of ratings for a movie, the more popular it is. We then obtained the mean popularity of the movies that are rated by each user. This gave us the user profile and the recommender profile for popularity bias.

4. RESULTS

In this chapter, we present our main results on the ways in which user profile characteristics do or do not propagate into recommendation sets for different metrics. We have organized these results around the major characteristics we are considering: popularity and diversity. Throughout this analysis, we also consider the impact of popularity and diversity on the recommender's accuracy, as measured with RMSE (prediction accuracy) and MAP (top-N recommendation accuracy). We present our results for both top-10 and top-25 recommendation lists. The coefficients and p-values of linear regression models and paired t-tests are tabulated and are placed in Appendix section.

The popularity recommender algorithm we have used in the experiment is expected to not to propagate the users' input bias as it does not produce any kind of predictions. The recommendations produced to the users by this algorithm do not have any RMSE due to this reason.

4.1 STRUCTURE OF RESULTS

In each section, we present our observations in the following order:

1. We discuss if any recommender is propagating users' input profile bias through their recommendations.
2. We discuss the use of linear regression models and paired t-tests results to check if our observations mentioned above are supported.
3. We discuss about the histograms of the number of users and the users' input profile diversity/popularity bias.
4. We discuss the impact of popularity/diversity on the recommender's RMSE

5. We discuss about the impact of popularity/diversity on recommender's MAP values.

Graphs like recommender's output verses users' input profile, RMSE verses users' input profile and MAP verses users' input profile has scatter plot and line plots combined in one graph. Blue lines in these graphs are acquired by smoothing the scatter plot and the red lines in the graph are acquired by using the intercept and slope values which we obtained from the linear regression models. The graphs that support the linear regression models are mentioned in the Appendix section at the end of the document.

4.2 POPULARITY

As described in the methodology section, we used the mean popularity rank metric for measuring popularity bias. Ideally, the personalized recommenders are expected to propagate the user bias. But we see that mostly the recommenders do not propagate the popularity bias. The observations below will give much clear idea about it.

In figure 1 and figure 2, we see that all the algorithms except for Content-Based filtering (CBF) show pretty flat curves. CBF curve shows positive slope until some extent and them reduces. This shows that all algorithms except CBF do not propagate user popularity bias through its output.

The linear regression models and paired t-test are performed to find the correlation between the recommender profile and user profile. Table 1 and table 2 show the linear regression model variables for recommendation list sixe 25 and 10 respectively. The coefficient values are pretty low which tells us that the recommender profile and the user profile are not depend on each other. Table 3 shows the paired t-test values and the mean of the differences for every algorithm is pretty high which tells us that the differences are statistically significant.

From figure 7 and figure 3, we can see that recommender profile has wider range of popularity profile for Item-Item and CBF when compared to user profile. This shows that the recommenders tend to recommend less popular items even if the users are biased to more popular items.

In figure 4, we can see that RMSE verses user profile curves are pretty flat for every algorithm. This shows that RMSE values are not really impacted by user profile popularity bias. The RMSE curves are mainly determined by the recommender algorithm performance irrespective of the user bias in popularity. The linear regression model for RMSE verses user profile in table 1 show that they are not dependent on each other.

In figure 5 and figure 6, we can see that MAP verses user profile curves are pretty flat. Again, this shows that MAP doesn't change with the user profile popularity bias and the linear regression model variables support it (table 1 and table 2).

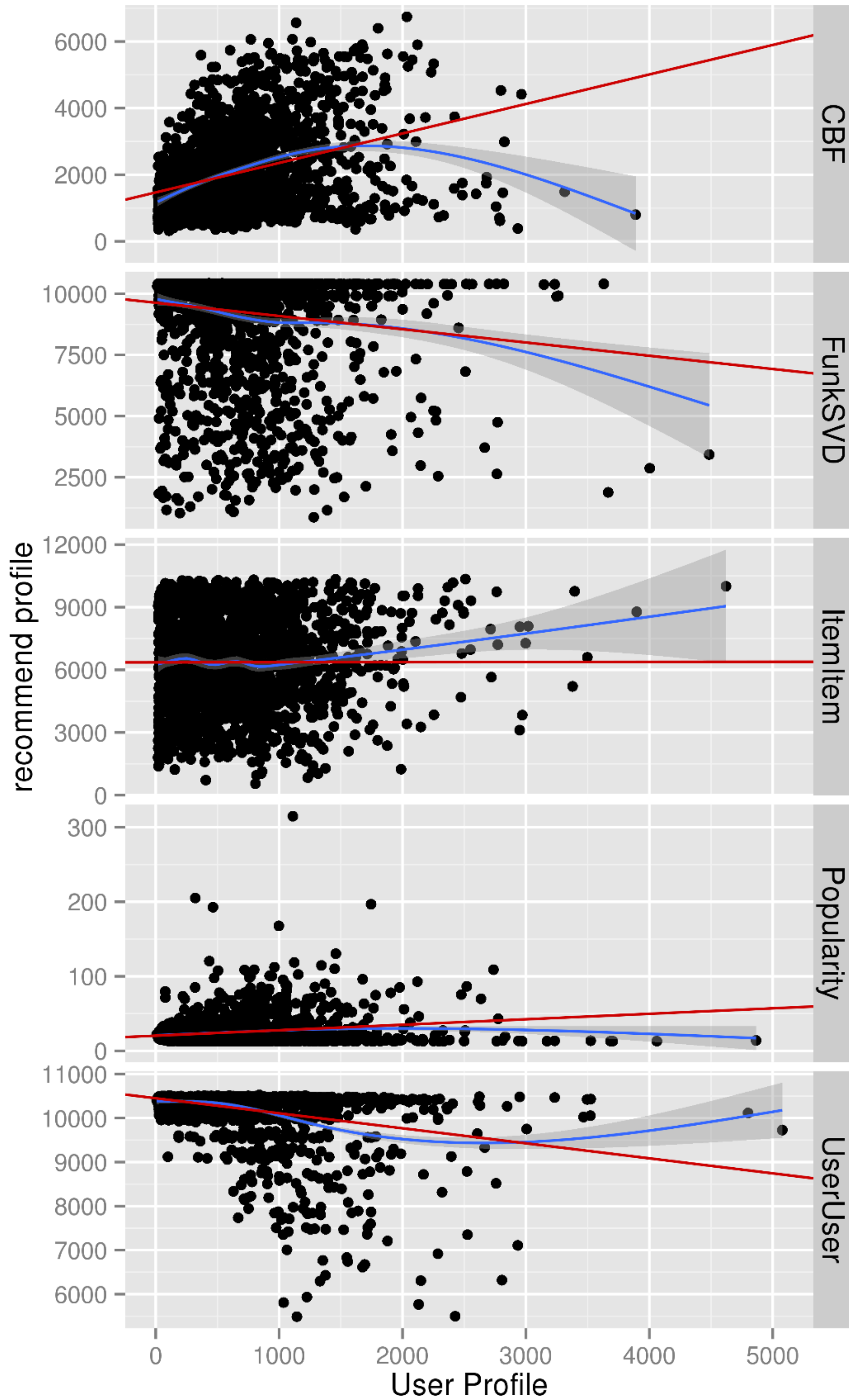

Figure 1: Recommender Profile verses User Profile for list size 25

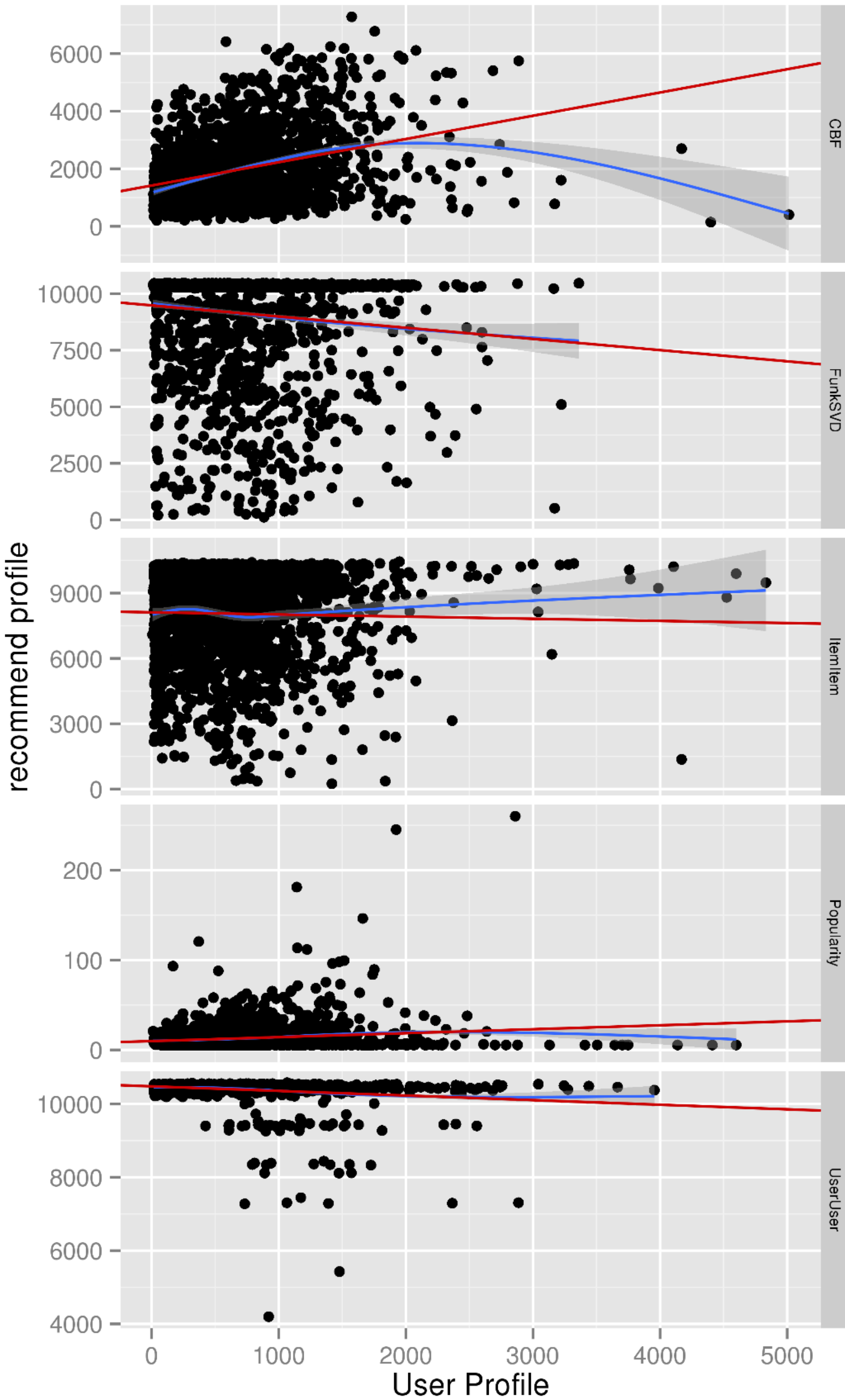

Figure 2: Recommender Profile verses User Profile for list size 10

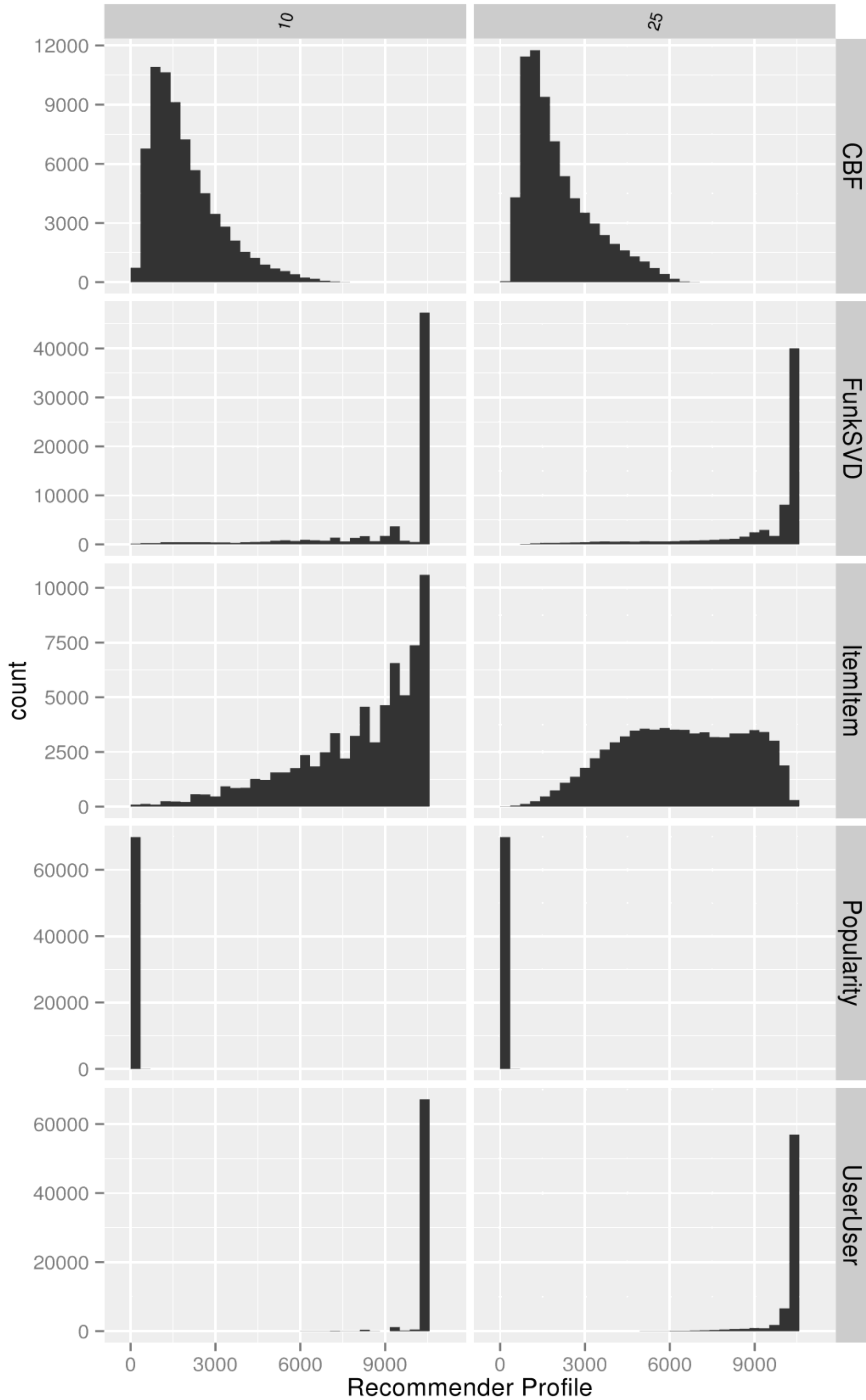

Figure 3: Histograms for Recommender Profile for recommendation list size 10 and 25

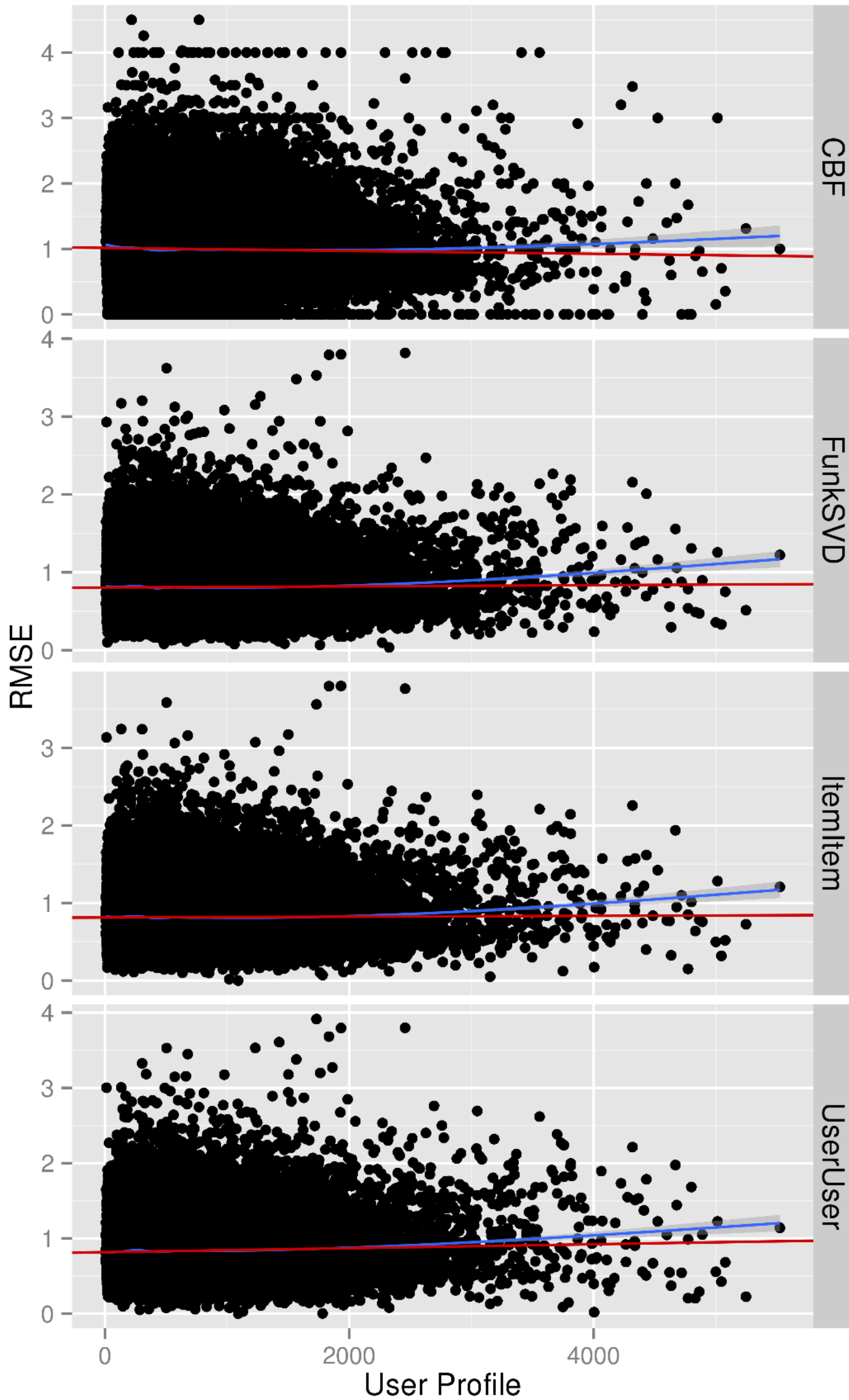

Figure 4: RMSE verses User Profile

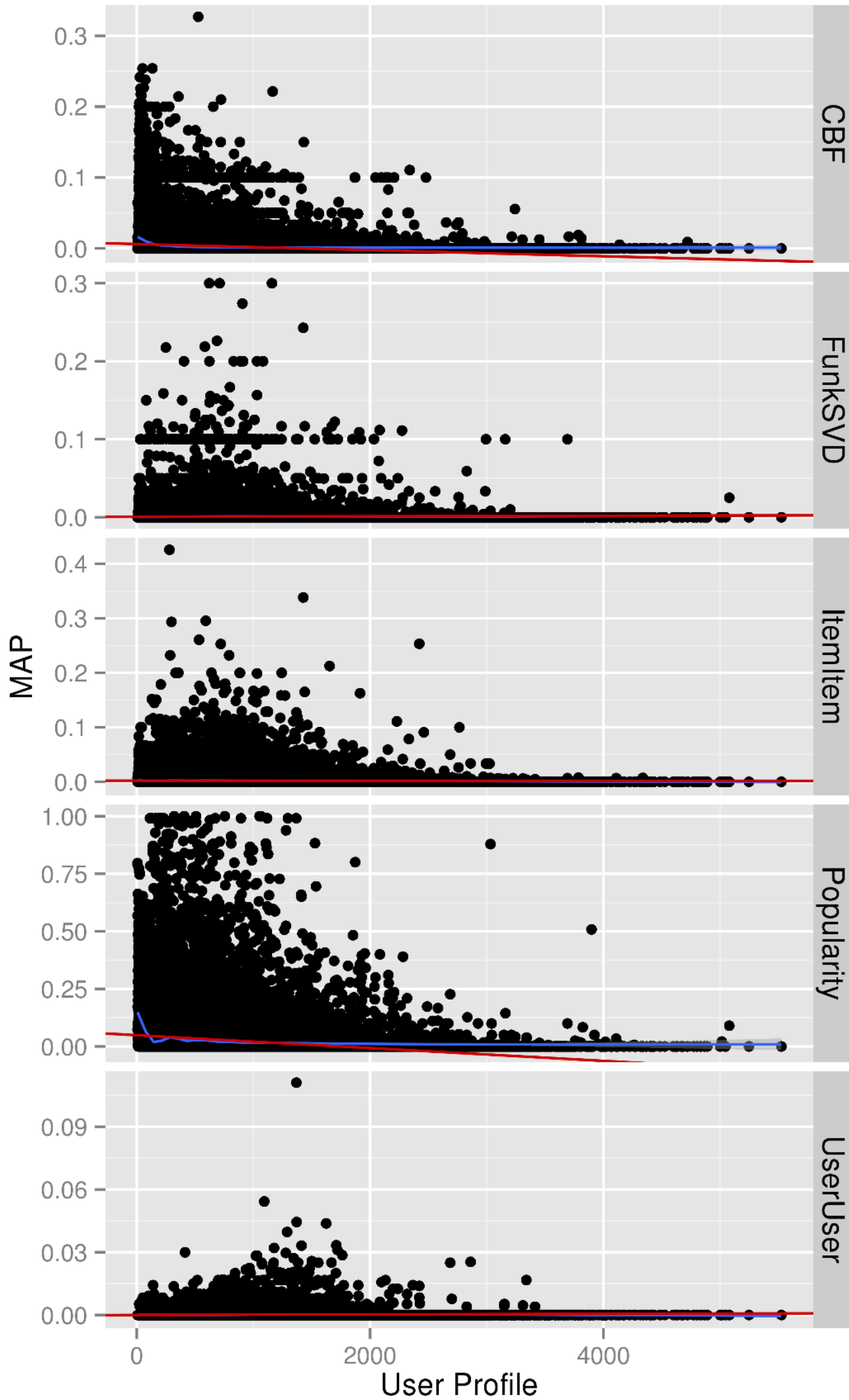

Figure 5: MAP verses User Profile for list size 25

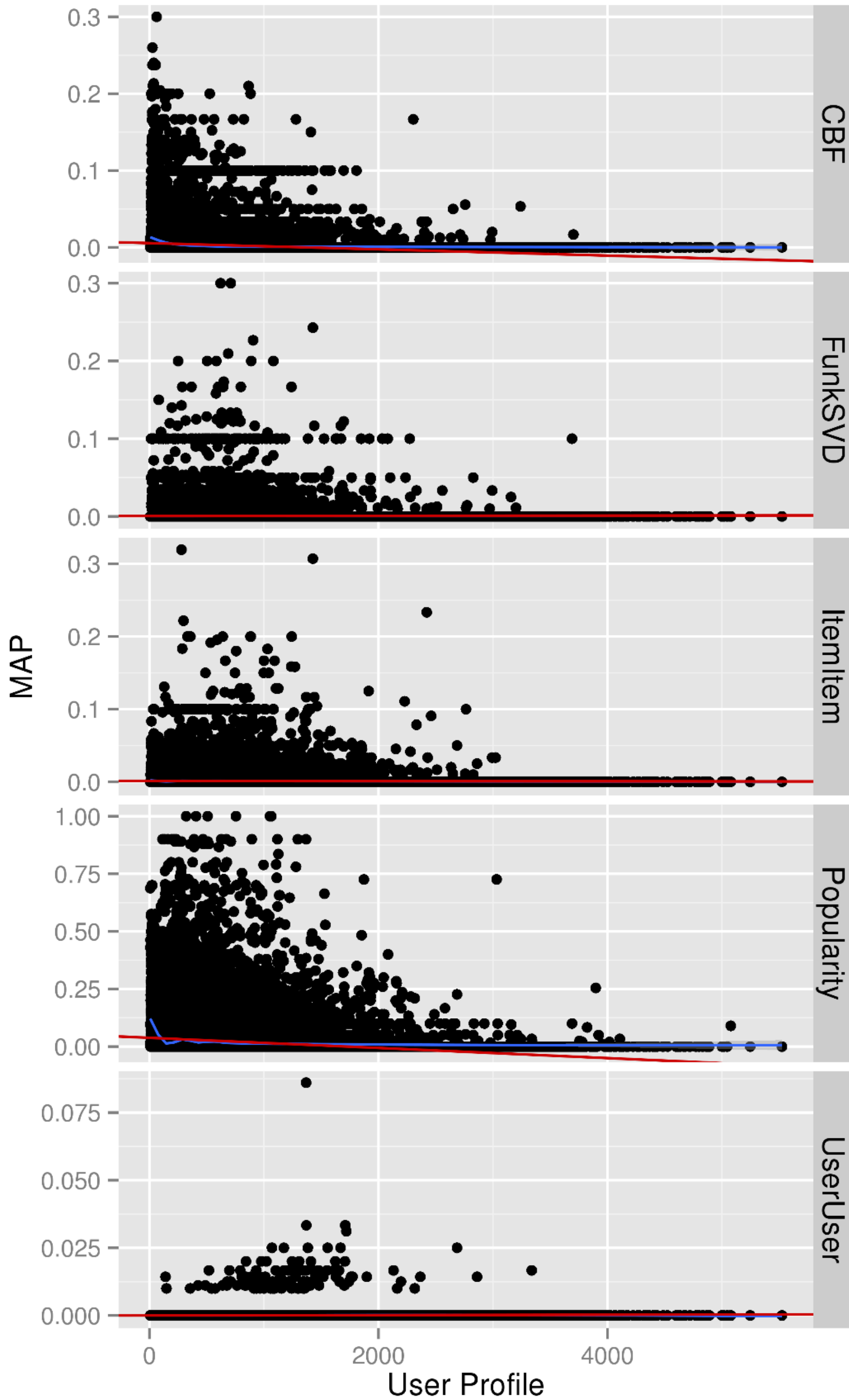

Figure 6: MAP verses User Profile for list size 10

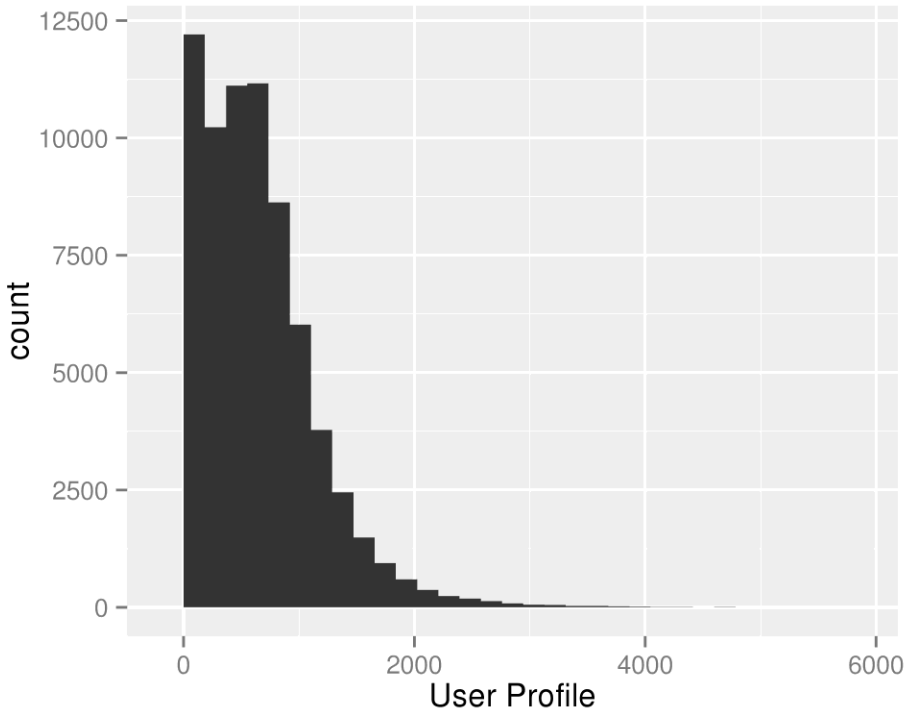

Figure 7: Histograms for User Profile for popularity bias

4.3 DIVERSITY BIAS

4.3.1 INTRA-LIST SIMILARITY (ILS) METRIC

The ILS metric is measured as explained in the Methodology section.

In figure 8 and figure 9, we see that recommender profile verses user profile curves for all the algorithms except CBF has flat curves. The negative slope in CBF shows that it actually counteracts the user profile diversity bias. This also tells us that the algorithms except CBF do not propagate diversity bias to recommender's outputs.

The linear model variables are tabulated in table 4 and table 5 in Appendix. The coefficients are pretty low which supports our observation in figure 8 and figure 9. The corresponding p-values are also quite low which means that the coefficients cannot be more than what we obtained. In table 10, the paired t-tests illustrate that the mean of the differences between recommender diversity and user profile diversity is not statistically significant.

From figure 14 and figure 10, we can see that recommender profile has wider range of diversity profile when compared to user profile. The recommender profile shows that the items are less diverse even if the user has slightly high diverse items in their profile.

In figure 11, we can see that RMSE verses user profile curves are pretty flat for every algorithm. This shows that RMSE values are not really impacted by user profile diversity bias. The RMSE curves are mainly determined by the recommender algorithm performance irrespective of the user bias in diversity. The linear regression model for RMSE verses user profile in table 4 show that they are not dependent on each other.

In figure 12 and figure 13, we can see that MAP verses user profile curves are pretty flat. Again, this shows that MAP doesn’t change with the user profile diversity bias and the linear regression model variables support it (table 4 and table 5).

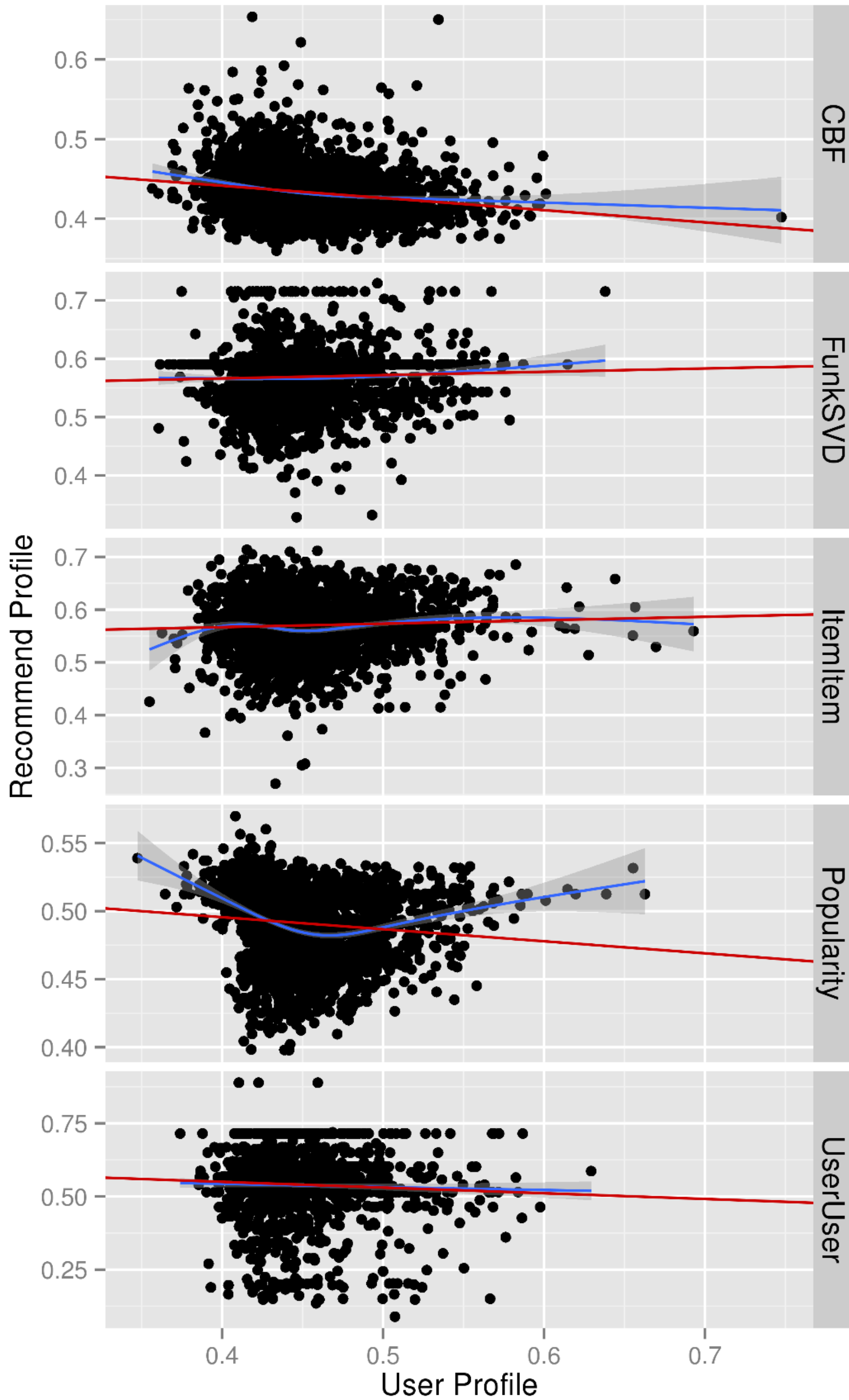

Figure 8: Recommender Profile verses User Profile for list size 25

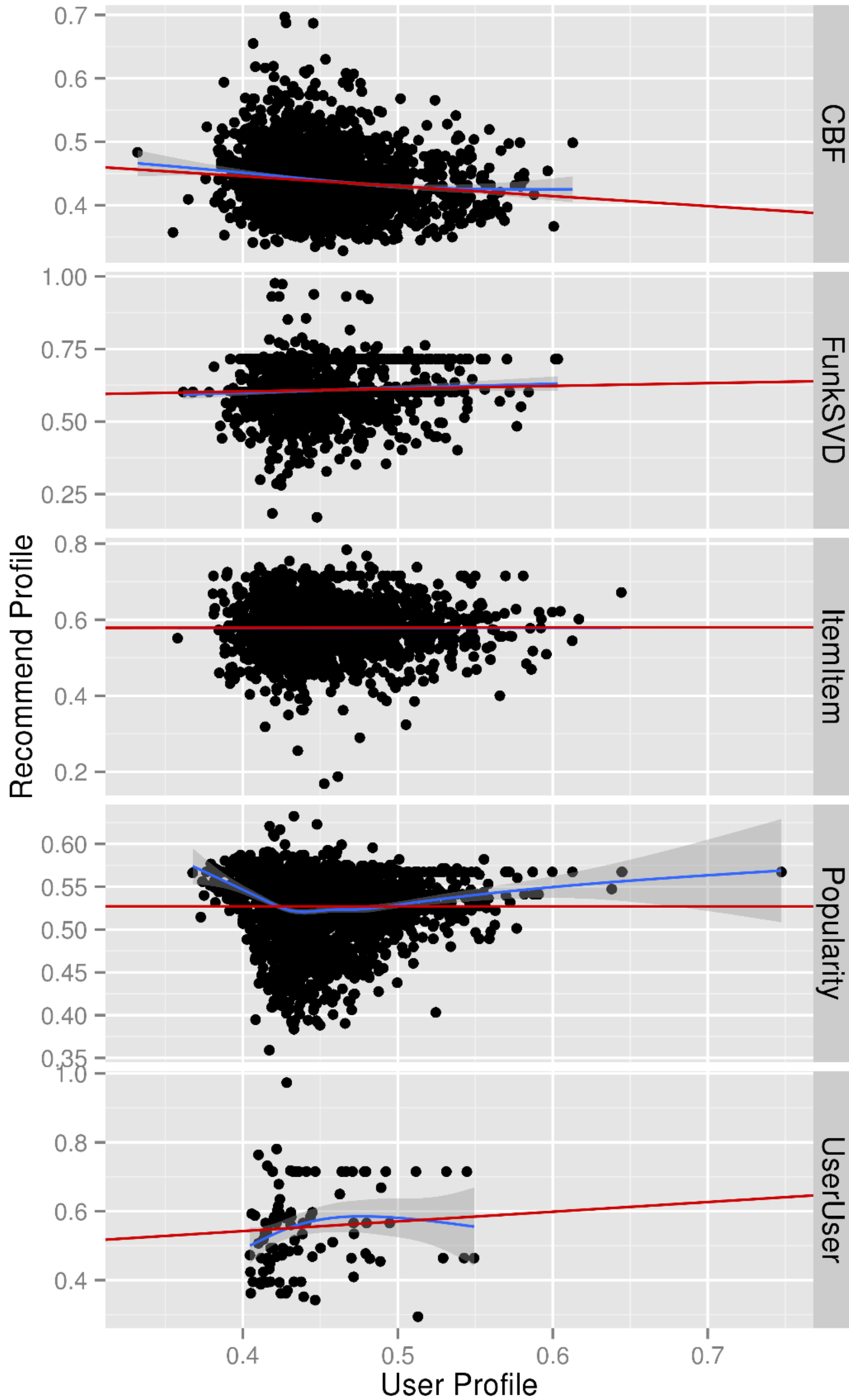

Figure 9: Recommender Profile verses User Profile for list size 10

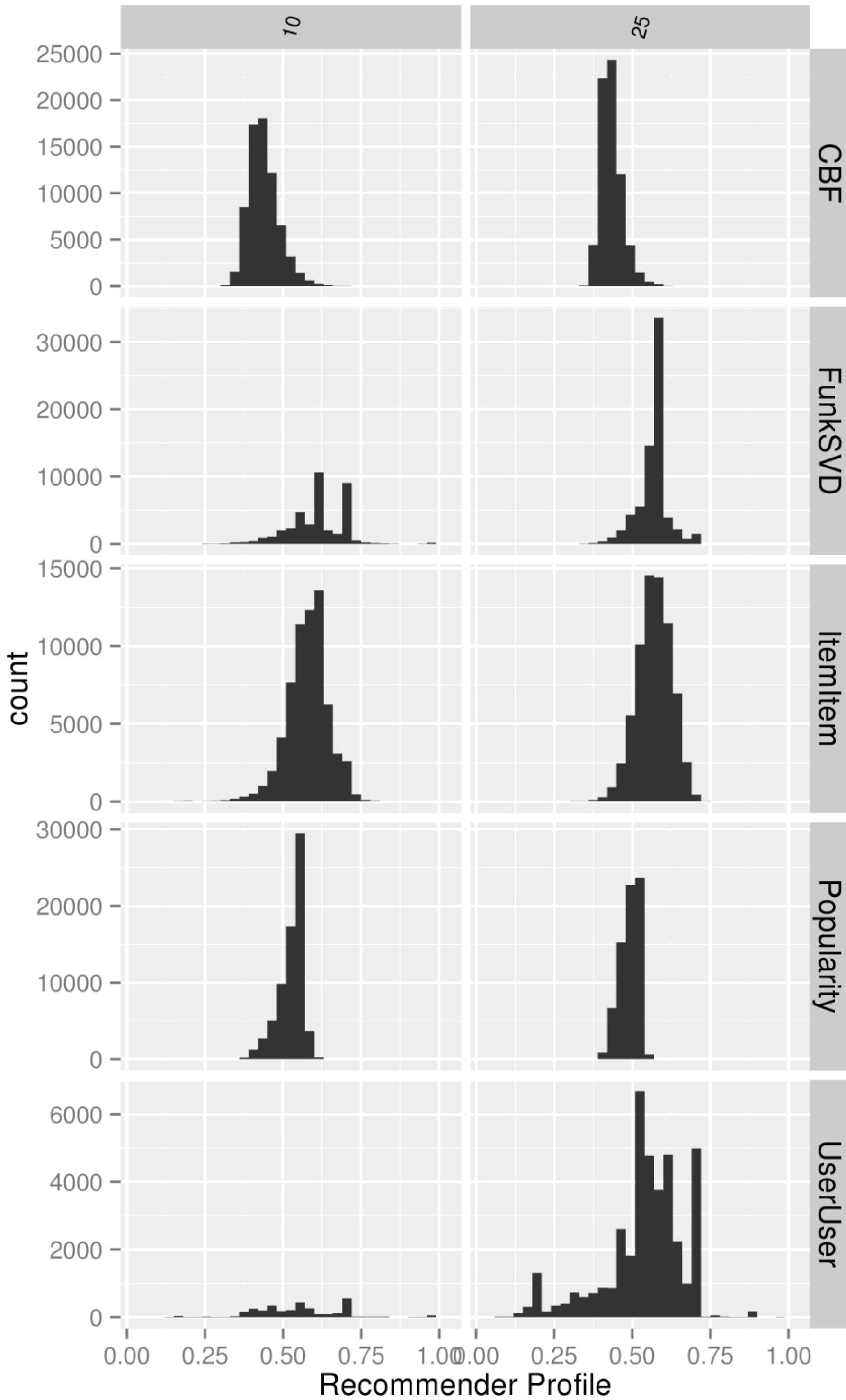

Figure 10: Histograms for Recommender Profile for recommendation list size 10 and 25

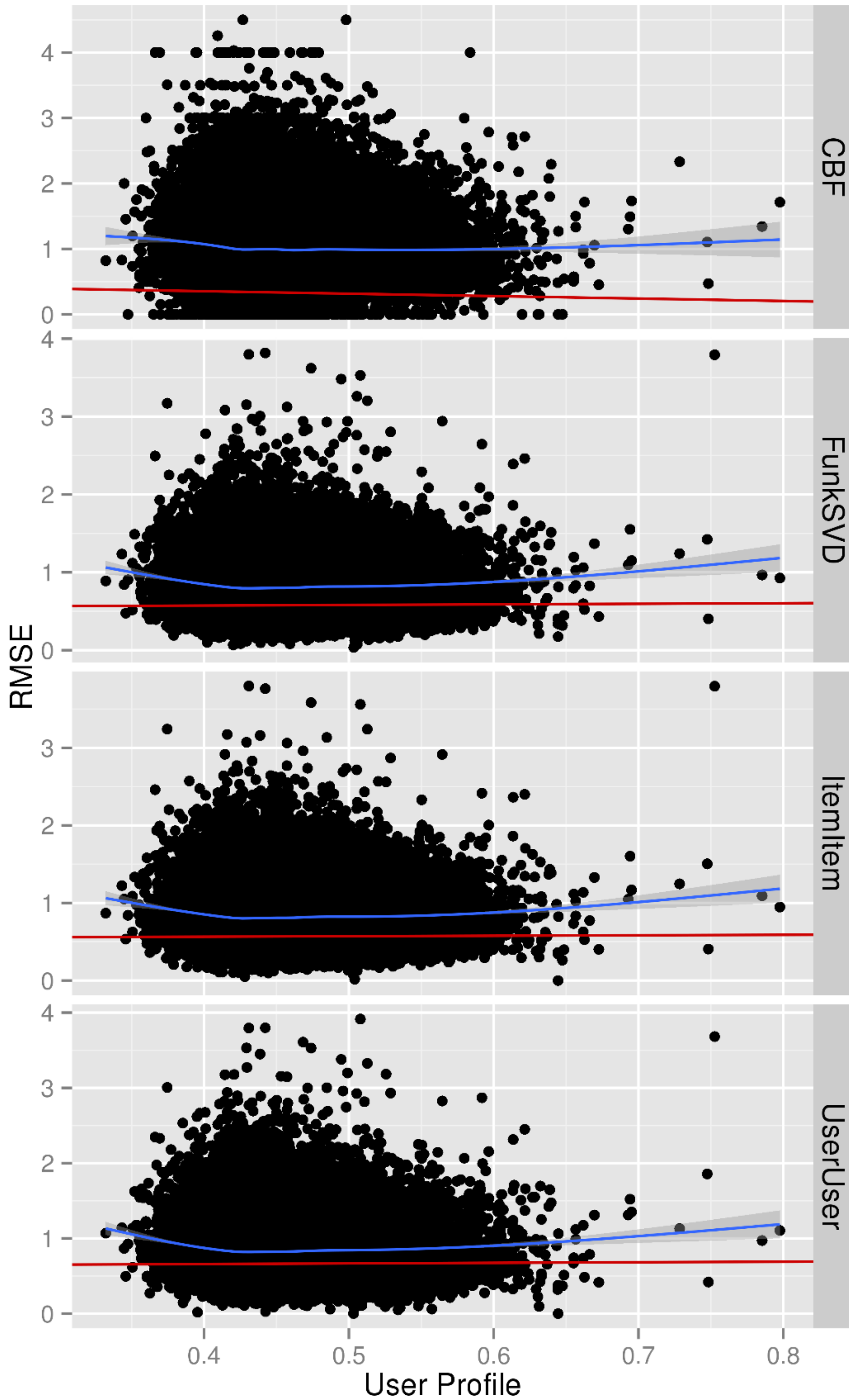

Figure 11: RMSE verses User Profile

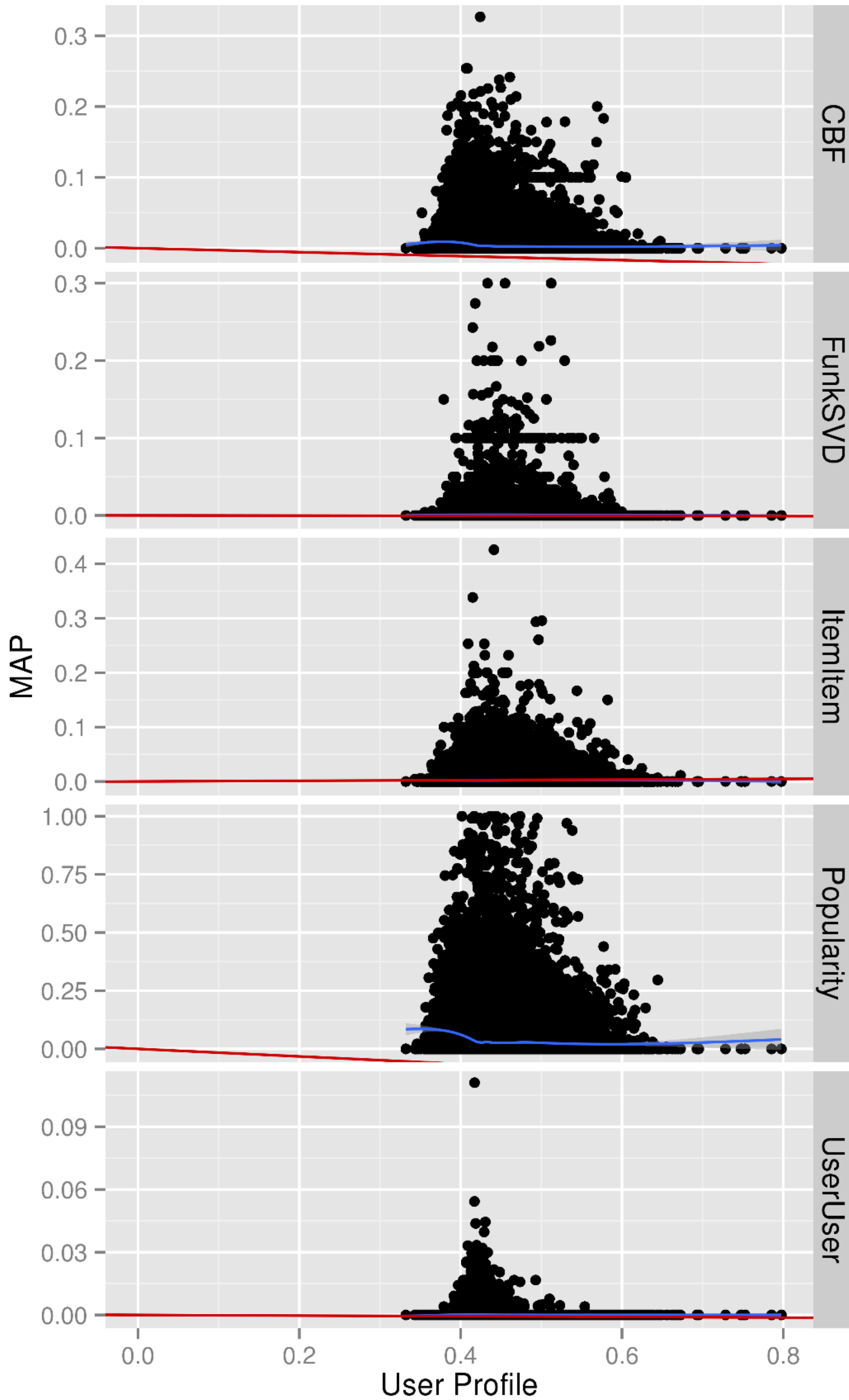

Figure 12: MAP verses User Profile for list size 25

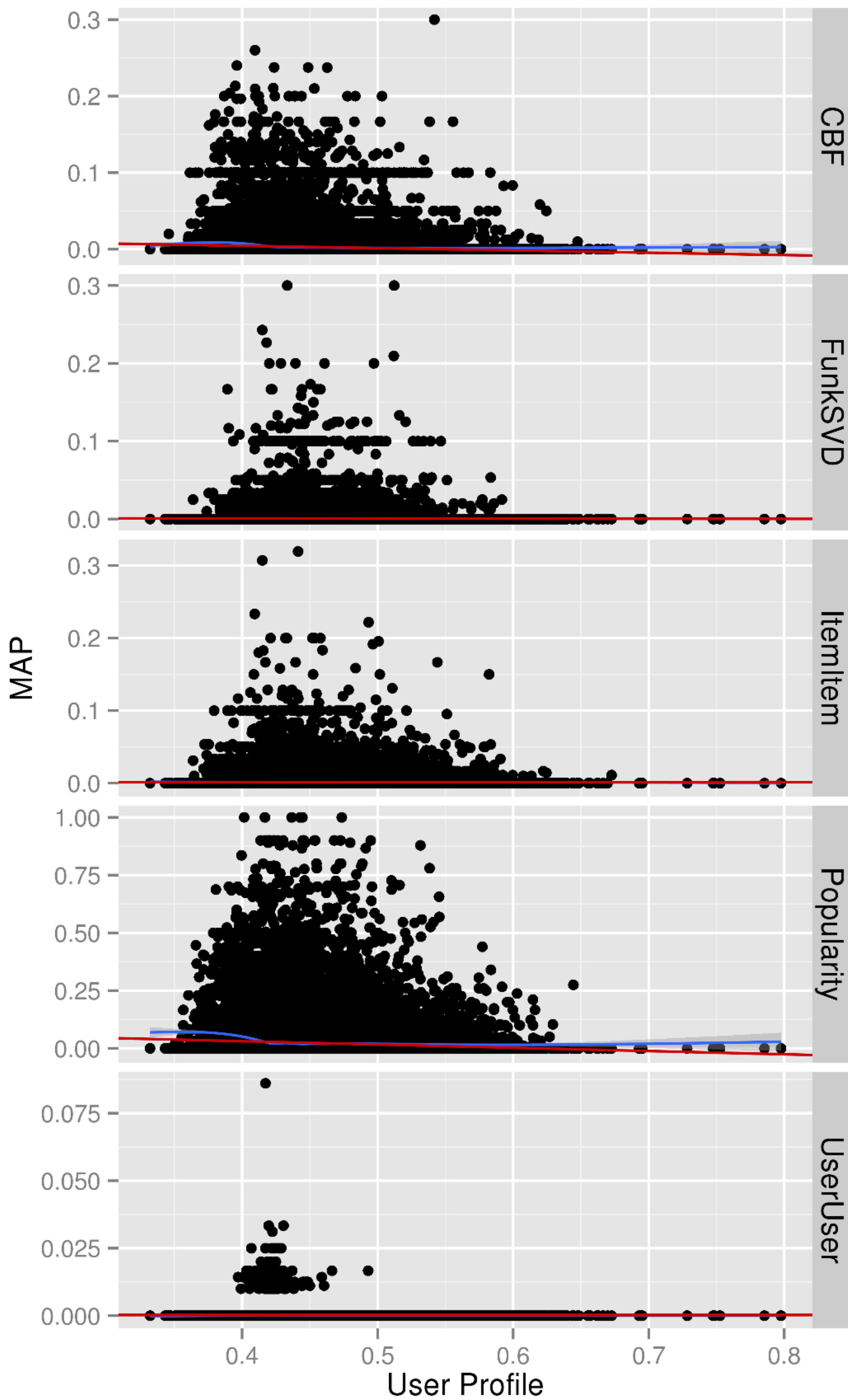

Figure 13: MAP verses User Profile for list size 10

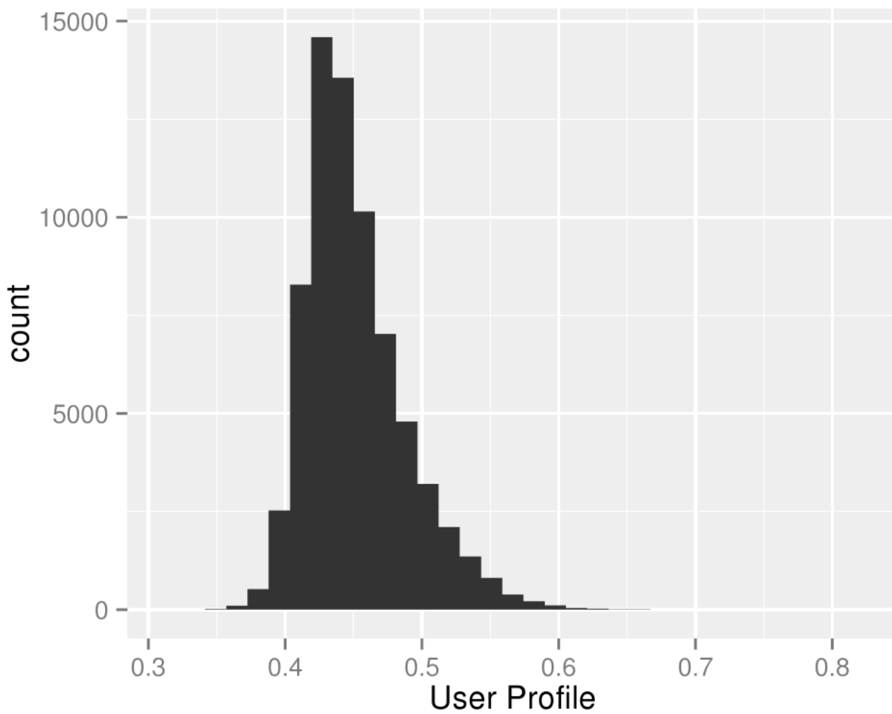

Figure 14: Histograms for User Profile

4.3.2 AVERAGE INTRA-LIST SIMILARITY (ILS) METRIC

The Average ILS metric is obtained as explained in the Methodology section.

In figure 15 and figure 16, we see that recommender profile verses user profile curves for all the algorithms are flat curves. This tells us that the algorithms we have used do not propagate diversity bias to recommender's outputs.

The linear model variables are tabulated in table 7 and table 8. The coefficients are pretty low which support our observations in figure 15 and figure 16. The corresponding p-values are also quite low which mean that the coefficients cannot be more than what we obtained. We didn't perform paired t-tests because the user profile is obtained using ILS and the recommender profile is obtained using average ILS.

From figure 14 and figure 17, we can see that recommender profile has wider range of diversity profile when compared to user profile. This shows us that the recommenders tend to recommend more diverse items even if the users are biased to less diverse items.

Note that since the user profile items doesn’t have any order in the list, we use ILS metric for user profile. Due to this reason, the correlations between MAP and user profile and RMSE and user profile are same as in ILS metric (figure 11, figure 12 and figure 13).

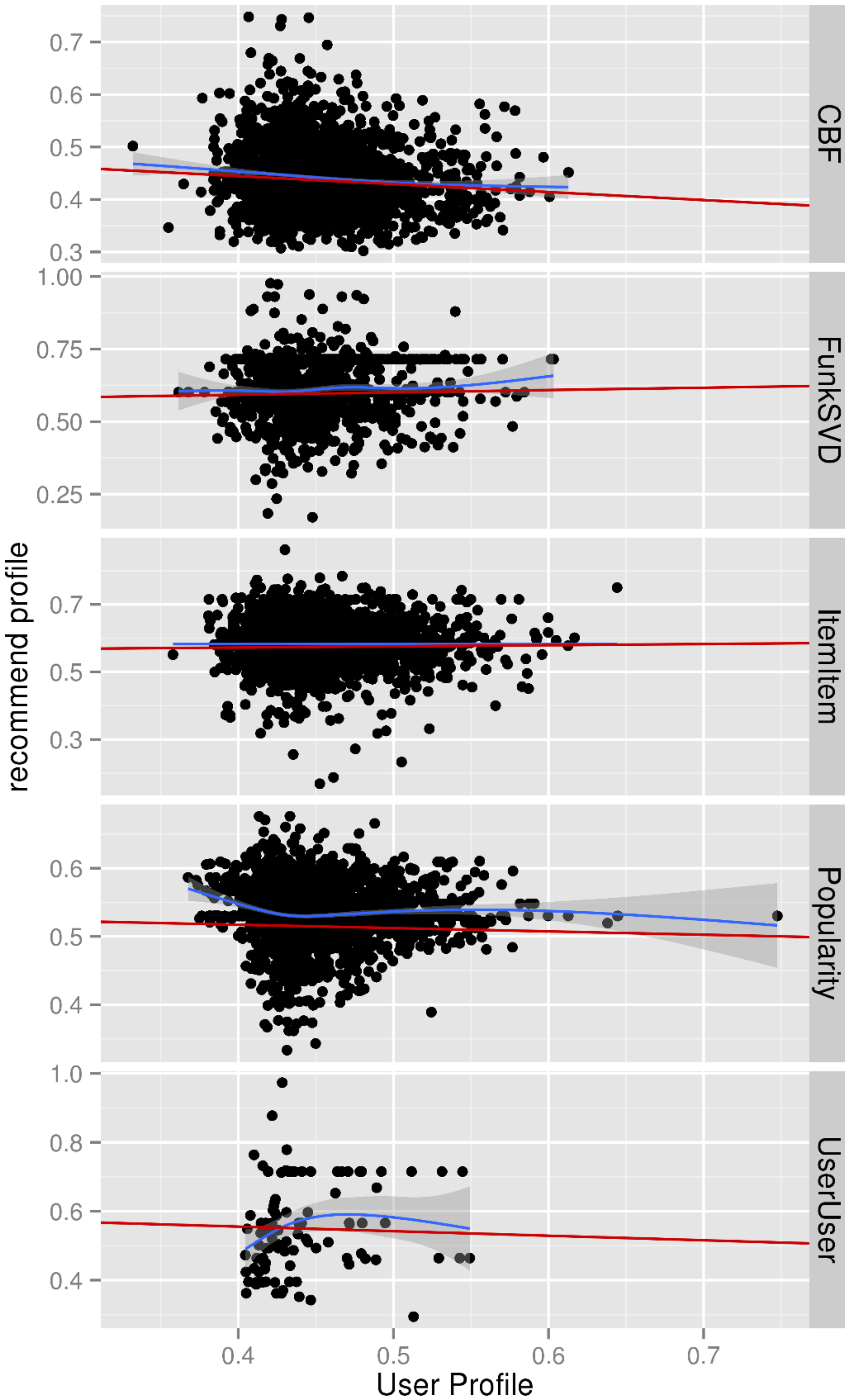

Figure 15: Recommender Profile verses User Profile for list size 25

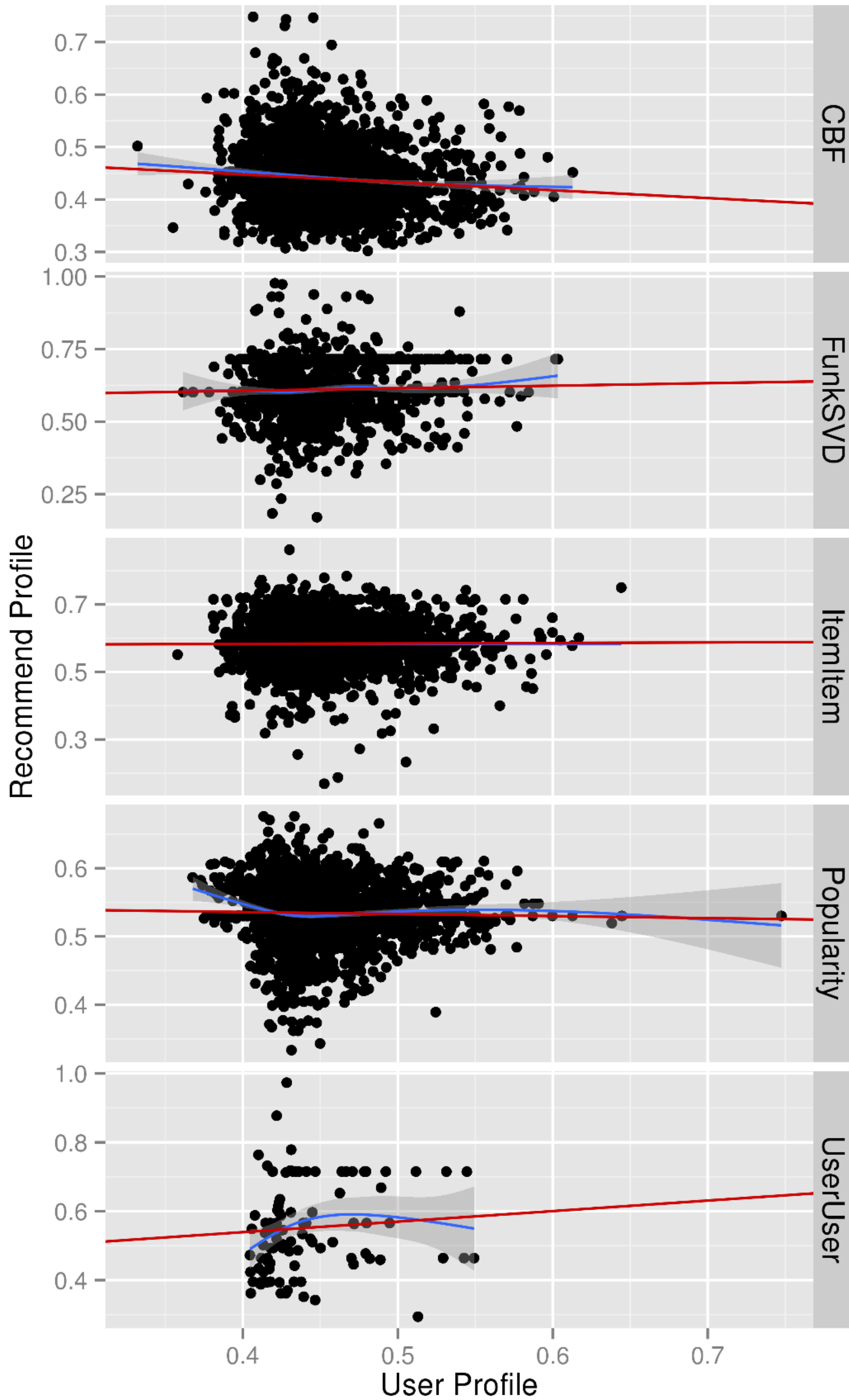

Figure 16: Recommender Profile verses User Profile for list size 10

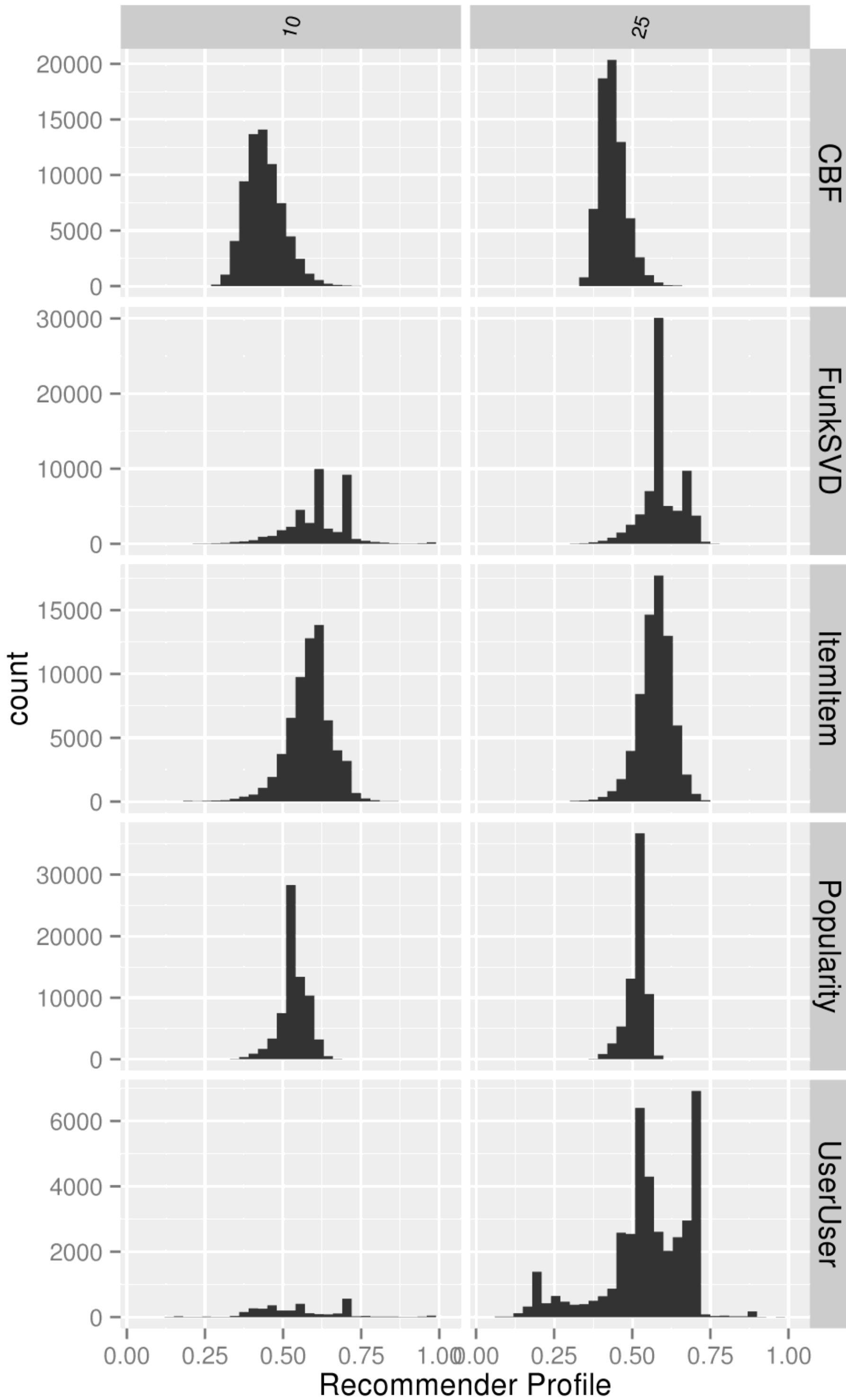

Figure 17: Histograms for Recommender Profile for recommendation list size 10 and 25

4.3.3 ENTROPY

The entropy metric is measured as explained in the Methodology section.

In figure 18 and figure 19, we see that recommender profile verses user profile curves for all the algorithms are flat curves. This tells us that the algorithms we have used do not propagate diversity bias to recommender's outputs.

The linear model variables are tabulated in table 9 and table 10. The coefficients are pretty low which supports our observation in figure 18 and figure 19. The corresponding p-values are also quite low which means that the coefficients cannot be more than what we obtained. In table 10, the paired t-tests illustrate that the mean of the differences between recommender diversity and user profile diversity is not statistically significant.

From figure 24 and figure 20, we can see that the range of entropy is between 9 and 10. Comparatively, the recommender profile has wider range but it's almost insignificant. This shows that entropy is not a good metric to measure

In figure 21, we can see that RMSE verses user profile curves are pretty flat for every algorithm. This shows that RMSE values are not really impacted by the user profile diversity bias. The RMSE curves are mainly determined by the recommender algorithm performance irrespective of the user bias in diversity. The linear regression model for RMSE verses user profile in table 9 show that they are not dependent on each other.

In figure 22 and figure 23, we can see that MAP verses user profile curves are pretty flat. Again, this shows that MAP doesn't change with the user profile diversity bias and the linear regression model variables support it (table 9 and table 10). We observed that entropy is just between 9.5 and 9.7 which shows that they do not capture diversity very accurately. Hence, we don’t consider entropy as a very good metric for diversity. We mention entropy in the document for completeness. The discounted entropy depends on entropy so it is also not considered to be a very good way of measuring the diversity.

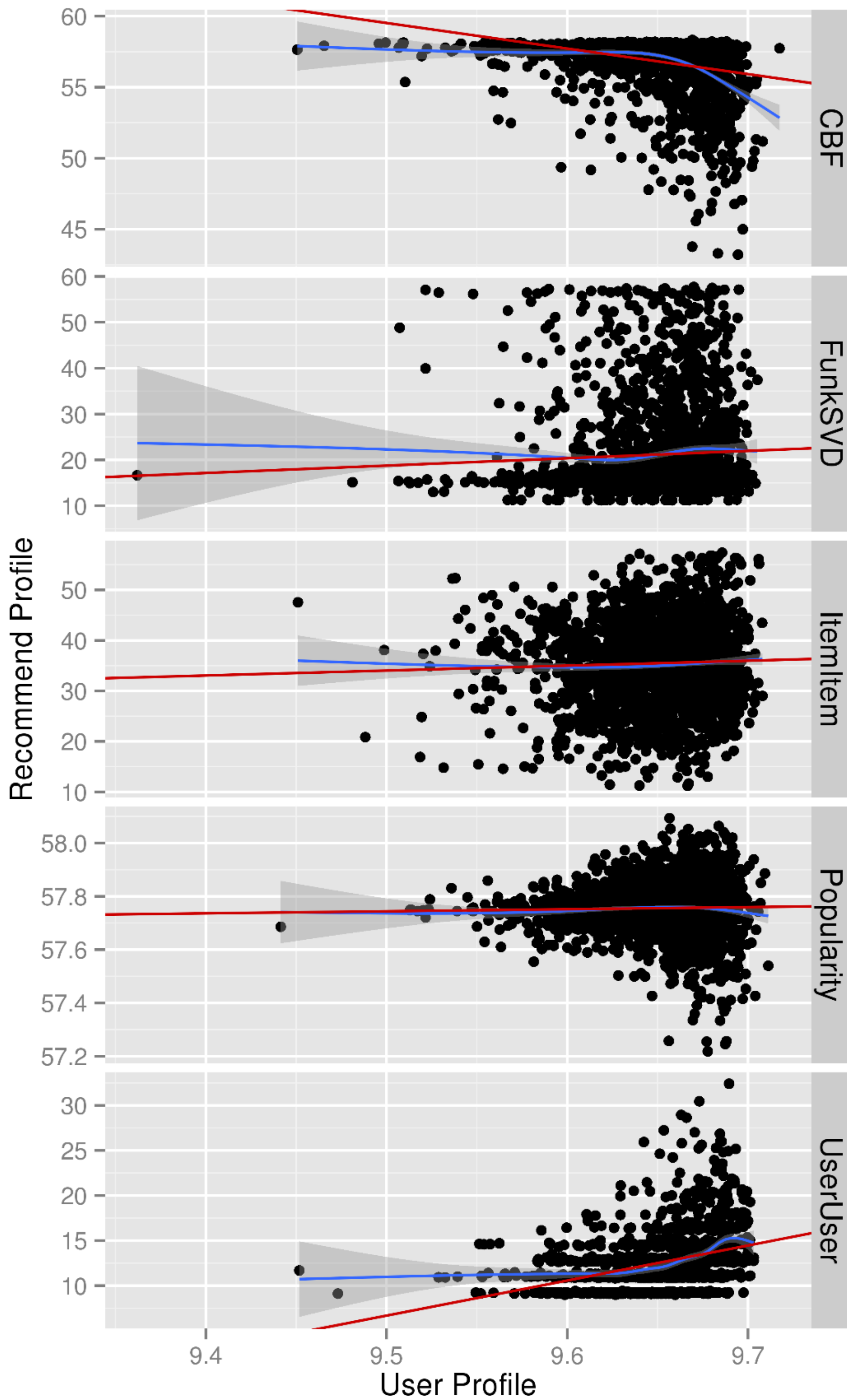

Figure 18: Recommender Profile verses User Profile for list size 25

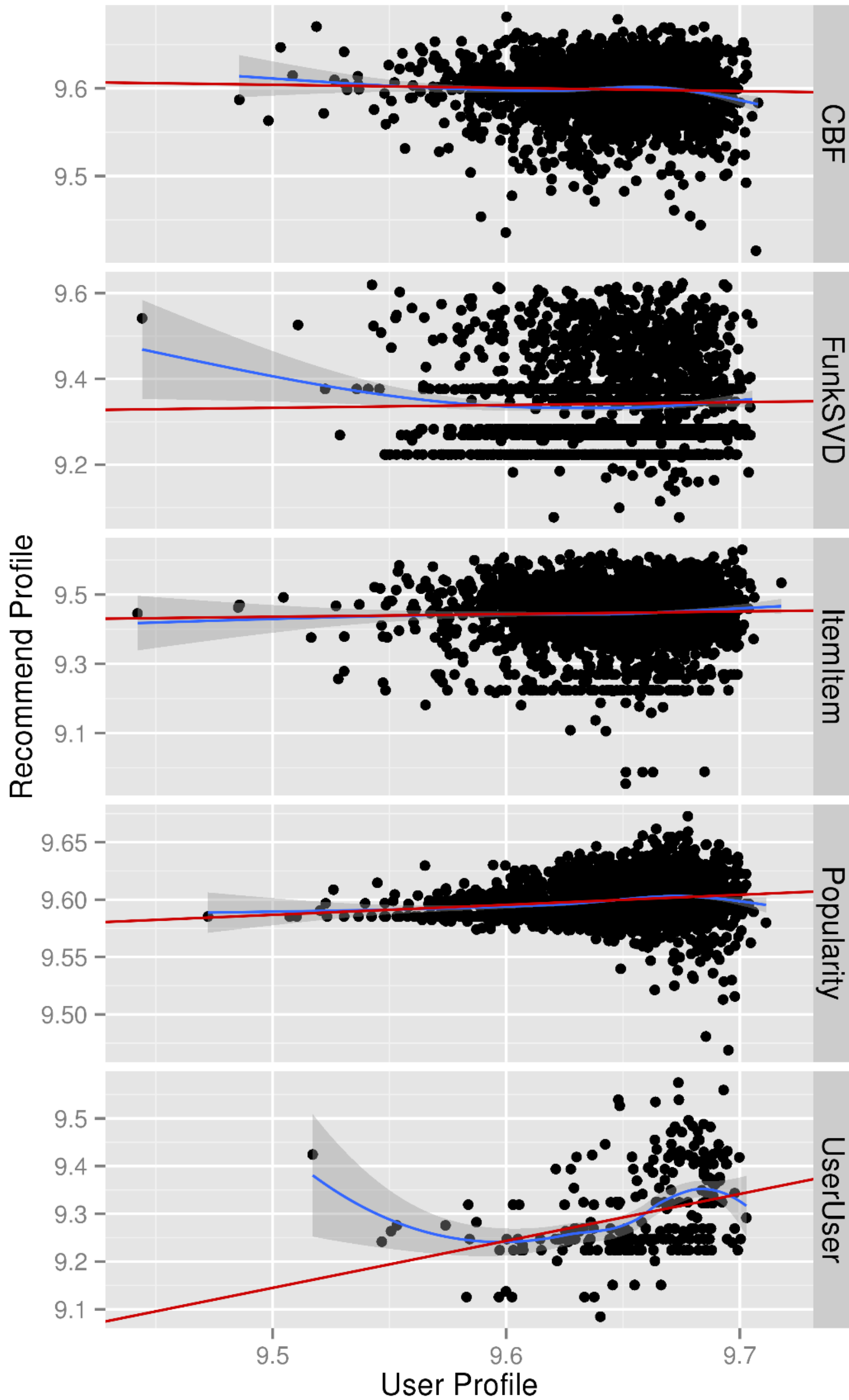

Figure 19: Recommender Profile verses User Profile for list size 10

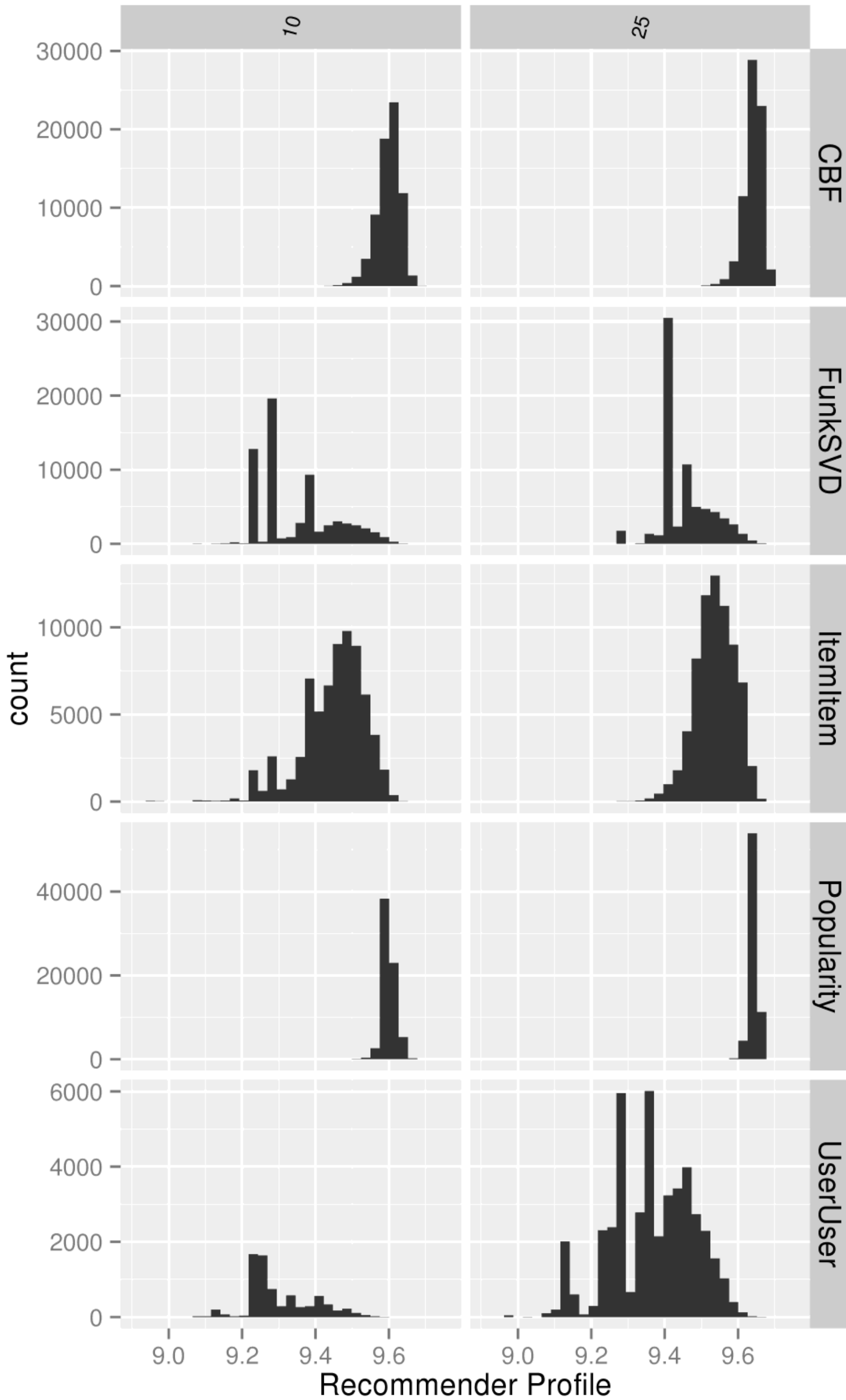

Figure 20: Histograms for Recommender Profile for recommendation list size 10 and 25

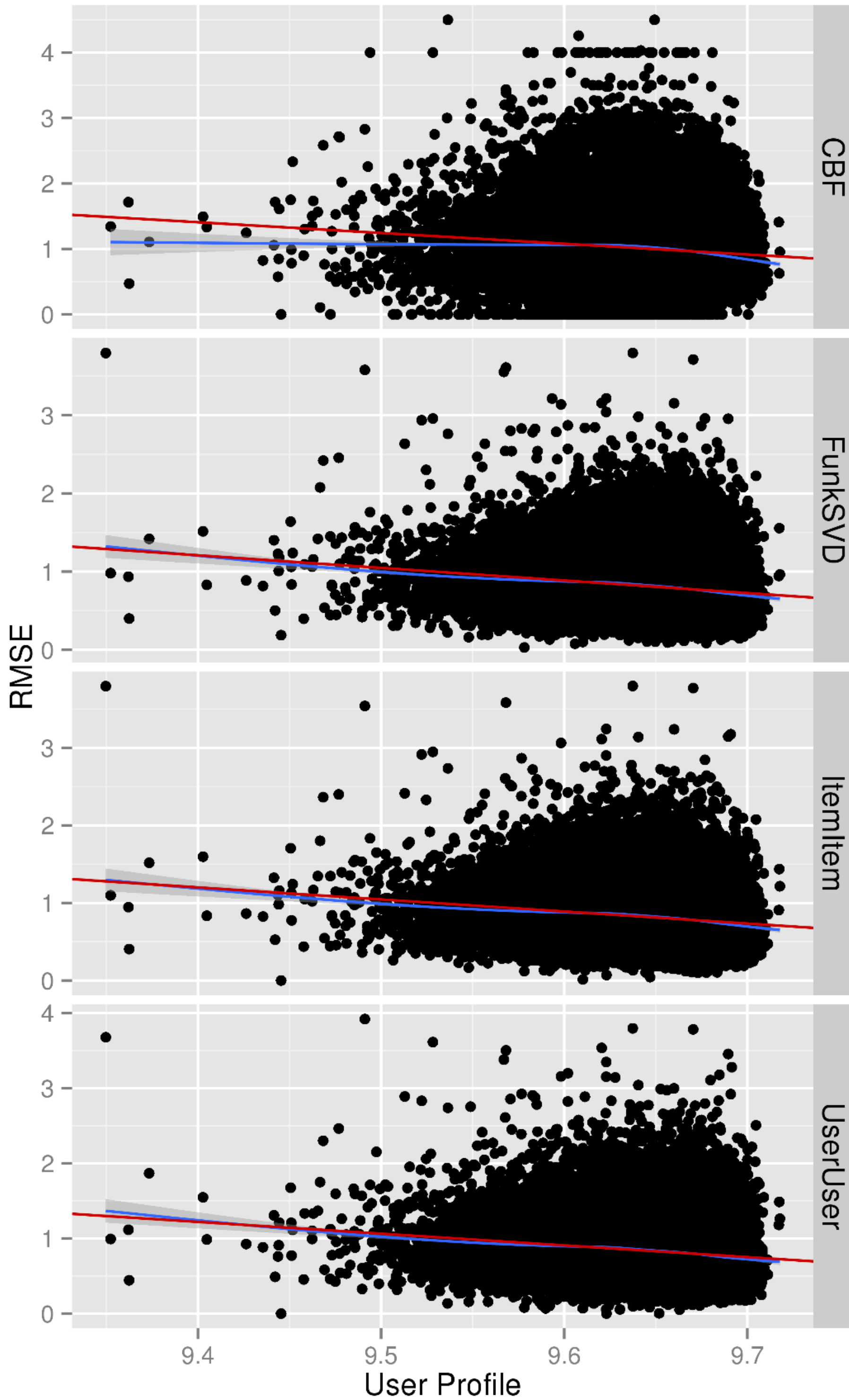

Figure 21: RMSE verses User Profile

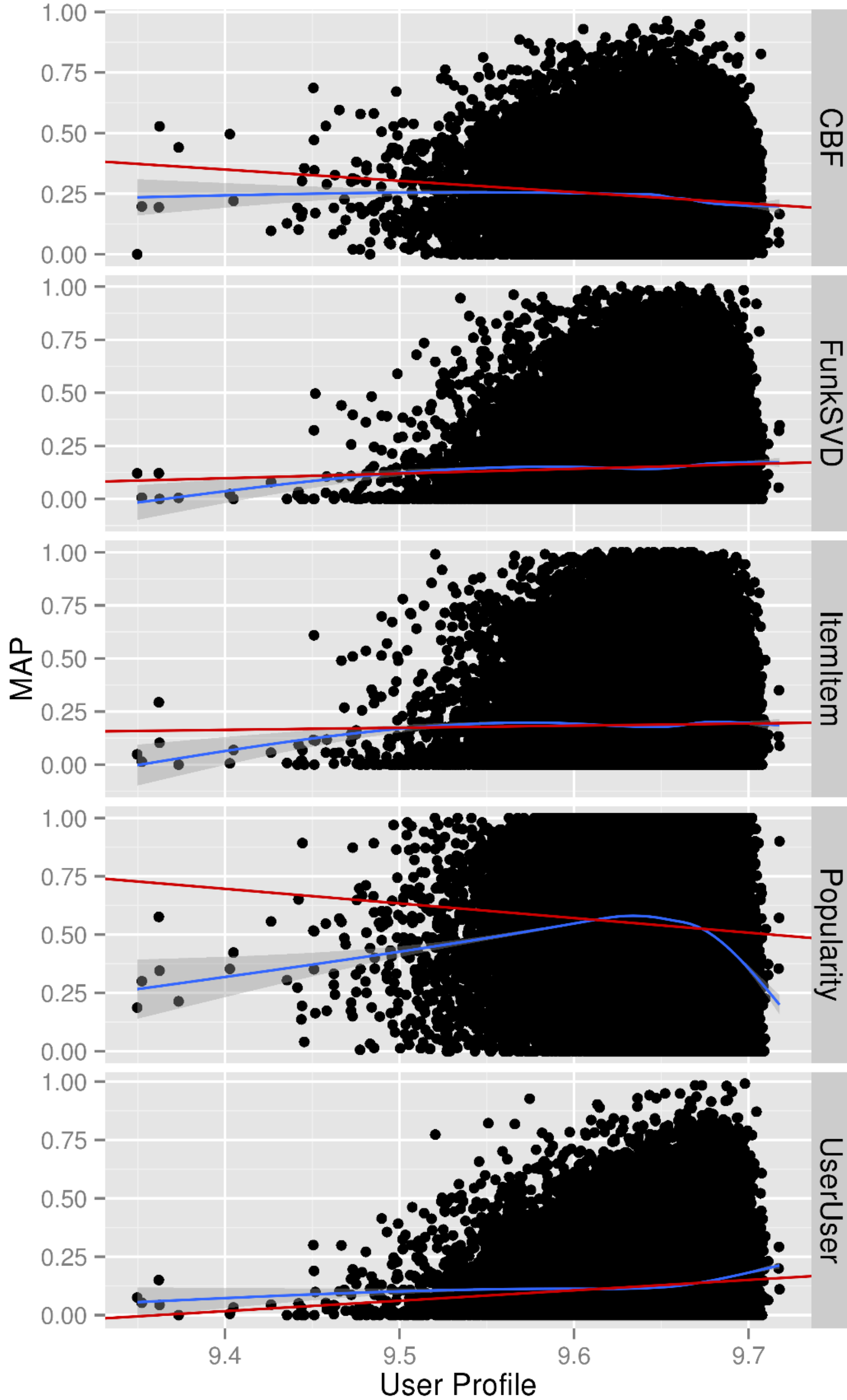

Figure 22: MAP verses User Profile for list size 25

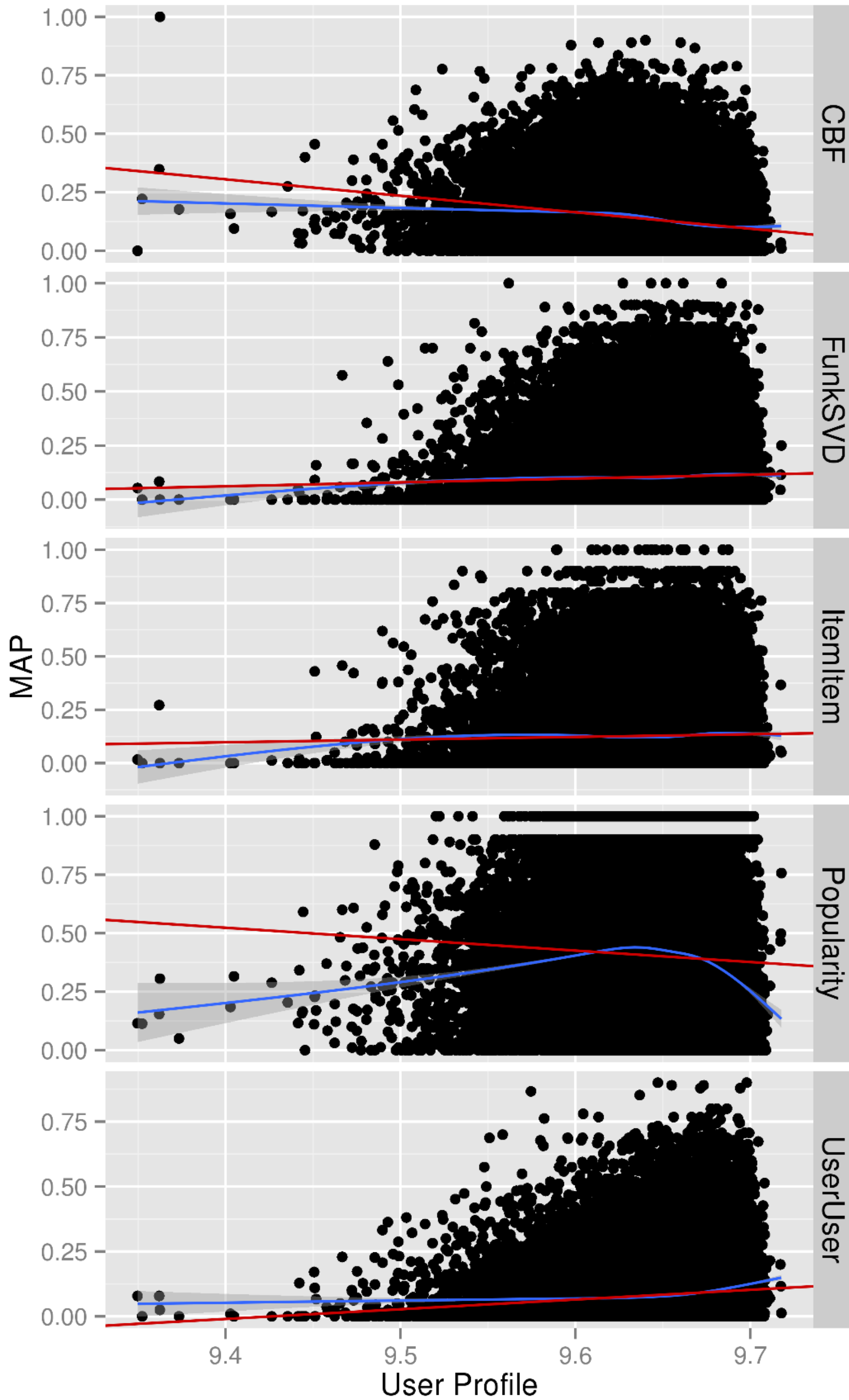

Figure 23: MAP verses User Profile for list size 10

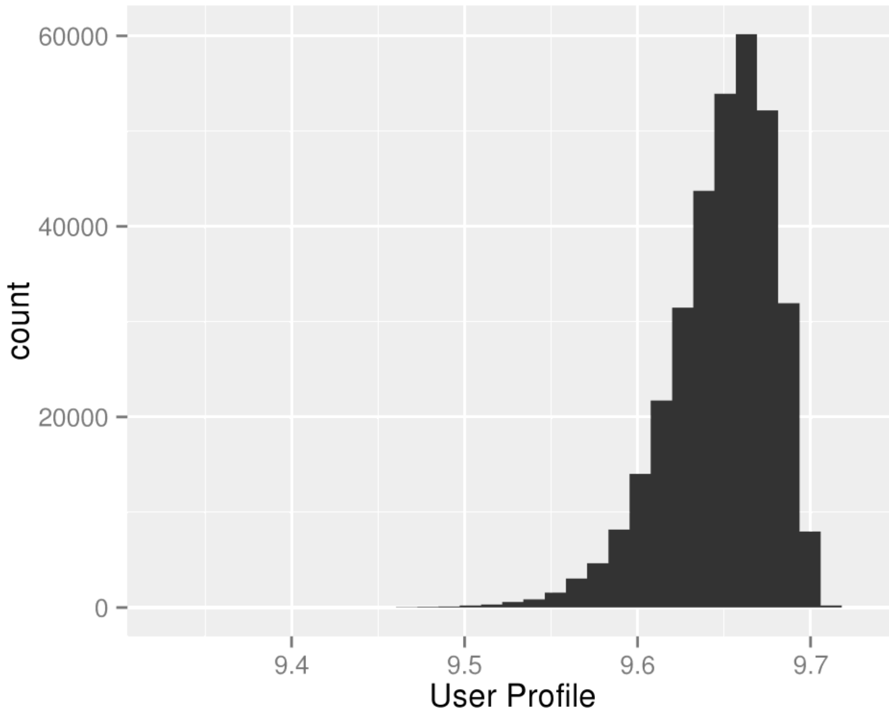

Figure 24: Histograms for User Profile

4.3.4 DISCOUNTED ENTROPY

The Discounted entropy metric is obtained as explained in the Methodology section.

In figure 25 and figure 26, we see that recommender profile verses user profile curves for all the algorithms are flat curves. This tells us that the algorithms we have used do not propagate diversity bias to recommender's outputs.

The linear model variables are tabulated in table 12 and table 13. The coefficients are pretty low which supports our observation in figure 25 and figure 26. The corresponding p-values are also quite low which means that the coefficients cannot be more than what we obtained. We didn't perform paired t-tests because the user profile is obtained using ILS and the recommender profile is obtained using average ILS.

From figure 24 and figure 27, we can see that the metric has a range of 9-10. Comparatively, the recommender profile has a wider range but it's almost insignificant. This shows that entropy is not a good metric to measure

In figure 21, we can see that RMSE verses user profile curves are pretty flat for every algorithm. This shows that RMSE values are not really impacted by the user profile diversity bias. The RMSE curves are mainly determined by the recommender algorithm performance irrespective of the user bias in diversity. The linear regression model for RMSE verses user profile in table 12 show that they are not dependent on each other.

In figure 22 and 23, we can see that MAP verses user profile curves are pretty flat. Again, this shows that MAP doesn't change with the user profile diversity bias and the linear regression model variables support it (table 12 and table 13).

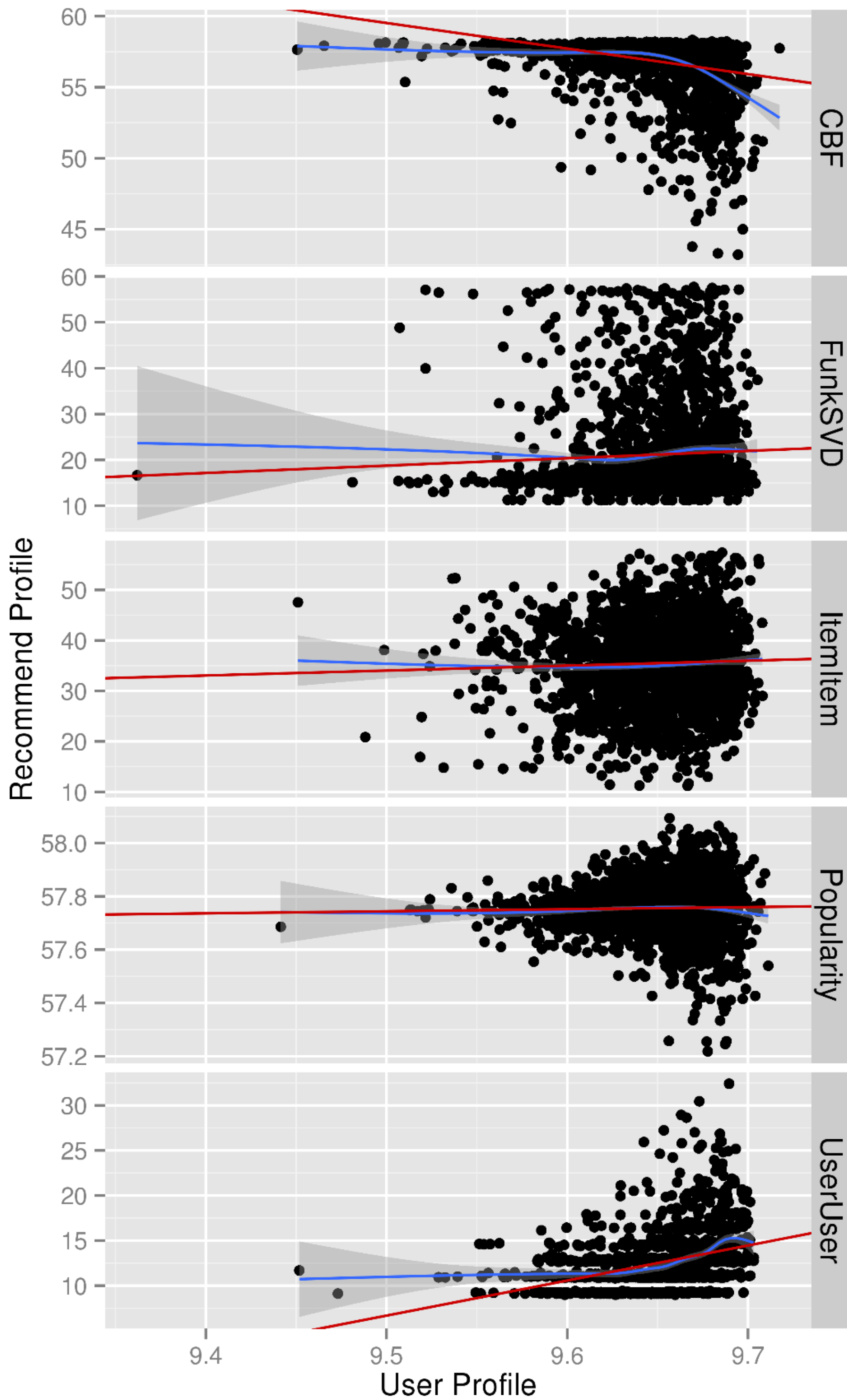

Figure 25: Recommender Profile verses User Profile for list size 25

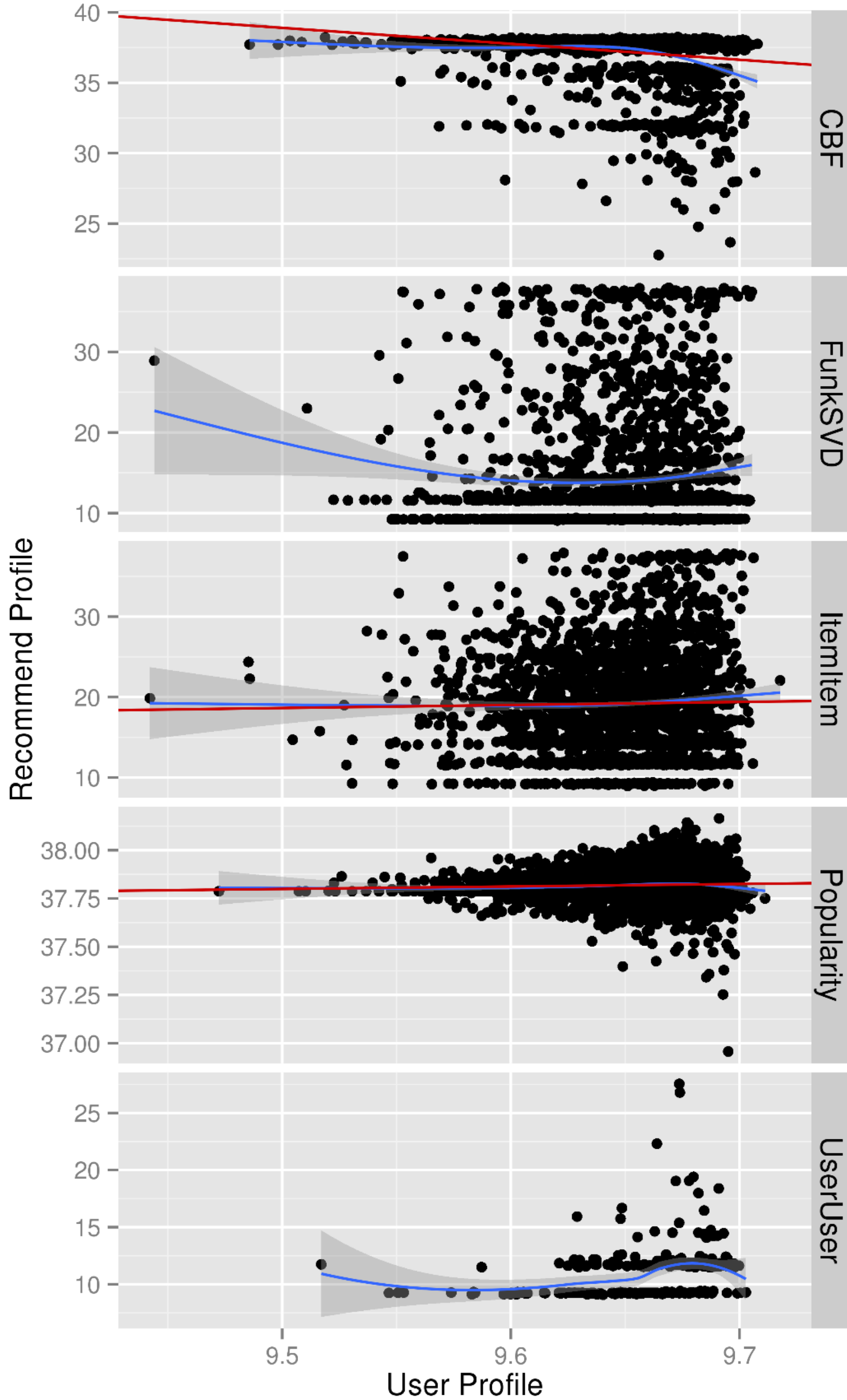

Figure 26: Recommender Profile verses User Profile for list size 10

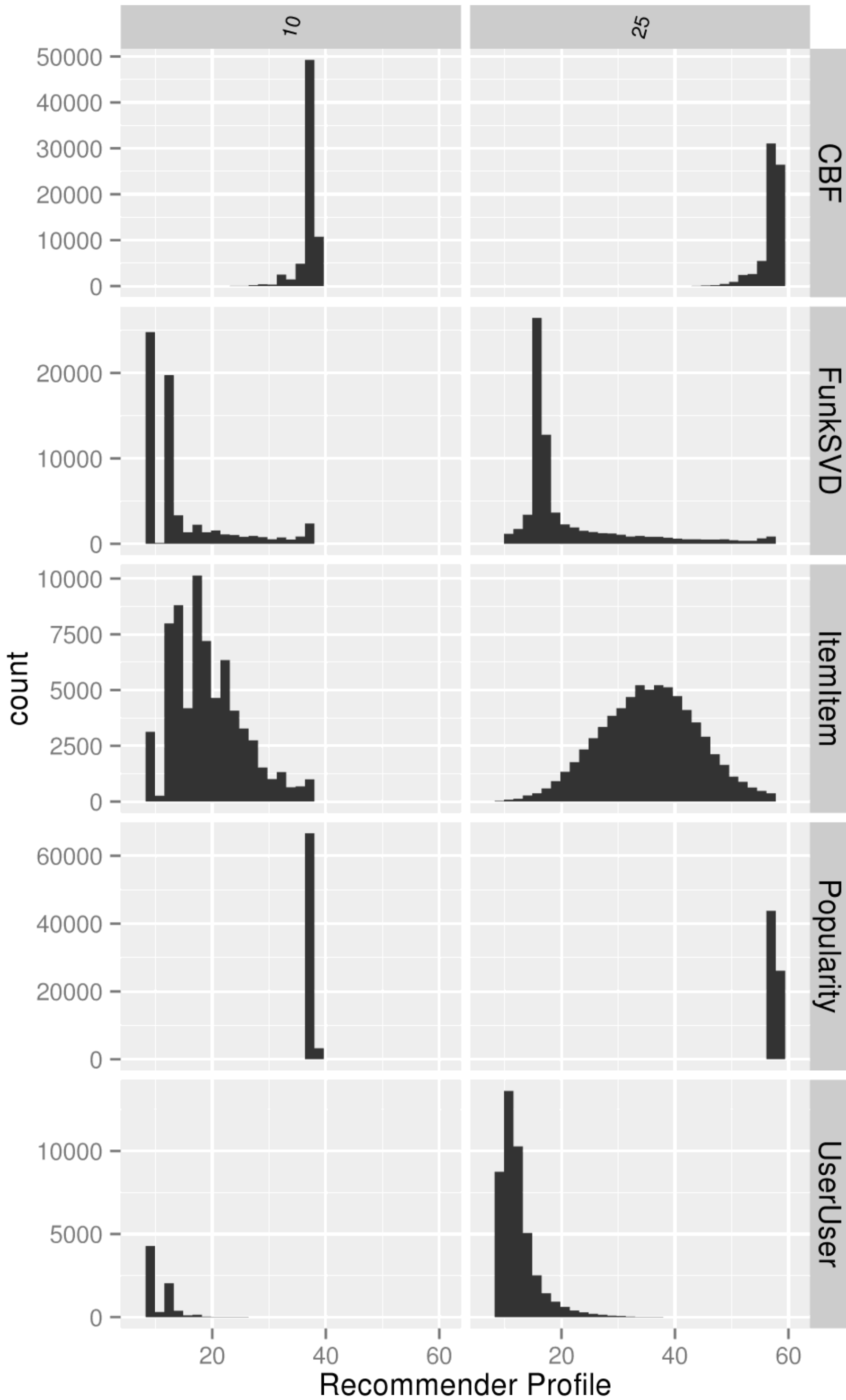

Figure 27: Histograms for Recommender Profile for recommendation list size 10 and 25

5. DISCUSSION

We used different metrics to measure the diversity bias in the users' input profile. We discovered that the algorithms we considered do not propagate very much — if any — of the users' popularity or diversity bias into their recommendations. In few cases, some of the recommender algorithms seem to be propagating a little but it is hard to quantify. Hence, it seems that recommenders may be missing a component of personalization. The recommender algorithms we have used except for popularity recommender are personalized recommenders. In an ideal case, they should be able to propagate every component of personalization. From our work, the recommenders are missing the aspect of recommending items based on the users' choice being more biased towards popularity or diversity.

However, this is not necessarily a problem, and may make recommenders resilient to other types of problems. One of the concerns in recommender systems is that users being trapped instead of exposing themselves to the wide collection of data that is available. In (Nguyen et al. 2014), the filter bubble effect was studied to describe the effect of recommender in narrowing the content that's recommended to the users. The authors raised a question if the recommendations are narrowed over the time as the users consume the recommendations due to personalization of the algorithms. On contrary, we see that irrespective of the users' bias over diversity, the recommenders do not propagate the bias to the recommendation lists. This raises the question of "How can we justify that the users are trapped by using recommender systems if the recommenders do not propagate the users' bias?"

6. CONCLUSIONS AND FUTURE WORK

We have analyzed how recommender algorithms propagated the users' input profile bias in their recommendations. This work discovered that the personalized recommenders are missing a component of personalization in terms of diversity and popularity. So far, we have observed that accuracy metrics have no impact on the users' profile bias. The recommenders tended, to varying degrees, to produce recommendations that are less diverse than the user's historical profile. More research can be done by resampling the dataset to produce dataset with different biases. This way we can have a better insight on how the recommenders produce recommendations for different biases.

Our work is limited to only one domain, which is movies. In future, this analysis can be expanded to other datasets like music and books to understand the general trend in the behavior of the recommenders. Further experiments can help us understand if the results obtained are limited to only this domain.

APPENDIX SECTION

Table 1: Coefficient values for different algorithms in linear regression model for list size 25 for mean popularity metric

Algorithm	Recommender Popularity			RMSE			MAP		
	Coeff	p-value	Adj R^2	Coeff	p-value	Adj R^2	Coeff	p-value	Adj R^2
Item-Item	5.436e-03	0.7437	-1.278e-05	5.220e-06	0.0258	5.681e-05	-8.537e-08	0.2194	7.279e-06
User-User	-3.412e-01	< 2.2e-16	0.08644	2.624e-05	< 2.2e-16	0.001639	1.381e-07	< 2.2e-16	0.004146
Funk SVD	-0.54174	< 2.2e-16	0.01746	7.464e-06	0.001223	0.0001353	3.194e-07	3.839e-09	0.0004822
CBF	8.855e-01	< 2.2e-16	0.1312	-2.230e-05	9.517e-11	0.0005895	-4.308e-06	< 2.2e-16	0.02643
Popularity	7.347e-03	< 2.2e-16	0.03011				-2.790e-05	< 2.2e-16	0.03205

Table 2: Coefficient values for different algorithms in linear regression model for list size 10 for mean popularity metric

Algorithm	Recommender Popularity			RMSE			MAP		
	Coefficient	p-value	Adj R^2	Coefficient	p-value	Adj R^2	Coefficient	p-value	Adj R^2
Item-Item	-0.09890	2.243e-09	0.0004972	5.220e-06	0.0258	5.681e-05	-1.106e-07	0.04005	4.602e-05
Funk SVD	-0.49462	< 2.2e-16	0.01194	7.464e-06	0.001223	0.001639	1.146e-07	0.02146	6.138e-05
User-User	-1.254e-01	< 2.2e-16	0.03616	2.624e-05	< 2.2e-16	0.0001353	6.306e-08	< 2.2e-16	0.00205
CBF	8.080e-01	< 2.2e-16	0.1099	-2.230e-05	9.517e-11	0.0005895	-4.054e-06	< 2.2e-16	0.02277
Popularity	0.004393	< 2.2e-16	0.02269				-2.205e-05	< 2.2e-16	0.02739

Table 3: Mean of the differences and p-values in paired t-tests for mean popularity metric

	List size 25		List size 10	
Algorithm	Recommender Popularity		Recommender Popularity	
	Mean of the differences	p-value	Mean of the differences	p-value
Item-Item	5708.75	<2.2e-16	7401.288	<2.2e-16
Funk SVD	8630.561	<2.2e-16	8509.705	<2.2e-16
User-User	9577.183	<2.2e-16	9746.65	<2.2e-16
CBF	1393.285	<2.2e-16	1292.082	<2.2e-16
Popularity	-626.5807	<2.2e-16	-638.9796	<2.2e-16

Table 4: Coefficient values for different algorithms in linear regression model for list size 25 for ILS metric

Algorithm	Recommender Diversity			RMSE			MAP		
	Coeff	p-value	Adj R^2	Coeff	p-value	Adj R^2	Coeff	p-value	Adj R^2
Item-Item	0.065086	<2.2e-16	0.001895	0.06033	0.05908	3.668e-05	0.005846	7.124e-10	0.0005292
User-User	-0.194970	<2.2e-16	0.002733	0.07724	0.02047	6.256e-05	-1.547e-03	2.2e-16	0.002789
Funk SVD	0.056774	<2.2e-16	0.001778	0.07424	0.01847	6.512e-05	-0.001192	0.1071	2.285e-05
CBF	-0.153197	<2.2e-16	0.02679	-0.37305	1.123e-15	0.0009104	-0.02816	< 2.2e-16	0.006053
Popularity	-0.088523	<2.2e-16	0.01264				-0.162411	< 2.2e-16	0.005819

Table 5: Coefficient values for different algorithms in linear regression model for list size 10 for ILS metric

Algorithm	Recommender Diversity			RMSE			MAP		
	Coeff	p-value	Adj R^2	Coeff	p-value	Adj R^2	Coeff	p-value	Adj R^2
Item-Item	0.003295	0.6424	-1.191e-05	0.06033	0.05908	3.668e-05	-9.602e-06	0.9896	-1.431e-05
User-User	0.28108	0.0001	0.004227	0.07724	0.02047	6.256e-05	-5.142e-04	7.178e-13	0.0007224
Funk SVD	0.095354	5.254e-13	0.001294	0.07424	0.01847	6.512e-05	-0.001230	0.07039	3.254e-05
CBF	-0.156831	< 2.2e-16	0.01434	-0.37305	1.123e-15	0.0009104	-0.030164	< 2.2e-16	0.006753
Popularity	-0.000218	0.9584	-1.427e-05				-0.141551	< 2.2e-16	0.006048

Table 6: Mean of the differences and p-values in paired t-tests for ILS metric

	List size 25		List size 10	
Algorithm	Recommender Popularity		Recommender Popularity	
	Mean of the differences	p-value	Mean of the differences	p-value
Item-Item	0.118700	<2.2e-16	0.127564	<2.2e-16
Funk SVD	0.117621	<2.2e-16	0.157229	<2.2e-16
User-User	0.088737	<2.2e-16	0.112728	<2.2e-16
CBF	-0.01800	<2.2e-16	-0.013986	<2.2e-16
Popularity	0.039454	<2.2e-16	0.075617	<2.2e-16

Table 7: Coefficient values for different algorithms in linear regression model for list size 25 for average ILS metric

Algorithm	Recommender Diversity			RMSE			MAP		
	Coeff	p-value	Adj R^2	Coeff	p-value	Adj R^2	Coeff	p-value	Adj R^2
Item-Item	0.034685	2.102e-10	0.000563	5.220e-06	0.0258	5.681e-05	0.0058465	7.124e-10	0.0005292
User-User	-0.13219	7.194e-11	0.001054	2.624e-05	< 2.2e-16	0.001639	-1.547e-03	< 2.2e-16	0.002789
Funk SVD	0.081605	< 2.2e-16	0.002329	7.464e-06	0.001223	0.0001353	-0.0011929	0.1071	2.285e-05
CBF	-0.152277	< 2.2e-16	0.01778	-2.230e-05	9.517e-11	0.0005895	-0.0281690	< 2.2e-16	0.006053
Popularity	-0.048797	< 2.2e-16	0.003348				-0.162411	< 2.2e-16	0.005819

Table 8: Coefficient values for different algorithms in linear regression model for list size 10 for average ILS metric

Algorithm	Recommender Diversity			RMSE			MAP		
	Coeff	p-value	Adj R^2	Coeff	p-value	Adj R^2	Coeff	p-value	Adj R^2
Item-Item	0.014000	0.06631	3.602e-05	0.06033	0.05908	5.681e-05	-9.602e-06	0.9896	-1.431e-05
User-User	0.30575	0.0001092	0.004583	0.07724	0.02047	0.001639	-5.142e-04	7.178e-13	0.0007224
Funk SVD	0.087004	4.244e-10	0.0009625	0.07424	0.01847	0.0001353	-0.0012307	0.07039	3.254e-05
CBF	-0.149665	< 2.2e-16	0.008085	-0.37305	1.123e-15	0.0005895	-0.0301640	< 2.2e-16	0.006753
Popularity	-0.02972	2.682e-11	0.0006209				-0.141551	< 2.2e-16	0.006048

Table 9: Coefficient values for different algorithms in linear regression model for list size 25 for entropy metric

Algorithm	Recommender Popularity			RMSE			MAP		
	Coeff	p-value	Adj R^2	Coeff	p-value	Adj R^2	Coeff	p-value	Adj R^2
Item-Item	0.116568	2.2e-16	0.00449	-1.55269	< 2.2e-16	0.02382	0.09952	3.108e-05	0.000234
User-User	0.95883	< 2.2e-16	0.06595	-1.56101	< 2.2e-16	0.02215	0.44528	< 2.2e-16	0.01026
Funk SVD	0.137474	< 2.2e-16	0.004053	-1.60980	< 2.2e-16	0.02632	0.22078	< 2.2e-16	0.001609
CBF	0.002244	0.4522	-6.224e-06	-1.64181	< 2.2e-16	0.01256	-0.46591	< 2.2e-16	0.009255
Popularity	-0.014315	< 2.2e-16	0.00170				-0.62641	< 2.2e-16	0.005517

Table 10: Coefficient values for different algorithms in linear regression model for list size 10 for entropy metric

Algorithm	Recommender Popularity			RMSE			MAP		
	Coeff	p-value	Adj R^2	Coeff	p-value	Adj R^2	Coeff	p-value	Adj R^2
Item-Item	0.07694	1.62e-12	0.0007087	-1.55269	< 2.2e-16	0.02382	0.12580	1.062e-10	0.0005823
User-User	0.98391	< 2.2e-16	0.1145	-1.56101	< 2.2e-16	0.02215	0.37392	< 2.2e-16	0.01173
Funk SVD	0.06595	1.134e-06	0.0003531	-1.60980	< 2.2e-16	0.02632	0.17841	< 2.2e-16	0.001602
CBF	-0.036168	< 2.2e-16	0.001194	-1.64181	< 2.2e-16	0.01256	-0.7036	< 2.2e-16	0.03132
Popularity	0.08741	< 2.2e-16	0.02329				-0.48837	< 2.2e-16	0.003389

Table 11: Mean of the differences and p-values in paired t-tests for entropy metric

	List size 25		List size 10	
Algorithm	Recommender Popularity		Recommender Popularity	
	Mean of the differences	p-value	Mean of the differences	p-value
Item-Item	-0.1105027	<2.2e-16	-0.2005839	<2.2e-16
Funk SVD	-0.1915025	<2.2e-16	-0.3053631	<2.2e-16
User-User	-0.2884622	<2.2e-16	-0.3576204	<2.2e-16
CBF	-0.0067576	<2.2e-16	-0.04905952	<2.2e-16
Popularity	-0.0058066	<2.2e-16	-0.04817786	<2.2e-16

Table 12: Coefficient values for different algorithms in linear regression model for list size 25 for discounted entropy metric

Algorithm	Recommender Popularity			RMSE			MAP		
	Coeff	p-value	Adj R^2	Coeff	p-value	Adj R^2	Coeff	p-value	Adj R^2
Item-Item	9.758	2.2e-16	0.001234	-1.55346	< 2.2e-16	0.02386	0.003886	0.000577	0.0001552
User-User	38.9081	< 2.2e-16	0.1302	-1.55802	< 2.2e-16	0.02207	0.0025653	< 2.2e-16	0.005426
Funk SVD	16.07	< 2.2e-16	0.002345	-1.614	< 2.2e-16	0.02648	0.0040997	3.255e-06	0.0002956
CBF	-17.9788	< 2.2e-16	0.08275	1.64181	< 2.2e-16	0.01256	-0.038194	< 2.2e-16	0.007862
Popularity	0.07808	7.165e-11	0.0005934				-0.024048	0.01201	7.598e-05

Table 13: Coefficient values for different algorithms in linear regression model for list size 10 for discounted entropy metric

Algorithm	Recommender Popularity			RMSE			MAP		
	Coeff	p-value	Adj R^2	Coeff	p-value	Adj R^2	Coeff	p-value	Adj R^2
Item-Item	3.7492	9.338e-07	0.0003343	-1.55346	< 2.2e-16	0.02386	0.0013116	0.1339	1.785e-05
User-User	18.8022	< 2.2e-16	0.07836	-1.55802	< 2.2e-16	0.02207	8.423e-04	< 2.2e-16	0.001381
Funk SVD	3.4924	0.0003075	0.0001872	-1.614	< 2.2e-16	0.02648	0.0016109	0.04658	4.237e-05
CBF	-11.3351	< 2.2e-16	0.04264	1.64181	< 2.2e-16	0.01256	-0.027176	< 2.2e-16	0.003864
Popularity	0.12977	< 2.2e-16	0.002066				-0.029547	0.0003059	0.0001722

LITERATURE CITED

Auguie, Baptiste, and Anton Antonov. 2016. *gridExtra: Miscellaneous Functions for "Grid" Graphics* (version 2.2.1). https://cran.r-project.org/web/packages/gridExtra/index.html.

Bradley, Keith, and Barry Smyth. 2001. "Improving Recommendation Diversity." In *Proceedings of the Twelfth National Conference in Artificial Intelligence and Cognitive Science (AICS-01)*, 75–84. Citeseer. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.8.5232&rep=rep1&type=pdf.

Breese, John S., David Heckerman, and Carl Kadie. 1998. "Empirical Analysis of Predictive Algorithms for Collaborative Filtering." In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, 43–52. Morgan Kaufmann Publishers Inc. http://dl.acm.org.libproxy.txstate.edu/citation.cfm?id=2074100.

Ekstrand, Michael D., F. Maxwell Harper, Martijn C. Willemsen, and Joseph A. Konstan. 2014. "User Perception of Differences in Recommender Algorithms." In *Proceedings of the 8th ACM Conference on Recommender Systems*, 161–68. ACM. http://dl.acm.org.libproxy.txstate.edu/citation.cfm?id=2645737.

Ekstrand, Michael D., Michael Ludwig, Joseph A. Konstan, and John T. Riedl. 2011. "Rethinking the Recommender Research Ecosystem: Reproducibility, Openness, and LensKit." In *Proceedings of the Fifth ACM Conference on Recommender Systems*, 133–40. RecSys '11. New York, NY, USA: ACM. doi:10.1145/2043932.2043958.

Harper, F. Maxwell, and Joseph A. Konstan. 2015. "The MovieLens Datasets: History and Context." *ACM Trans. Interact. Intell. Syst.* 5 (4): 19:1–19:19. doi:10.1145/2827872.

Herlocker, Jonathan L., Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. "Evaluating Collaborative Filtering Recommender Systems." *ACM Transactions on Information Systems (TOIS)* 22 (1): 5–53.

Hurley, Neil, and Mi Zhang. 2011. "Novelty and Diversity in Top-N Recommendation – Analysis and Evaluation." *ACM Trans. Internet Technol.* 10 (4): 14:1–14:30. doi:10.1145/1944339.1944341.

Kamba, Tomonari, Krishna A. Bharat, and Michael C. Albers. 1995. "The Krakatoa Chronicle-an Interactive, Personalized Newspaper on the Web." https://smartech.gatech.edu/handle/1853/3572.

Knijnenburg, Bart P., Martijn C. Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. 2012. "Explaining the User Experience of Recommender Systems." *User Modeling and User-Adapted Interaction* 22 (4-5): 441–504.

Konstan, Joseph A., and John Riedl. 2012. "Recommender Systems: From Algorithms to User Experience." *User Modeling and User-Adapted Interaction* 22 (1-2): 101–23.

Lathia, Neal, Stephen Hailes, Licia Capra, and Xavier Amatriain. 2010. "Temporal Diversity in Recommender Systems." In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 210–17. ACM. http://dl.acm.org.libproxy.txstate.edu/citation.cfm?id=1835486.

Linden, G., B. Smith, and J. York. 2003. “Amazon.com Recommendations: Item-to-Item Collaborative Filtering.” *IEEE Internet Computing* 7 (1): 76–80. doi:10.1109/MIC.2003.1167344.

McNee, Sean M., John Riedl, and Joseph A. Konstan. 2006. “Being Accurate Is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems.” In *CHI’06 Extended Abstracts on Human Factors in Computing Systems*, 1097–1101. ACM. http://dl.acm.org.libproxy.txstate.edu/citation.cfm?id=1125659.

Nguyen, Tien T., Pik-Mai Hui, F. Maxwell Harper, Loren Terveen, and Joseph A. Konstan. 2014. “Exploring the Filter Bubble: The Effect of Using Recommender Systems on Content Diversity.” In *Proceedings of the 23rd International Conference on World Wide Web*, 677–86. ACM. http://dl.acm.org.libproxy.txstate.edu/citation.cfm?id=2568012.

Ozok, A. Ant, Quyin Fan, and Anthony F. Norcio. 2010. “Design Guidelines for Effective Recommender System Interfaces Based on a Usability Criteria Conceptual Model: Results from a College Student Population.” *Behaviour & Information Technology* 29 (1): 57–83.

Pérez, Fernando, and Brian E. Granger. 2007. “IPython: A System for Interactive Scientific Computing.” *Computing in Science & Engineering* 9 (3): 21–29. doi:10.1109/MCSE.2007.53.

Pu, Pearl, Li Chen, and Rong Hu. 2011. “A User-Centric Evaluation Framework for Recommender Systems.” In *Proceedings of the Fifth ACM Conference on Recommender Systems*, 157–64. ACM. http://dl.acm.org.libproxy.txstate.edu/citation.cfm?id=2043962.

R. Core Team. 2012. *A Language and Environment for Statistical Computing*. *R Foundation for Statistical Computing, Vienna, Austria*. http://www.R-project.org/.

Resnick, Paul, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. “GroupLens: An Open Architecture for Collaborative Filtering of Netnews.” In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, 175–86. CSCW ’94. New York, NY, USA: ACM. doi:10.1145/192844.192905.

Resnick, Paul, and Rahul Sami. 2007. “The Influence Limiter: Provably Manipulation-Resistant Recommender Systems.” In *Proceedings of the 2007 ACM Conference on Recommender Systems*, 25–32. ACM. http://dl.acm.org/citation.cfm?id=1297236.

Resnick, Paul, and Hal R. Varian. 1997. “Recommender Systems.” *Communications of the ACM* 40 (3): 56–58.

Shani, Guy, and Asela Gunawardana. 2011. “Evaluating Recommendation Systems.” In *Recommender Systems Handbook*, 257–97. Springer. http://link.springer.com.libproxy.txstate.edu/10.1007/978-0-387-85820-3_8.

Vargas, Saúl, and Pablo Castells. 2011. “Rank and Relevance in Novelty and Diversity Metrics for Recommender Systems.” In *Proceedings of the Fifth ACM Conference on Recommender Systems*, 109–16. ACM. http://dl.acm.org.libproxy.txstate.edu/citation.cfm?id=2043955.

Vig, Jesse, Shilad Sen, and John Riedl. 2009. “Tagsplanations: Explaining Recommendations Using Tags.” In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, 47–56. IUI ’09. New York, NY, USA: ACM. doi:10.1145/1502650.1502661.

Wickham, Hadley. 2014. *reshape2: Flexibly Reshape Data: A Reboot of the Reshape Package* (version 1.4.1). https://cran.r-project.org/web/packages/reshape2/index.html.

Wickham, Hadley. 2016. *Plyr: Tools for Splitting, Applying and Combining Data* (version 1.8.4). https://cran.r-project.org/web/packages/plyr/index.html.

Wickham, Hadley, Winston Chang, and RStudio. 2016. *ggplot2: An Implementation of the Grammar of Graphics* (version 2.1.0). https://cran.r-project.org/web/packages/ggplot2/index.html.

Wickham, Hadley, Romain Francois, and RStudio. 2016. *Dplyr: A Grammar of Data Manipulation* (version 0.5.0). https://cran.r-project.org/web/packages/dplyr/index.html.

Xiao, Bo, and Izak Benbasat. 2007. “E-Commerce Product Recommendation Agents: Use, Characteristics, and Impact.” *Mis Quarterly* 31 (1): 137–209.

Yu, Cong, Laks Lakshmanan, and Sihem Amer-Yahia. 2009. “It Takes Variety to Make a World: Diversification in Recommender Systems.” In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, 368–78. ACM. http://dl.acm.org.libproxy.txstate.edu/citation.cfm?id=1516404.

Zhang, Mi, and Neil Hurley. 2008. “Avoiding Monotony: Improving the Diversity of Recommendation Lists.” In *Proceedings of the 2008 ACM Conference on Recommender Systems*, 123–30. ACM. http://dl.acm.org.libproxy.txstate.edu/citation.cfm?id=1454030.

Zhang, Yuan Cao, Diarmuid Ó Séaghdha, Daniele Quercia, and Tamas Jambor. 2012. “Auralist: Introducing Serendipity into Music Recommendation.” In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, 13–22. ACM. http://dl.acm.org.libproxy.txstate.edu/citation.cfm?id=2124300.

Ziegler, Cai-Nicolas, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. 2005. “Improving Recommendation Lists Through Topic Diversification.” In *Proceedings of the 14th International Conference on World Wide Web*, 22–32. WWW ’05. New York, NY, USA: ACM. doi:10.1145/1060745.1060754.