Recommendation Opportunities: Improving Item Prediction Using Weighted Percentile Methods in Collaborative Filtering Systems

Panagiotis Adamopoulos
padamopo@stern.nyu.edu

Alexander Tuzhilin
atuzhili@stern.nyu.edu

Department of Information, Operations and Management Sciences
Leonard N. Stern School of Business, New York University

ABSTRACT

This paper proposes a novel method for estimating unknown ratings and *recommendation opportunities* and illustrates the practical implementation of the proposed approach by presenting a certain variation of the classical k-NN method in neighborhood-based collaborative filtering systems using weighted percentiles. We conduct an empirical study showing that the proposed method outperforms the standard user-based collaborative filtering approach by a wide margin in terms of item prediction accuracy and utility-based ranking metrics across various experimental settings. We also demonstrate that this performance improvement is not achieved at the expense of other popular performance measures, such as catalog coverage and aggregate diversity. The proposed approach can also be applied to other popular methods for rating estimation.

Categories and Subject Descriptors

H.4.m [**Information Systems Applications**]: Miscellaneous

Keywords: Weighted Percentiles; Item Accuracy; Collaborative Filtering; Recommendations; Recommender Systems

1. INTRODUCTION

Although there have been many rating estimation methods developed over the last 20 years [6], the classical user-based k-NN collaborative filtering (CF) method still remains one of the most popular and prominent methods used in the recommender systems (RSs) community.

In this paper, we propose a novel method for estimating unknown ratings and *recommendation opportunities* and illustrate the practical implementation of the proposed approach by presenting a certain variation of the classical k-NN method in which the estimation of an unknown rating of the user for an item is based not on the weighted average of the k nearest neighbors but on the weighted percentile of the ratings of these k neighbors. The key intuition behind using this weighted percentiles method, instead of weighted averages, is that high percentiles (such as in the 70% to 90% range) constitute more realistic estimates of how much a targeted user *could* possibly like the candidate item based on the experiences of his/her neighbors. Using such high percentiles is analogous to "shifting the needle" in our rating combining function from the middle of the rating distribution, as in the case of the weighted average, toward the tail on the right side of the distribution, targeting recommendations that the users will like better.

To support this claim, we conducted an empirical study and showed that the proposed percentile method outperforms the standard user-based CF approach by a very wide margin, across various experimental settings, in terms of popular item prediction accuracy and utility-based ranking metrics. Finally, we demonstrate that this performance improvement is not achieved at the expense of some other popular performance measures, such as aggregate diversity.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
RecSys'13, October 12–16, 2013, Hong Kong, China.
Copyright 2013 ACM 978-1-4503-2409-0/13/10 ...$15.00.
http://dx.doi.org/10.1145/2507157.2507229.

2. RELATED WORK

In the recommender systems literature, since the first CF systems were proposed in the mid-90's [11], [21], there have been many attempts to improve their performance [20]. For instance, [12] studies variations of rating normalization, similarity weighting and neighbor selection schemes, and [22] calculates similarities using regressions. However, there is still a long way to go in terms of satisfaction of users' actual needs [17] and many approaches that go beyond the rating prediction perspective trying to alleviate the problems pertaining to the narrow rating prediction accuracy-based focus [1] have been gaining significant attention [14].

Moving beyond this narrow focus, in the related field of data mining, [19] discusses the use of combining functions in clustering and [18] utilizes the concept of percentiles, looking for the 80 percent of the conditional spending distribution of customers, in order to identify new sales prospects. Even though this idea of percentiles was applied to clustering techniques in data mining, it has not yet been applied to the RSs problems. In this paper, we apply the concept of percentiles, proposed in the data mining community, to the CF approach and show that this method improves performance on neighborhood models in very significant ways.

3. MODEL

Under a definition of *recommendation opportunity* as how much a user *could* realistically like an item, we are looking for a high percentile (e.g. 80 percent) of the conditional distribution of the rating for the specific target user and candidate item, given all the information we have about them. Utilizing the high percentiles, we aim at recommending items that the users will remarkably like. One of the

challenges here is to get a good estimate of the conditional rating percentile for each user and item from the available data. In the next section, we illustrate the practical implementation of the proposed approach in the context of neighborhood models.

3.1 Neighborhood Models

User-based neighborhood recommendation methods predict the rating $r_{u,i}$ of user u for item i using the ratings given to i by users most similar to u, called nearest neighbors and denoted by $\mathcal{N}_i(u)$. Taking into account the fact that the neighbors can have different levels of similarity, $w_{u,v}$, and considering the k users v with the highest similarity to u (i.e. the standard user-based k-NN collaborative filtering approach), the predicted rating is:

$$\hat{r}_{u,i} = \frac{\sum_{v \in \mathcal{N}_i(u)} w_{u,v} r_{v,i}}{\sum_{v \in \mathcal{N}_i(u)} |w_{u,v}|} \quad (1)$$

However, the ratings given to item i by the nearest neighbors of user u can be combined into a single estimation using various combining (or aggregating) functions [6].

In this paper, we propose to use higher weighted percentiles as a combining function. Such a high percentile p (e.g. 70^{th}, ..., 90^{th}) of the conditional distribution of the user's rating, given all the information that we have available, characterizes how much the target user u *could* realistically like the candidate item i. Intuitively, using high percentiles is analogous to "shifting the needle" in our rating combining function from the middle of the rating distribution, as is the case with the weighted average, toward the tail on the right side of the distribution targeting recommendations that the users will like better. Formally, the percentile, denoted by $\hat{r}^p_{u,i}$, is defined such that the probability that user u would rate item i with a rating of $\hat{r}^p_{u,i}$ or less is p%. Note that both low and high ratings contribute to the estimation since they affect the rank of the values and, thus, the percentile quantity of interest.

In a typical k-NN collaborative filtering model, the information that we have available in order to estimate an unknown rating, and respectively the quantity $\hat{r}^p_{u,i}$, is the neighbors of user u, denoted by $\mathcal{N}_i(u)$, the similarity levels of these neighbors $w_{\mathcal{N}_i(u)} := (w_{u,v} : v \in \mathcal{N}_i(u))$, and the corresponding ratings $r_{\mathcal{N}(u),i} := (r_{v,i} : v \in \mathcal{N}_i(u))$. As an example of the proposed method and its differences from the classical approaches, consider the neighborhood $\mathcal{N}(u)$ of size 4 with similarity weights $w_{\mathcal{N}(u)} = (0.2, 0.4, 0.3, 0.1)$ and items x and y with ratings $r_{\mathcal{N}(u),x} = (2, 3, 3, 4)$ and $r_{\mathcal{N}(u),y} = (2, 2, 4, 4)$, respectively. Using the standard combining function as in (1), item x would be recommended. However, using, for instance, the weighted 80^{th} percentile of the variable $r_{\mathcal{N}(u),i}$, item y would be recommended since the specific percentile for item y, denoted by $\hat{r}^{p=80}_{u,y}$, corresponds to a higher rating than item x and, thus, there is high potential that user u could realistically like item y more than x; equivalently, the probability of user u assigning a rating greater or equal to 4 is higher for item y than x.

Algorithm 1 summarizes the user-based k-nearest neighbors (k-NN) collaborative filtering approach with a general combining function and Algorithm 2 shows a procedure to estimate a weighted percentile $\hat{r}^p_{u,i}$ (i.e. the proposed combining function), where the values $r_{\mathcal{N}(u),i}$ are the ratings given to candidate item i by neighbors $\mathcal{N}_i(u)$, the k users most similar to target user u, and the weights $w_{\mathcal{N}_i(u)}$ are the corresponding similarity levels of neighbors to user u.

ALGORITHM 1: k-NN Recommendation Algorithm

Input: User-Item Rating matrix R
Output: Recommendation lists of size l

k: Number of users in the neighborhood of user u, $\mathcal{N}_i(u)$

for *each user* u **do**
  Find the k users most similar to user u, $\mathcal{N}_i(u)$;
  for *each item* i **do**
    Combine ratings given to item i by neighbors $\mathcal{N}_i(u)$;
  end
  Recommend to user u the top-l items having the highest predicted rating $\hat{r}_{u,i}$;
end

ALGORITHM 2: Weighted Percentile Estimation

Input: Values $v_1, \ldots, v_n$, Weights $w_1, \ldots, w_n$, and p percentile to be estimated.
Output: p-th weighted percentile of ordered values $v_1, \ldots, v_n$

Order values $v_1, \ldots, v_n$ from least to greatest;
Rearrange weights $w_1, \ldots, w_n$ based on ordered values;
Calculate the percent rank for p based on weights $w_1, \ldots, w_n$;
Use linear interpolation between the two nearest ranks;

4. EXPERIMENTAL SETTINGS

To empirically validate the proposed method and evaluate the generated recommendations, we conduct a large number of experiments on "real-world" data sets and compare our results to the k-NN CF approach, which has been found to perform well also in terms of other performance measures besides the classical accuracy metrics [7], [13], [3, 4].

4.1 Data sets

The data sets that we used are the RecSys HetRec 2011 MovieLens data set [8] and the BookCrossing data set [23].

The RecSys HetRec 2011 MovieLens (ML) data set contains personal ratings and tags about movies and consists of 855,598 ratings from 2,113 users on 10,197 items.

The BookCrossing (BX) data set is described by [23] and gathered from Bookcrossing.com, a social networking site founded to encourage the exchange of books. Following Ziegler et al. [23] and owing to the extreme sparsity of the data, we decided to condense the data set in order to obtain more meaningful results from collaborative filtering algorithms. Hence, we discarded as in [23] all books for which we were not able to find any information, along with all the ratings referring to them. Next, we also removed book titles with fewer than 4 ratings and community members with fewer than 8 ratings each. The dimensions of the resulting data set were considerably more moderate, featuring 8,824 users, 7,818 books, and 107,367 explicit ratings.

4.2 Experimental settings

Using the ML and BX data sets, we conducted a large number of experiments and compared our method against the standard user-based k nearest neighbors collaborative filtering approach. In order to test the proposed approach of weighted percentiles under various experimental settings, we used 2 data sets, 6 different sizes of neighborhoods ($k \in \{30, 40, \ldots, 80\}$), 9 different percentiles as combining functions ($p \in \{10, 20, \ldots, 90\}$), and generated recommendation

Table 1: Item Prediction Accuracy (F_1 score$*10^2$).

Data Set	Method		Recommendation List Size 3	5	10	30	50	100
ML	k-NN		0.0026	0.0039	0.0078	0.0164	0.3575	0.3362
	percentile	60^{th}	0.0902	0.1533	0.2975	0.7208	0.8094	0.8611
		70^{th}	0.1784	0.2907	0.5309	1.2440	1.8051	2.3565
		80^{th}	0.0993	0.1575	0.2848	0.6966	0.9525	1.2930
		90^{th}	0.0456	0.0854	0.1848	0.4552	0.6316	0.9344
BX	k-NN		0.1606	0.1876	0.2415	0.2882	0.2899	0.2807
	percentile	60^{th}	0.1743	0.2396	0.3149	0.3654	0.3716	0.3526
		70^{th}	0.1864	0.2418	0.3184	0.3751	0.3841	0.3823
		80^{th}	0.2130	0.2590	0.3654	0.4419	0.4361	0.4256
		90^{th}	0.2126	0.3065	0.3772	0.4592	0.4795	0.4728

lists of 13 different sizes ($l \in \{1, 3, 5, 10, 20, \ldots, 100\}$), resulting in 1,404 experiments in total.

For the computation of the weighted percentiles, we used the `gmisclib` [10] scientific library. Besides, we used Pearson correlation to measure similarity. Finally, we used a holdout validation scheme in all of our experiments with 80/20 splits of data to the training/test part in order to avoid overfitting.

5. RESULTS

The aim of this study is to demonstrate that the proposed method is indeed effectively increasing the classical item prediction accuracy measures and performs well in terms of other popular performance measures, such as catalog coverage, by a comparative analysis of our method and the standard k-NN algorithm, in different experimental settings.[1]

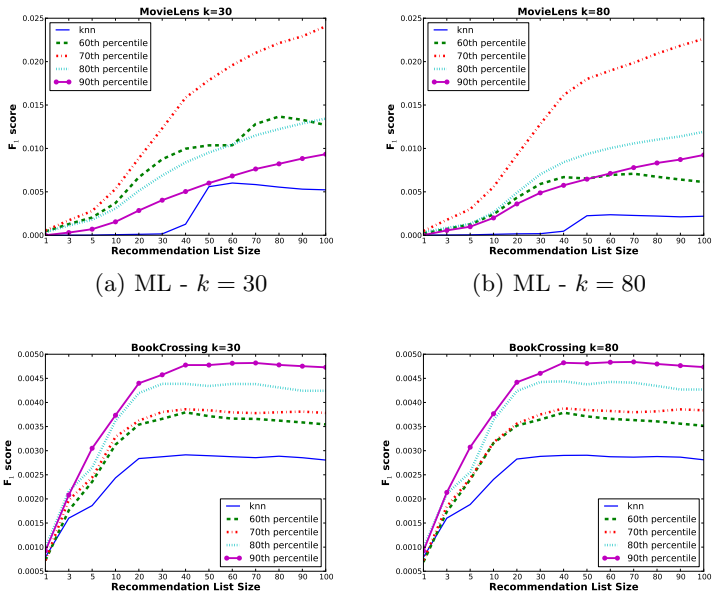
5.1 Comparison of Item Prediction

The goal in this section is to compare our method with the standard baseline methods in terms of traditional metrics for *item prediction*, such as precision, recall, and F_1 score. Table 1 presents the results obtained by applying our method to the MovieLens and BookCrossing data sets. The values reported are computed as the average performance over the six neighborhood sizes using the F_1 score for recommendation lists of size $l \in \{3, 5, 10, 30, 50, 100\}$. Respectively, Fig. 1 illustrates the average performance for neighborhoods of size $k \in \{30, 80\}$ and lists of size $l \in \{1, 3, 5, 10, 20, \ldots, 100\}$.

Table 1 and Fig. 1 demonstrate that the proposed method outperforms the k-NN method by a wide margin. In particular, for both data sets, accuracy was improved in all the experiments using high percentiles. Besides, as we can observe, the increase in performance is larger for recommendation lists of larger size. For the ML data set the maximum F_1 score was achieved using the 70^{th} percentile (0.024) whereas for BX the maximum was 0.0048 using the 90^{th} percentile.

To determine statistical significance, we have tested the null hypothesis that the performance of each of the methods is the same using the Friedman test. Based on the results, we reject the null hypothesis with $p < 0.0001$. Performing post hoc analysis on Friedman's Test results, the differences between the *Baseline* k-NN and each one of the experimental settings are statistically significant. Fig. 2 presents the box-and-whisker diagrams displaying the aforementioned differences among the various methods.

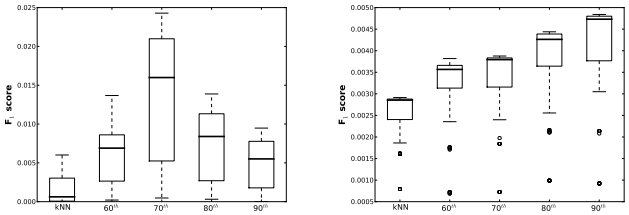
[1]Similar results were also obtained using the k users with the highest similarity to the target user u, $\mathcal{N}(u)$, independently of whether they rated the specific candidate item i. For each metric, only the most interesting dimensions are discussed. Finally, results for low percentiles are not presented, since they constantly underperform the experiments using the high percentiles.



(a) ML - $k = 30$ (b) ML - $k = 80$

(c) BX - $k = 30$ (d) BX - $k = 80$

Figure 1: Prediction Accuracy (F_1 score) for the (a), (b) MovieLens (ML) and (c), (d) BookCrossing (BX) data sets.



(a) MovieLens data set (b) BookCrossing data set

Figure 2: Post hoc analysis for Friedman's Test of Item Prediction Accuracy (F_1 score) for both data sets.

Similar results were also obtained using standard utility-based ranking metrics, such as the normalized discounted cumulative gain (nDCG) and mean reciprocal rank (MRR).

5.2 Beyond Accuracy

In this section we investigate the effect of the proposed method on coverage and aggregate diversity, two important metrics for RSs [20], that go beyond the classical perspective of rating prediction accuracy [1]. The results obtained using the *catalog coverage* metric [9] (i.e. the percentage of items in the catalog that are ever recommended to users) are equivalent to those using the *diversity-in-top-N* metric for aggregate diversity [5]; henceforth, only results on coverage are presented. Table 2 presents the results obtained by applying our method to the ML and BX data sets. The values reported are computed as the average catalog coverage over six neighborhood sizes ($k \in \{30, 40, \ldots, 80\}$) for recommendation lists of size $l = \{3, 5, 10, 30, 50, 100\}$.

Table 2 demonstrates that the proposed method performs at least as well as, and is some cases even better than, the

Table 2: Catalog Coverage Performance.

Data Set	Method	Recommendation List Size 3	5	10	30	50	100
ML	k-NN	0.50%	0.57%	0.68%	0.92%	2.80%	3.98%
	percentile **60th**	1.17%	1.29%	1.50%	1.92%	2.30%	3.77%
	percentile **70th**	2.62%	2.88%	3.24%	3.89%	4.29%	5.28%
	percentile **80th**	5.99%	6.46%	6.98%	8.08%	8.63%	9.77%
	percentile **90th**	11.31%	13.29%	15.02%	16.94%	17.93%	19.32%
BX	k-NN	45.45%	54.34%	65.52%	84.14%	90.50%	95.36%
	percentile **60th**	45.44%	54.04%	65.52%	83.85%	90.35%	95.16%
	percentile **70th**	44.85%	53.34%	64.84%	84.02%	90.13%	95.09%
	percentile **80th**	46.47%	54.66%	65.90%	84.26%	90.39%	95.16%
	percentile **90th**	46.33%	54.58%	66.04%	84.28%	90.44%	95.09%

standard user-based k-NN method. In particular, for the ML data set, where the *Baseline* k-NN results in low coverage, performance is increased on average by 643.77%, with the 90$^{\text{th}}$ percentile exhibiting the highest coverage. For the BX data, where the *Baseline* k-NN results in high coverage because of the specifics of the particular data set and the larger number of users, the performance is on average the same (+0.00). In terms of statistical significance, using the Friedman test and performing post hoc analysis, for the ML data set the differences among the standard user-based k-NN method and all the experimental settings are statistically significant ($p < 0.005$). For the BX data set, only the differences between the 70$^{\text{th}}$ percentile and the remaining experimental settings are statistically significant.

The generated recommendation lists can also be evaluated for the inequality across items using the Gini coefficient. In particular, for the ML and BX data sets the Gini coefficient was on average improved by 2.58% and 0.27%, respectively. As we can conclude, in the recommendation lists generated from the proposed method, the number of times an item is recommended is more equally distributed.

In summary, we demonstrated that the proposed method outperforms the standard user-based k-NN algorithm by a wide margin in terms of item prediction accuracy and utility-based ranking metrics and performs at least as well as, and in some cases even better than, the standard baseline method in terms of some other popular performance measures.

6. DISCUSSION AND CONCLUSIONS

In this paper, we present a novel method for estimating unknown ratings and *recommendation opportunities* based on weighted percentiles. We illustrate the practical implementation of the proposed approach in the context of neighborhood models adapting the classical k-nearest neighbors method. In addition, we conduct an empirical study showing that the proposed method outperforms the standard user-based collaborative filtering approach by a wide margin in terms of item prediction accuracy and utility-based ranking measures, such as the F-measure and normalized discounted cumulative gain, across various experimental settings. We also demonstrate that this performance improvement is not achieved at the expense of some other popular performance measures, such as aggregate diversity.

Moreover, apart from the user-based and item-based k-NN collaborative filtering approaches, other popular methods that can be easily extended, with the use of quantile regression, in order to allow us both to build models that predict high percentiles and to evaluate them with regard to the goal of predicting percentiles of estimated ratings, include content-based methods, and Matrix Factorization [16].

Nevertheless, the proposed approach should be tested using various rating normalization and similarity weighting schemes [15] as well as different distance metrics [12].

As a part of the future work, we would like to conduct live experiments with real users in an on-line retail setting as well as in a platform for massive open on-line courses [2]. Also, we will study the impact of the proposed method on novelty, serendipity, and unexpectedness [3, 4] of RSs.

7. REFERENCES

[1] P. Adamopoulos. Beyond Rating Prediction Accuracy: On New Perspectives in Recommender Systems. In *Proceedings of RecSys '13*. ACM, 2013.

[2] P. Adamopoulos. What Makes a Great MOOC? An Interdisciplinary Analysis of Student Retention in Online Courses. In *Proceedings of ICIS*, 2013.

[3] P. Adamopoulos and A. Tuzhilin. On Unexpectedness in Recommender Systems: Or How to Expect the Unexpected. In *DiveRS 2011*, RecSys '11. ACM, 2011.

[4] P. Adamopoulos and A. Tuzhilin. On Unexpectedness in Recommender Systems: Or How to Better Expect the Unexpected. *Working Paper: CBA-13-03, New York University*, 2013. http://ssrn.com/abstract=2282999.

[5] G. Adomavicius and Y. Kwon. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Trans. on Knowl. and Data Eng.*, 24(5):896 –911, 2012.

[6] G. Adomavicius and A. Tuzhilin. Toward the next generation of Recommender Systems: A Survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, June 2005.

[7] R. Burke. Hybrid recommender systems: Survey and experiments. *User Model. User-Adapt. Interact.*, 12(4), 2002.

[8] I. Cantador, P. Brusilovsky, and T. Kuflik. In *Proceedings of HetRec 2011*, RecSys '11, New York, NY, USA, 2011. ACM.

[9] M. Ge, C. Delgado-Battenfeld, and D. Jannach. Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *Proceedings of RecSys '10*. ACM, 2010.

[10] gmisclib, Scientific Library, 2013. http://kochanski.org/gpk/code/speechresearch/gmisclib/.

[11] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[12] J. Herlocker, J. A. Konstan, and J. Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information retrieval*, 5(4):287–310, 2002.

[13] D. Jannach, L. Lerche, F. Gedikli, and G. Bonnin. What recommenders recommend–an analysis of accuracy, popularity, and sales diversity effects. In *UMAP '13*. Springer, 2013.

[14] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich. *Recommender systems: an introduction*. Cambridge University Press, 2010.

[15] R. Jin, J. Y. Chai, and L. Si. An automatic weighting scheme for collaborative filtering. In *Proceedings of SIGIR '04*, 2004.

[16] A. Karatzoglou and M. Weimer. Quantile matrix factorization for collaborative filtering. In *Proceedings of EC-Web '10*, 2010.

[17] J. A. Konstan and J. T. Riedl. Recommender systems: from algorithms to user experience. *User Modeling and User-Adapted Interaction*, 22:101–123, 2012.

[18] R. Lawrence, C. Perlich, S. Rosset, J. Arroyo, Callahan, et al. Analytics-driven solutions for customer targeting and sales-force allocation. *IBM Systems Journal*, 46(4), 2007.

[19] F. Provost and T. Fawcett. *Data Science for Business: What you need to know about data mining and data-analytic thinking*. O'Reilly Media, 2013.

[20] F. Ricci and B. Shapira. *Recommender systems handbook*. Springer, 2011.

[21] E. Rich. User modeling via stereotypes. *Cognitive science*, 3(4):329–354, 1979.

[22] A. Töscher, M. Jahrer, and R. Legenstein. Improved neighborhood-based algorithms for large-scale recommender systems. In *Proceedings of Netflix '08*. ACM, 2008.

[23] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *Proceedings of WWW '05*. ACM, 2005.