Analyzing Weighting Schemes in Collaborative Filtering: Cold Start, Post Cold Start and Power Users

Alan Said
DAI-Lab
Technische Universität Berlin
alan@dai-lab.de

Brijnesh J. Jain
DAI-Lab
Technische Universität Berlin
jain@dai-lab.de

Sahin Albayrak
DAI-Lab
Technische Universität Berlin
sahin@dai-lab.de

ABSTRACT

Collaborative filtering recommender systems provide their users with relevant items based on information from other similar users. Popular collaborative filtering approaches such as Pearson correlation coefficient and cosine similarity, compute the similarity between users based on the set of their co-rated items. However, similarities are commonly computed without taking the popularity of the set of two users' co-rated items into consideration, e.g. an item rated by very many users should have less impact on the similarity measure, and analogously an item rated by few should have a larger impact on the similarity score of two users. In this paper, we investigate the effects of common weighting schemes on different types of users, i.e. new users with few ratings (so-called cold start users), post cold start users, and power users. Empirical studies over two datasets have shown in which of these cases weighting schemes are beneficial in terms of recommendation quality.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval - Information filtering, Retrieval models, Search process, Selection process

General Terms

Algorithms, Design, Experimentation, Human Factors

Keywords

recommender systems, similarity metrics, collaborative filtering, item weighting scheme, cold start, diversity, user modeling, popularity

1. INTRODUCTION

Collaborative filtering (CF) has been used in recommender systems in the online world for almost two decades. Independent of domain, they have shown to perform better than their content-based, popularity-based and other counterparts [5].

CF calculates the relevance of an item for a user based on other users' rating information on items co-rated by the group and the user. CF approaches are commonly categorized as either model-based or memory-based [2]. In this work we focus on the latter, which creates item predictions for a user by finding users similar to that user (in terms of co-rated items) in a training model, a neighborhood. Using information from the neighborhood, it predicts items which should be of interest and are not previously rated by the user. Memory-based, or neighborhood-based, approaches commonly use measures such as the Pearson correlation coefficient or cosine similarity to create the neighborhoods.

Model-based approaches treat CF as a classification problem and provide item recommendations by initially creating a probabilistic model of user ratings using algorithms such as Bayesian networks, clustering, etc. where users are grouped into the same classes based on their rating history. Following this, the conditional probability of a user being in a rating class for a specific item is calculated.

With the growth of the Web, the user base increases as well and results in massive amounts of item-user interaction data being created. In most single domain systems (i.e. movies, music, URLs etc.), the vast majority of ratings or interactions are performed on a small percentage of popular items [13] (also shown in Section 3.2, Fig. 3). As most common similarity measures (e.g. cosine, Pearson correlation coefficient) base their similarities on those items which are co-rated by two users, our assumption is that the similarities become skewed towards the most popular items. This results in "leaving out" those items that might define a user more than the popular items do, especially during a post cold start phase where the user has more than few ratings (i.e. more than the first initial dozens [10]) and can be profiled in a direction not pointing straight to the most popular items.

In this paper, we look at other means of measuring the similarity of users, in order to early identify the items which (i) users are interested in, and (ii) tell the systems more about users in order to early be able to profile them - and hence be able to produce better recommendations from an early point in time. We evaluate different weighting schemes for popular items using two datasets in a regular scenario for all users no matter the amount of ratings per user, and in three different phases of a user; cold start (few ratings), post cold start (more than few ratings), and when the user has become a power user (many ratings).

SAC'12 March 26-30, 2012, Riva del Garda, Italy.

We show that weighting schemes have different effects on the recommendation quality depending on the structure of the data, from none, to improving precision values by in excess of 20%.

2. RELATED WORK

The underlying assumption of feature weighting is that some items may provide more predictive information than others. In particular those items which are most popular are considered as less informative. To address this issue Breese et al. [2] used the inverse user frequency as weights to devalue the contribution of popular items in the Pearson correlation coefficient. Herlocker et al. [3] modified the Pearson correlation coefficient by incorporating an item-variance factor. In doing so, items with low variance, such as commonly liked items, are devalued.

More sophisticated approaches to feature weighting have also been explored. An information-theoretic approach for item weighting based on mutual information has been proposed by Yu et al. [14]. Jin et al. [6], learn the item weights from the ratings by maximizing the average similarity between users. In addition, the weighting scheme was empirically validated not only for the Pearson correlation coefficient but also for other configurations of memory- and model-based approaches. Due to contradicting empirical results, Baltrunas [1] conducted an empirical comparison of different weighting schemes. In his work, the author suggested a feature weighting based on a singular value decomposition in conjunction with the Pearson correlation.

Luo et al. [8] use feature weighting as one of several ingredients to model the relationship of users using local and global user similarity. By assuming that the ratings obey a Laplacian distribution, they construct so-called surprisal-based vector similarities. The term surprisal refers to quantities of information contained in the ratings in order to express the relationship between any two users.

Symeonidis et al. [12] compare how weights on different features affect the user profile in terms of recommendation accuracy.

All of the above listed approaches do not differentiate between users in terms of their logged activity, i.e. number of ratings in our case. The work presented in this paper takes this into consideration looking at cold start users, post cold start users and power users.

3. ANALYSIS AND PROBLEM SETTING

Our problem, that popular items tend to get higher importance in similarity calculations, is directly related to how users interact with items. In the following sections we describe the characteristics of two datasets and describe how similarity metrics could be altered in order to compensate for popularity-induced bias.

3.1 Data Analysis

We analyzed two popular recommendation datasets in order to identify how the actual inter-user similarities could be calculated, and how common measures would perform in a recommendation scenario.

The datasets chosen were the Movielens 10M100K dataset[1] which contains 10 million movie ratings by 70 thousand users [9], and the Moviepilot dataset[2] which contains 4.5 million ratings by 105 thousand users [11]. We used the full datasets for analysis and smaller subsets for some of the experiments.

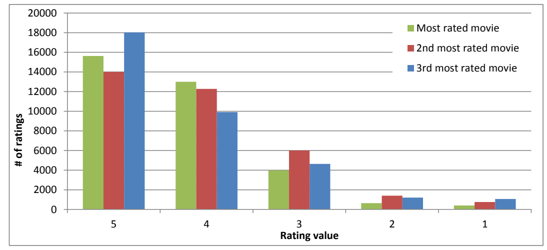


Figure 1: The rating distributions for the three most popular movies in the Movielens dataset. Note how many ratings (ca. 50%) have the highest value.

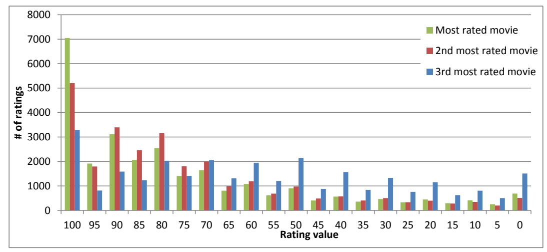


Figure 2: The rating distributions of the three most rated movies in the Moviepilot dataset. In conformity with the Movielens data shown in Fig. 1, a large portion of the ratings have the highest value.

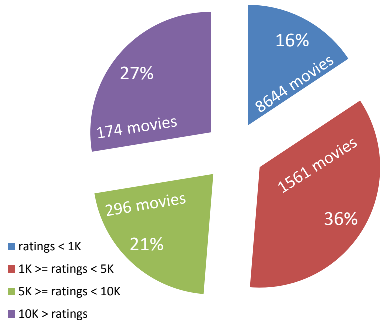
The choice of datasets was not only based on popularity, both datasets use different rating schemes which would allow for a more thorough analysis of when and how different similarity measures and weighting schemes are appropriate. In Movielens, users can rate movies on a $\{1, 2, 3, 4, 5\}$ scale, in Moviepilot the ratings are made on a $\{0, 0.5, 1.0, \ldots, 9.0, 9.5, 10.0\}$ scale making them more fine grained.

When looking at co-occurring movies, those movies which are popular (even if rated differently by different users) will add to the similarity due to the simple fact that two users will have rated the same movies. When looking at the Movielens dataset, the $69,878$ users have rated the three most popular movies $34,864$, $34,457$ and $33,667$ times respectively, as shown in Fig. 1. The implication of this is that, even if two users have rated very disjoint movies, they will have a roughly 50% chance to have rated the same movie. Additionally, almost half of the ratings on each of the three most popular movies have the same values (i.e. 5), meaning that circa 25% of all users have rated at least one of the three most popular movies identically. A similar trend is visible in the Moviepilot dataset, however, due to the rating scale being more fine grained, the effect is not as discernible. Still, as shown in Fig. 2, it is distinguishable that most of the ratings on the three most popular movies are in the upper end of the rating scale.

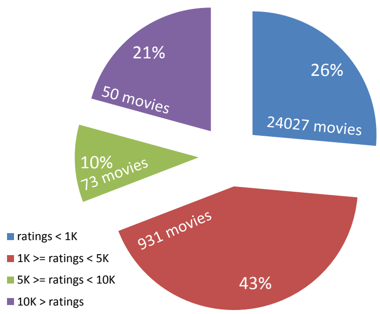
Looking at the percentage of ratings performed on movies grouped by their popularity, as shown in Fig. 3, it is visible that there is a clear dominance of (relatively) few items

[1] http://www.grouplens.org/node/470

[2] http://www.dai-labor.de/camra2010/datasets/

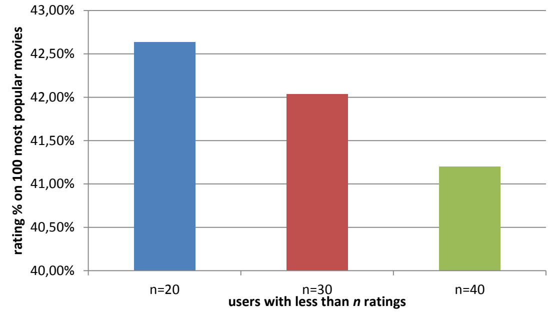


(a) The Movielens dataset

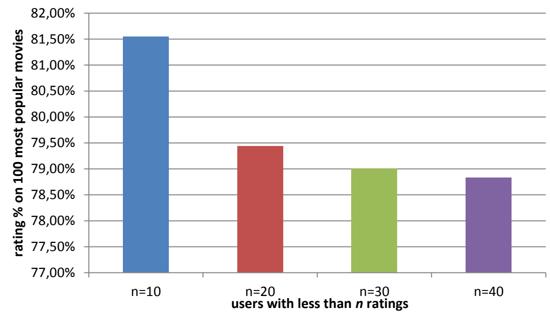


(b) The Moviepilot dataset.

Figure 3: Proportion of ratings (for movies) on number of movies in two popular collaborative filtering datasets. The sizes of the pie slices indicate the percentage of ratings performed on the most popular movies. The numbers in the slices correspond to the number of movies the ratings in each slice are made on. Note that both datasets have a relatively small number of very dominant items (the purple and green slices).



(a) Movielens



(b) Moviepilot

Figure 4: Percentages of ratings on the 100 most popular movies by users with fewer than n ratings. In the Movielens dataset there are no users with less than 20 ratings, thus the corresponding percentage for the dataset cannot be provided.

when it comes to how many times they have been rated. In Movielens (Fig. 3(a)) 174 movies, each with more than 10, 000 ratings, correspond to 27% of all ratings performed. If additionally including movies with more than 5, 000 ratings each, the number of movies rises to 470 (out of a total of 10, 000), the rating percentage grows to 48%, meaning that less than 5% of the movies correspond to almost 50% of the ratings. Similar numbers are shown for Moviepilot in Fig. 3(b).

In addition, when looking at users with few ratings, i.e. the so-called cold start problem, the effect of popular items becomes even more apparent. Fig. 4 shows the percentage of ratings on the 100 most popular movies by users with less than 10, 20, 30 and 40 ratings respectively. The density of ratings on popular items is especially discernible in the Moviepilot dataset, Fig. 4(b), where users having fewer than 10 ratings have rated almost exclusively in the set of the 100 most popular movies, up to 81.55% of their ratings are on these movies. The inclinations in the Movielens dataset, Fig. 4(a) are similar, although lower.

3.2 Similarity Measures

The cosine similarity, Euclidean distance, and Pearson correlation coefficient have been thoroughly investigated and belong to the most common (dis)similarity measures applied in neighborhood-based CF [4, 7]. In the light of our discussion in Section 3.1, one problem of these similarity measures is that popular items dominate the similarity value between two users although they are potentially less informative. For example consider a worst case scenario, with two users that have rated a few items only. In this case it is likely that they have co-rated some popular items highly according to our findings in Section 3.1. In addition, it is less likely that they have co-rated regular items, or that they agree on co-rated regular items. In this case, both users are considered as highly similar, because the similarity value is dominated by the co-ratings on the popular items. As a consequence, the recommender suggests further popular items to both users increasing their similarity even more.

One way to make recommendations of regular, but interesting items, more likely consists in assigning weights to items that devalue ratings given to popular items and appreciate ratings given to regular items. We presents weighted versions of the cosine similarity, the Euclidean distance, and the Pearson correlation coefficient. For this, we introduce some notations to simplify technicalities. We denote the set of users by $\mathcal{U}$, the set of items by $\mathcal{I}$, and the set of ratings by $\mathcal{R}$. The elements of $\mathcal{R}$ are ratings r_{ui} made by a user $u \in \mathcal{U}$ for an item $i \in \mathcal{I}$. Ratings may take values from some discrete set $\mathcal{S} \subseteq \mathbb{R}$. We write $\mathcal{I}_u$ for the subset of all items that have been rated by user u and $\mathcal{I}_{uv} = \mathcal{I}_u \cap \mathcal{I}_v$ for the subset of all items that have been co-rated by users u and v.

We summarize all ratings of a user u to an extended rating vector $\boldsymbol{x}_u = (x_{ui})_{i \in \mathcal{I}}$ of dimension $|\mathcal{I}|$ with elements

$$x_{ui} = \begin{cases} r_{ui} & i \in \mathcal{I}_u \\ \varepsilon & i \notin \mathcal{I}_u \end{cases}$$

where $\varepsilon \in \mathbb{R} \setminus \mathcal{S}$ is a pre-specified value outside $\mathcal{S}$ denoting the null- or void-rating. We define $\varepsilon + x = x + \varepsilon = 0$ and $\varepsilon \cdot x = x \cdot \varepsilon = 0$ for all $x \in \mathbb{R}$.

Suppose that $\boldsymbol{w} \in \mathcal{R}^{|\mathcal{I}|}$ is a weight vector associating a weight w_i to each item $i \in \mathcal{I}$. To formulate the weighted (dis)similarity measures, we use the following shortcut notations

$$\langle \boldsymbol{x}_u, \boldsymbol{x}_v \rangle_{\boldsymbol{w}} = \sum_{i \in \mathcal{I}_{uv}} w_i \cdot r_{ui} \cdot r_{vi} \tag{1}$$

$$\|\boldsymbol{x}_u\|_{\boldsymbol{w}} = \sqrt{\langle \boldsymbol{x}_u, \boldsymbol{x}_u \rangle_{\boldsymbol{w}}} = \sqrt{\sum_{i \in \mathcal{I}_u} w_i \cdot r_{ui}^2} \tag{2}$$

Since $\varepsilon \cdot x = 0$, only co-occurring ratings contribute to the weighted inner product of two extended rating vectors.

We define the weighted cosine similarity (cos), the weighted Euclidean distance (euc), and the weighted Pearson correlation coefficient (pcc), resp., between two users u and v by

$$\cos_{\boldsymbol{w}}(u, v) = \frac{\langle \boldsymbol{x}_u, \boldsymbol{x}_v \rangle_{\boldsymbol{w}}}{\|\boldsymbol{x}_u\|_{\boldsymbol{w}} \cdot \|\boldsymbol{x}_v\|_{\boldsymbol{w}}} \tag{3}$$

$$\text{euc}_{\boldsymbol{w}}(u, v) = \|\boldsymbol{x}_u - \boldsymbol{x}_v\|_{\boldsymbol{w}} \tag{4}$$

$$\text{pcc}_{\boldsymbol{w}}(u, v) = \frac{\langle \boldsymbol{x}_u - \bar{\boldsymbol{r}}_u, \boldsymbol{x}_v - \bar{\boldsymbol{r}}_v \rangle_{\boldsymbol{w}}}{\|\boldsymbol{x}_u - \bar{\boldsymbol{r}}_u\|_{\boldsymbol{w}} \cdot \|\boldsymbol{x}_v - \bar{\boldsymbol{r}}_v\|_{\boldsymbol{w}}}, \tag{5}$$

where $\bar{\boldsymbol{r}}_u = (\bar{r}_u, \cdots, \bar{r}_u)$ denotes the $|\mathcal{I}|$-dimensional vector with all elements being the average rating $\bar{r}_u$ of user u, that is

$$\bar{r}_u = \frac{1}{|\mathcal{I}_u|} \sum_{i \in \mathcal{I}_u} r_{ui}. \tag{6}$$

By $\boldsymbol{r}_u = (\bar{r}_u, \cdots, \bar{r}_u)$ we denote the vector consisting of $|\mathcal{I}|$ components each of which has value $\bar{r}_u$. Choosing $w_i = 1$ for all items $i \in \mathcal{I}$, we recover the unweighted standard dis(similarity) measures.

Different weighting schemes that devalue ratings of an item proportional to its popularity are possible. Here we consider the inverse user frequency (iuf) [2] and a linear weighting scheme (lin)

$$w_i^{\text{iuf}} = \ln \frac{|\mathcal{U}|}{|\mathcal{U}_i|} \tag{7}$$

$$w_i^{\text{lin}} = 1 - \frac{|\mathcal{U}_i|}{|\mathcal{R}|} \tag{8}$$

for all items $i \in \mathcal{I}$.

4. EXPERIMENTS

In our experiments we investigate how the proposed weighting schemes affect the prediction accuracy for different types of users (cold start, post cold start, power users), i.e. when a weighting scheme should be employed for best results.

The experiments were performed on the Movielens and Moviepilot datasets described in Section 3.1.

4.1 Experimental Setup

Two types of experimental setups were used, one where the weighting schemes were not used, and one where they were. Each experimental setup was performed on the similarity measures presented in Section 3.2.

For each of the datasets, several training and validation runs were performed. We chose to define positive ratings, i.e. true positives, as values higher than the user's rating average + 0.5 of the user's standard rating deviation. These were removed from the training model and extracted to a validation set. The validation sets were used to calculate precision values.

In the first type of experiment, three exhaustive recommendations were conducted on the full datasets. For all users falling within our criteria (i.e. those having true positive ratings): (i) recommendations for all users with at least 10 ratings - evaluated with the Precision@5 (P@5) metric, (ii) recommendations for all users with at least 20 ratings - evaluated with Precision@10 (P@10), and finally (iii) recommendations for all users who had rated at least 100 items - evaluated with Precision@50 (P@50).

For the cold start, post cold start and power user cases, subsets of the Movielens and Moviepilot datasets were created. Each of the subsets contained 10% of the ratings from the full datasets, selected randomly. Similarly to the experiments on the full dataset, all users in the subset falling within our criteria (those having true positive values) were evaluated. Additionally, in order to be able to identify whether an improvement of recommendation quality during the cold start, post cold start or later in a user's time line was possible, users were divided into groups. Each group consisted of users who had rated a similar amount of movies. The first group contained users with 1-4 ratings, the second with 5-10 ratings, the third with 10-20 ratings, etc. up to users with 140-150 ratings.

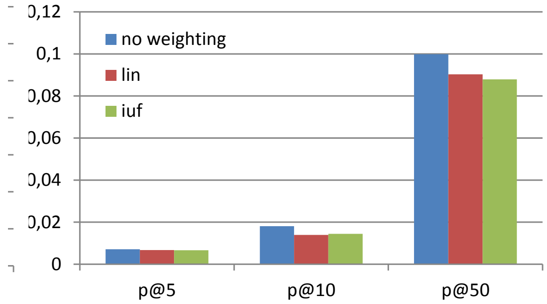
Each group was evaluated with Precision @half of the maximum ratings allowed in the group, i.e. users who had rated between 10 and 20 items were evaluated with Precision @10, users who had rated between 20 and 30 items were evaluated with Precision @15, etc.

All experiments were conducted on both datasets, using all three similarity measures and the unweighted as well as both weighted approaches.

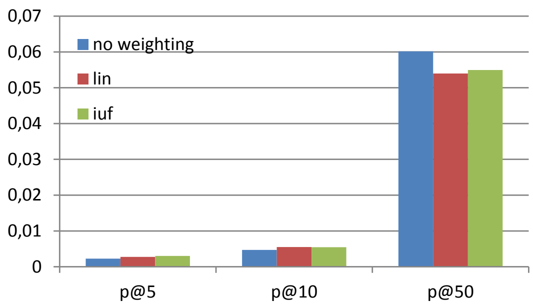
4.2 Results

The results of the first set of experiments are shown in Fig. 5. The precision@n values show that when using the weighting functions, the resulting precision@n is slightly higher for low values of n than for the unweighted approach for the Moviepilot dataset (n=5). For the Movielens dataset, the unweighted approach seems to have the upper hand. However, as n increases, the improvement decreases and at a relatively large n (n=50) the weighted approaches perform worse than the non weighted one. In the Movielens case, the unweighted approach always outperforms the weighted ones, irrelevant of n's value. This seems to be in agreement with the findings by Herlocker et al. [3]. Results for the Euclidean and cosine measures showed very similar trends and have thus been omitted.

The results of the second set of experiments shown in Fig. 6 and Fig. 7 show a slight improvement when using the linear weighting approach ($\sim 1\%$) in the case of the Euclidean distance measure for the Movielens dataset. This



(a) Movielens



(b) Moviepilot

Figure 5: The precision at 5, 10 and 50 obtained with the unweighted and both weighted approaches on the full Movielens and Moviepilot datasets.

appears for users with 40 to 60 ratings and then again for users with more than 80 ratings. For the Moviepilot dataset the (Euclidean-based) precision values were practically identical (similar to Fig. 6(b)) and have been omitted.

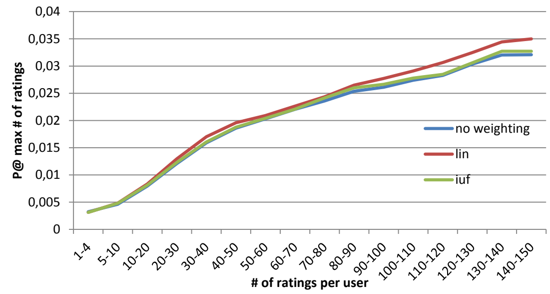
In the case of the cosine measure, all three approaches perform almost identically across the set set of precision values and users. Fig. 7(a) shows the resulting graph on the Moviepilot dataset, the Movielens counterpart has been omitted as it shows an almost identical behavior.

For the Pearson correlation, results differ from those obtained with the cosine and Euclidean measures for one of the datasets (Movielens), as shown in Fig. 7. Similar to Fig. 6(b), the results obtained on the Moviepilot dataset seem little affected by the weighting approaches resulting in a similar curve for all three models. The results for the Movielens dataset, Fig. 7(b), show a distinction between the weighted and unweighted approaches, especially in the case of users who have between 20 and 100 items (what we call the post cold start phase). In this span, the weighted approaches outperform their unweighted counterpart by (at best) more than 20%.

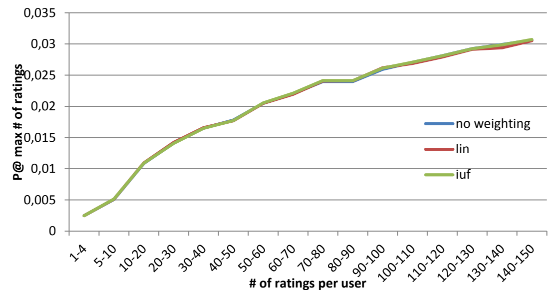
The precision changes on all datasets and weighting approaches for the Pearson correlation measure are summarized in Fig. 8.

5. CONCLUSION AND FUTURE WORK

In this paper, we have studied the effects of two weighting approaches on three common similarity measures using two different movie recommendation datasets. We have performed a set of tests in order to identify whether or not weighting schemes can be beneficial for the purpose of overcoming problems related to cold start as well as profiling

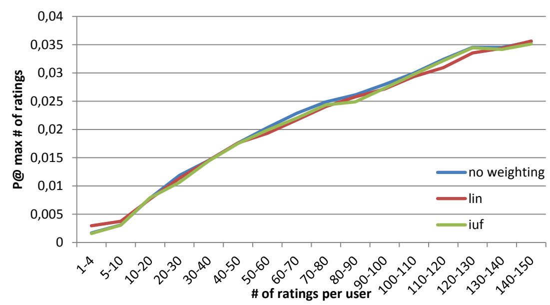


(a) Precision values for the Movielens dataset for the Euclidean distance measure

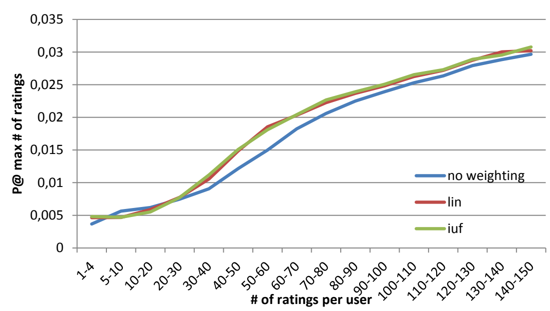


(b) Precision values for the Moviepilot dataset for the cosine similarity

Figure 6: The P@ maximum number of ratings per user in each group for the unweighted and weighted approaches for the Euclidean and cosine measures.



(a) Precision values for the Moviepilot dataset for the Pearson correlation for the unweighted and weighted approaches.



(b) Precision values for the Movielens dataset for the Pearson correlation for the unweighted and weighted approaches.

Figure 7: The P@n values for Movielens and Moviepilot using both weighted and unweighted approaches for the Person correlation.

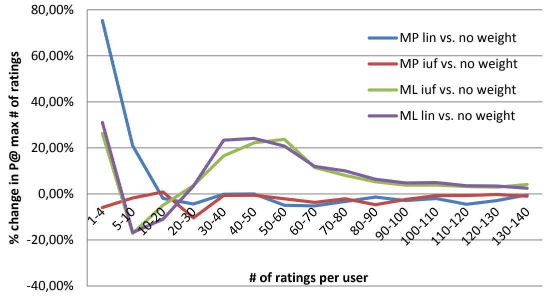


Figure 8: The changes of precision values for the weighted and unweighted approaches using the Pearson correlation for both datasets.

users in order to generate more accurate profiles not based on the most popular items.

We have observed that the weighting schemes seem to have little effect on datasets with a wide rating scale and high concentration of ratings on popular items. For instance, in the case of the Moviepilot dataset, which has a rating scale stretching from 0 to 10 in steps of 0.5, and where 80% (refer to Fig. 3 and Fig. 4) of all ratings by users with few ratings are performed on movies from the 100 most popular movies.

Furthermore, our experiments imply that the cosine measure is very insignificantly affected by any weighting measure and produces results identical no matter if weighting is applied or not. These observations hold irrelevant of the profile of the users (i.e. no matter whether the user is a new user with few ratings or a power user with many ratings).

However, there seems to be relatively much to gain in terms of precision during the post cold start phase for datasets similar to the Movielens dataset when using the Pearson correlation measure. At best, the improvement of precision values in this case is above 20%, as shown in Fig. 8.

Our results show that, in order to successfully employ weighting approaches, one needs to properly analyze the dataset, the task at hand, and fit the weighting scheme according to those.

In our future, work we intend to further delve into weighting approaches based on dataset analysis in order to identify the key features and tasks which can be beneficial to, or gain benefits from weighting approaches.

In this paper, we have presented an analysis of two popular movie rating datasets and showed how the usage of weighted similarity measures affect the performance of recommenders based on two different recommendation scenarios. We concluded that weighting schemes have little value when using the cosine similarity or datasets with wide ratings scales, the can however be beneficial if used at a stage where users are profiled in order to create more personalized and diverse recommendations.

6. ACKNOWLEDGMENTS

The authors would like to express their gratitude to the Moviepilot team who contributed to this work with dataset, relevant insights and support.

The work in this paper was conducted in the scope of the KMulE project which was sponsored by the German Federal Ministry of Economics and Technology (BMWi).

7. REFERENCES

[1] Linas Baltrunas, 'Dynamic item weighting and selection for collaborative filtering', in *Web mining 2.0 Workshop, ECML-PKDD 2007*, (2007).

[2] J S Breese, D Heckerman, and C Kadie, *Empirical analysis of predictive algorithms for collaborative filtering*, volume 461, 43–52, San Francisco, CA, 1998.

[3] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl, 'An algorithmic framework for performing collaborative filtering', in *Proc. of the 22nd annual Intl. ACM SIGIR conference on Research and development in information retrieval - SIGIR '99*, volume 54, pp. 230–237, New York, New York, USA, (1999). ACM Press.

[4] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl, 'Evaluating collaborative filtering recommender systems', *ACM Trans. Inf. Syst.*, **22**, 5–53, (January 2004).

[5] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich, *Recommender Systems: An Introduction*, volume 40, Cambridge University Press, 2010.

[6] Rong Jin, Joyce Y. Chai, and Luo Si, 'An automatic weighting scheme for collaborative filtering', in *Proc. of the 27th annual Intl. conference on Research and development in information retrieval - SIGIR '04*, p. 337, New York, New York, USA, (2004). ACM Press.

[7] YoungOk Kwon, 'Improving top-n recommendation techniques using rating variance', in *Proc. of the 2008 ACM conference on Recommender systems*, RecSys '08, pp. 307–310, New York, NY, USA, (2008). ACM.

[8] Heng Luo, Changyong Niu, Ruimin Shen, and Carsten Ullrich, 'A collaborative filtering framework based on both local user similarity and global user similarity', *Machine Learning*, **72**(3), 231–245, (July 2008).

[9] MovieLens. MovieLens 10M/100k Data Set README. `http://www.grouplens.org/system/files/README_10M100K.html` (retrieved April, 2011).

[10] István Pilászy and Domonkos Tikk, 'Recommending new movies: even a few ratings are more valuable than metadata', in *Proc. of the third ACM conference on Recommender systems*, RecSys '09, pp. 93–100, New York, NY, USA, (2009). ACM.

[11] A. Said, S. Berkovsky, and E. W. De Luca, 'Putting things in context: Challenge on context-aware movie recommendation', in *Proc. of the Workshop on Context-Aware Movie Recommendation*, CAMRa '10, New York, NY, USA, (2010). ACM.

[12] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos, 'Feature-weighted user model for recommender systems', in *Proc. of the 11th Intl. conference on User Modeling*, UM '07, Berlin, Heidelberg, (2007). Springer-Verlag.

[13] R. Wetzker, C. Zimmermann, and C. Bauckhage, 'Analyzing social bookmarking systems: A del.icio.us cookbook', in *Mining Social Data (MSoDa) Workshop Proc.* ECAI 2008, (July 2008).

[14] Kai Yu, Xiaowei Xu, Martin Ester, and Hans-Peter Kriegel, 'Feature Weighting and Instance Selection for Collaborative Filtering: An Information-Theoretic Approach*', *Knowledge and Information Systems*, **5**(2), 201–224, (April 2003).