# Term Association Measures for Memory-based Recommender Systems

Eva Suárez-García
Information Retrieval Lab
Department of Computer Science
University of A Coruña, Spain
eva.suarez.garcia@udc.es

Alfonso Landin
Information Retrieval Lab
Department of Computer Science
University of A Coruña, Spain
alfonso.landin@udc.es

Daniel Valcarce
Information Retrieval Lab
Department of Computer Science
University of A Coruña, Spain
daniel.valcarce@udc.es

Álvaro Barreiro
Information Retrieval Lab
Department of Computer Science
University of A Coruña, Spain
barreiro@udc.es

## ABSTRACT

The adaptation of Information Retrieval techniques for the item recommendation task has become a fertile research area. Previous works have established the correspondence between these two fields that allowed to adapt several retrieval techniques successfully. One line of study aims to model the item recommendation problem as a profile expansion task following the methods for query expansion in pseudo-relevance feedback. To solve the query expansion task in ad-hoc retrieval, several term association measures have been proposed in the past. In this paper, we adapt several of these measures to the top-N recommendation problem, specifically to the collaborative filtering scenario. Moreover, we perform experiments to study their effectiveness regarding accuracy, diversity and novelty. Our results show that some of the proposed measures can improve these aspects over well-known and commonly used recommendation similarity metrics (cosine similarity and Pearson's correlation coefficient).

## CCS CONCEPTS

• **Information systems → Collaborative filtering**; *Recommender systems*; Similarity measures;

## KEYWORDS

Term association measures, collaborative filtering, recommender systems

**ACM Reference Format:**
Eva Suárez-García, Alfonso Landin, Daniel Valcarce, and Álvaro Barreiro. 2018. Term Association Measures for Memory-based Recommender Systems. In *CERI'18: 5th Spanish Conference in Information Retrieval, June 25–27, 2018, Zaragoza, Spain.* ACM, New York, NY, USA, Article 6, 8 pages. https://doi.org/10.1145/3230599.3230606


*CERI'18, June 25–27, 2018, Zaragoza, Spain*

ACM ISBN 978-1-4503-6543-7/18/06…$15.00
https://doi.org/10.1145/3230599.3230606

## 1 INTRODUCTION

Lately, Recommender Systems (RS) are gaining more and more importance in people's life. Due to the increasing amount of information available, it is getting more difficult for users to discriminate between what is interesting for them and what is not. Because of this, users are becoming more demanding: they want to receive relevant information in an automatic and personalised manner. Thus, recommender systems require continuous improvement to cope with users' needs.

When building a recommender system, there are three main approaches to decide among [1]. Content-based techniques [12] use the available metadata on the items to produce recommendations. Therefore, when choosing this approach developers need to retrieve such information, which can turn out to be a challenging task. On the other hand, Collaborative Filtering (CF) techniques [23] use the already available user-item interactions, which can take the form of ratings, clicks, visualizations or purchases, among other alternatives. Lastly, hybrid techniques combine both content-based and collaborative filtering approaches.

In this paper, we focus on collaborative filtering systems [23, 24]. These techniques are commonly classified into two types. On the one hand, model-based approaches [23] exploit interaction data to train a predictive model. On the other hand, neighbourhood-based approaches [24] (also called memory-based methods) use similarities between users or items, computed with the interaction information, to produce the personalised recommendations.

The advantages of neighbourhood-based methods are their simplicity and easy interpretability of the results [24]. In contrast to model-based techniques, memory-based methods do not require a training step. Although recommendation may be costly in some neighbourhood-based approaches, usually, its efficiency can be easily improved by pre-computing the neighbours.

As expected, the effectiveness of neighbourhood-based systems depends significantly on how the neighbours are computed [4, 24]. At first, $k$-means was the most extended technique, but nowadays, a

favourite way of computing the neighbours is using the well-known clustering technique called $k$-NN ($k$-Nearest Neighbours) [24, 27].

Originally, recommender systems were designed and evaluated for the task of rating prediction. However, soon enough evidence showed that it is not as important estimating an accurate rating as providing a ranking with interesting items to the user, as the predicted ratings are usually not shown to the user [9, 20]. Therefore, the traditional rating prediction task was replaced by the top-N recommendation task where the recommendation problem was modelled as an item ranking task [9]. Thanks to this approach, the recommendation can also be interpreted as an Information Retrieval (IR) task, which allows the researcher to explore the use of well-known IR techniques. This relation between recommendation and retrieval is a fertile research line that has been producing results lately [25, 29, 30, 32–34].

In particular, regarding that research line, Parapar et al. successfully modelled the recommendation problem as a query expansion task [25]. They establish a correspondence between both tasks by considering users as documents and queries, and items as terms. Following this analogy, IR techniques originally designed for query expansion can be used for recommendation.

In IR, several term association measures have been proposed to address the query expansion problem, such as Pointwise Mutual Information [7], Expected Mutual Information [8], Dice's coefficient [14, 28], Jaccard index [22] or Pearson's Chi-square measure [16]. These metrics take into account word co-occurrence to produce a ranking with the best candidate terms to expand a query. Thanks to this, they find terms related to the original ones. This characteristic is useful in memory-based collaborative filtering recommendation, where finding related items/users is a key step. In this paper, following the path paved away by Parapar et al. [25] we want to translate these techniques used in query expansion for ad-hoc retrieval to the collaborative filtering scenario both for weighting ratings and computing the neighbourhoods.

Thus, the contributions of this paper are (1) a framework for the adaptation of term association measures to the recommender systems field that can be applied to any metric based in term co-occurrence, (2) the specific adaptation of six term association metrics, and (3) an empirical comparison of these measures with two well-known similarity metrics, cosine similarity and Pearson's correlation coefficient, in terms of accuracy, novelty and diversity. The results show that several of our proposed measures provide better values in some of these three aspects. In particular, Pearson's Chi-squared measure achieves the best values in the three of them in a user-based scenario, followed by Jaccard index and Dice's coefficient. Furthermore, keeping the exact value of the ratings, instead of considering only the presence/absence of a rating, yields significant improvements.

## 2 BACKGROUND

We introduce in this section the top-N recommendation task and previous work in memory-based recommender systems. We also explain how the recommendation task can be approached as an IR task and present several term association measures used in IR for the query expansion task.

### 2.1 Top-N Recommendation

Top-N recommendation refers to the task of finding the top-N most relevant items for a user. Methods for top-N recommendation solve the task by presenting a list of items ordered by decreasing degree of estimated relevance to the user. Top-N recommendation task has been recognised as a more realistic solution to the recommendation problem than the traditional rating prediction task, where the objective is to predict the rating that the user will assign to a given item and error-based metrics are used to evaluate the performance [3, 9, 18, 20].

Recommender systems output a value for each user $u$ and item $i$, $\hat{r}_{u,i}$. In a traditional rating prediction system, this output is an estimate of the rating the user will assign to the item. In contrast, in the case of top-N methods, this output is used internally by the system as a score to rank all the items and it is not shown to the user.

### 2.2 Memory-based Recommender Systems

Memory-based approaches directly use the past interaction information (memory) without training any model [24]. Most of the time authors indistinctly call them neighbourhood-based methods referring to the characteristic approach of most of the memory-based methods of building neighbourhoods of users or items.

Memory-based methods are commonly classified into user-based ones, those that exploit the patterns in the user-user relationships, or item-based ones, where the item-item relationships are leveraged. While the idea of user-based methods is to recommend items liked by similar-minded people, item-based methods recommend items similar to the ones the user liked in the past.

When defining these methods, the following conventions are commonly taken. The set of users is denoted as $\mathcal{U}$ and the set of items as $\mathcal{I}$. For each user $u \in \mathcal{U}$ and item $i \in \mathcal{I}$ we denote $r_{u,i}$ as the rating the user gave to the item. This value will be zero if the user has not expressed any feedback on the item. We will denote the set of items that a user $u$ has rated as $\mathcal{I}_u$. Similarly, for an item $i$ we use $\mathcal{U}_i$ to refer to the set of users that have rated it.

To exploit the feedback information, memory-based methods infer relationships between users and/or items by identifying sets of similar users/items called neighbourhoods. Several clustering techniques can be used to create these neighbourhoods [4], being the most popular $k$-Nearest Neighbours (k-NN) [24, 27]. For each target user/item and given a similarity metric, the top $k$ most similar users/items are picked. Different measures like Pearson's correlation coefficient, cosine similarity or adjusted cosine metrics can be used. In particular, cosine similarity can be efficiently calculated between two users by representing them as vectors where each dimension is an item in the system and the values for these are the ratings the users have given to the corresponding item or zero for unrated items. Item representation can be obtained analogously. In the remaining of the paper, we will use $V_u$ to denote the set of neighbours of user $u$ and $J_i$ to denote the neighbourhood of item $i$.

There are several formulations for memory-based recommenders. Some of them try to compensate the fact that users can use different values to express a similar preference for an item. They do this by normalizing the ratings using mean centring or Z-score normalization [19]. This usually leads to improvements when the rating

prediction task is considered [19, 21]. However, when it comes to the top-N recommendation task, non-normalized neighbourhood ones have proven to perform better [9, 13].

One of these techniques, WSR (Weighted Sum Recommender), a state-of-the-art technique, stands out for its simplicity and performance [31]. Its formulations, both user and item based, are as follows:

$$\hat{r}_{u,i} = \sum_{v \in V_u} s_{u,v}\, r_{v,i} \qquad (1) \qquad\qquad \hat{r}_{u,i} = \sum_{j \in J_i} s_{i,j}\, r_{u,j} \qquad (2)$$

where $s_{\cdot,\cdot}$ is the similarity between two users or items and $r_{\cdot,\cdot}$ is zero for unrated items. Please, note that the model does not enforce that the similarity measure used for weighting the rating of each neighbour has to be the same as the similarity metric used for computing the neighbourhoods, although it is usually the case.

## 2.3 Recommendation as an IR task

As stated before, the recommendation problem can be viewed as an IR task, which allows the use of techniques from this area. To adapt such techniques, first, we need to establish a correspondence between both fields. In this paper, we will adopt the modelling presented in [25], which establishes an analogy between recommendation and query expansion [5] with pseudo-relevance feedback [11]. Query expansion reformulates the user's query to improve the results in retrieval tasks. Pseudo-relevance feedback is a technique for query expansion that assumes that the top retrieved documents are relevant, although there is no explicit indicator showing whether they are indeed relevant to the query — that is the reason why they are called pseudo-relevant documents. From these pseudo-relevant documents, a set of significant terms is selected to expand the original query, which is then used in a second retrieval. The documents obtained from this process are finally presented to the user.

In the model presented in [25], the set of users $\mathcal{U}$ plays the role of the set of documents with the items rated by each user as the terms of each document and the ratings corresponding with the term frequencies. At the same time, the target user plays the role of the query with the items rated as the query terms and the ratings as term frequencies. Target user's neighbours play the role of pseudo-relevant documents and candidate items for recommendation act as candidate terms for query expansion.

Similarly, the neighbourhood computation can also be viewed as an ad-hoc retrieval task from IR [2]. From this perspective, we can see the target user $u$ as the query, $\mathcal{U}$ as the set of documents in the collection, her neighbours $V_u$ as ranked documents for the query, and items as terms. Using this approach, retrieval models can be used for computing neighbourhoods, as can be seen in [31].

## 2.4 Term association measures

One of the key objectives of query expansion methods is to add terms related to the original terms of the query to try to solve the vocabulary mismatch problem, where relevant documents do no match the query because they are using different words to describe the same topic [10]. Several term association measures have been studied for this task in IR [5, 10, 26]. Before presenting them, we first will describe the notation and estimates used throughout this section.

For a term $a$, we denote the probability of the word $a$ of occurring in a document as $P(a)$. We use $P(a,b)$ to denote the probability of the term $a$ and the term $b$ of occurring in the same document. We use the following estimates for these probabilities:

$$P(a) = \frac{n_a}{N} \qquad (3) \qquad\qquad P(a,b) = \frac{n_{ab}}{N} \qquad (4)$$

where $n_a$ is the number of documents containing word $a$, $n_{ab}$ is the number of documents containing both $a$ and $b$, and $N$ is the number of documents in the collection.

*2.4.1 Pointwise Mutual Information (PMI).* This metric [7] measures the association between two terms by comparing the probability of observing them jointly with the probability of observing them independently. Hence, for two terms $a$ and $b$, PMI is defined as:

$$PMI(a,b) = \log \frac{P(a,b)}{P(a)P(b)} = \log N \frac{n_{ab}}{n_a n_b} \qquad (5)$$

PMI is 0 when word occurrences are independent.

*2.4.2 Mutual Information (MI).* This metric, also known as Expected Mutual Information [8] is the expected value of the PMI, and is defined as follows:

$$MI(a,b) = \sum_{X_a \in \{0,1\}} \sum_{X_b \in \{0,1\}} P(X_a, X_b) \log \frac{P(X_a, P_b)}{P(X_a)P(X_b)} \qquad (6)$$

where $X_a$ and $X_b$ are binary variables indicating whether term $a$ or $b$ occurs or not. Although the MI is generally calculated using Eq. 6, sometimes it is computed taking only into account the case where both terms occur, giving the following expression:

$$SMI(a,b) = P(a,b) \log \frac{P(a,b)}{P(a)P(b)} \overset{\text{rank}}{=} n_{ab} \log \left( N \frac{n_{ab}}{n_a n_b} \right) \qquad (7)$$

where $\overset{\text{rank}}{=}$ means *rank equivalence*, i.e. both expressions produce the same ranking. We will refer to Eq. 7 as Simplified Mutual Information (SMI).

*2.4.3 Sørensen-Dice coefficient.* Popularly known as Dice's coefficient [14, 28], this measure was originally designed to be applied to presence/absence of data. The formula for this measure is:

$$Dice(a,b) = \frac{2 \cdot n_{ab}}{n_a + n_b} \overset{\text{rank}}{=} \frac{n_{ab}}{n_a + n_b} \qquad (8)$$

*2.4.4 Jaccard Index.* The Jaccard index [22], also known as intersection over union is a statistic used for comparing the similarity and diversity of two sample sets:

$$J(a,b) = \frac{P(a,b)}{P(a) + P(b) - P(a,b)} = \frac{n_{ab}}{n_a + n_b - n_{ab}} \qquad (9)$$

*2.4.5 Pearson's Chi-squared measure.* Pearson's Chi-squared measure ($\chi^2$) [16] is a statistic used for testing independence, that compares the observed number of joint occurrences with the expected number of joint occurrences if both terms were independent. Given the observed number of co-occurrences $n_{ab}$ and estimating the expected number of co-occurrences as $NP(a)P(b)$, the value of the measure is defined as:

$$\chi^2(a,b) = \frac{(n_{ab} - NP(a)P(b))^2}{NP(a)P(b)} \overset{\text{rank}}{=} \frac{(n_{ab} - \frac{1}{N} n_a n_b)^2}{n_a n_b} \qquad (10)$$

## 3 TERM ASSOCIATION MEASURES FOR RECOMMENDATION

Term association measures try to capture the similarity between terms based on how much they occur together. We propose a way to use these measures to calculate item-item and user-user similarities, following the analogy between recommendation and query expansion with pseudo-relevance feedback. To calculate item-item similarities, we equate items with terms and the set of items rated by a user with a document. Furthermore, to estimate user-user similarities, we apply a similar analogy where users are the terms, and the set of users that rated an item are all the terms of a single document.

To adapt term association measures from IR to the recommendation task, we replace term probabilities for the item and user probabilities when calculating item-item and user-user similarities, respectively. Therefore, the primary challenge consists in transforming the term probability estimates into item and user probability estimates.

We denote the probability of the item $i$ of being rated as $P(i)$, while $P(i,j)$ is the probability of items $i$ and $j$ of being rated by the same user. The estimates for these probabilities are as follows:

$$P(i) = \frac{n_i}{N} = \frac{|\mathcal{U}_i|}{|\mathcal{U}|} \quad (11) \qquad P(i,j) = \frac{n_{ij}}{N} = \frac{|\mathcal{U}_i \cap \mathcal{U}_j|}{|\mathcal{U}|} \quad (12)$$

These estimates are *binarized* in the sense that the exact rating value is disregarded, and we only consider whether a user rated an item or not. However, we propose yet another way to compute these probabilities, in a *non-binarized* manner. In this approach, the specific rating value is kept in the calculation. First, let us reformulate Eq. 11 and 12:

$$P(i) = \frac{\sum_{u \in \mathcal{U}_i} \mathbb{I}[r_{u,i}]}{\sum_{u \in \mathcal{U}} 1} \quad (13)$$

$$P(i,j) = \frac{\sum_{u \in \mathcal{U}_i \cap \mathcal{U}_j} \frac{\mathbb{I}[r_{u,i}] + \mathbb{I}[r_{i,j}]}{2}}{\sum_{u \in \mathcal{U}} 1} \quad (14)$$

where $\mathbb{I}$ is the indicator function such that $\mathbb{I}[r_{u,i}] = 1$ if user $u$ rated item $i$ and 0 otherwise. Thus, for the non-binarized versions of Eq. 13 and 14 we need to remove the $\mathbb{I}$ function from the formula:

$$P(i)_{nb} = \frac{\sum_{u \in \mathcal{U}_i} r_{u,i}}{\sum_{u \in \mathcal{U}} \max_{i \in \mathcal{I}_u} r_{u,i}} \quad (15)$$

$$P(i,j)_{nb} = \frac{\sum_{u \in \mathcal{U}_i \cap \mathcal{U}_j} \left( \frac{r_{u,i} + r_{i,j}}{2} \right)}{\sum_{u \in \mathcal{U}} \max_{i \in \mathcal{I}_u} r_{u,i}} \quad (16)$$

As can be seen, the denominator has turned into the sum of the maximum rating of each user. This formula allows us to keep the expression as a probability function. In Eq. 15, $P(i)$ is 1 when all users have rated the item with their personal maximum rating value, while in Eq. 16, $P(i,j)$ is 1 when all users have rated both items with their personal maximum rating value. A different adaptation would be to turn the denominator into the sum of the maximum possible rating value in the system for each user, i.e. probability would be 1 if all users rated the item/s with the highest value allowed by the system. However, we found that this approach produced worse results than the ones presented here.

**Table 1: Datasets statistics.**

| Dataset | Users | Items | Ratings | Density |
|---|---|---|---|---|
| MovieLens 100k | 943 | 1,682 | 100,000 | 6.305% |
| MovieLens 1M | 6,040 | 3,706 | 1,000,209 | 4.468% |
| R3-Yahoo | 15,400 | 1,000 | 365,704 | 2.375% |
| LibraryThing | 7,279 | 37,232 | 749,401 | 0.277% |

For the sake of summarizing, we have the following equalities:

$$n_i = \sum_{u \in \mathcal{U}_i} \mathbb{I}[r_{u,i}] = |\mathcal{U}_i| \quad (17)$$

$$n_{ij} = \sum_{u \in \mathcal{U}_i \cap \mathcal{U}_j} \frac{\mathbb{I}\left[r_{u,i}\right] + \mathbb{I}\left[r_{i,j}\right]}{2} = |\mathcal{U}_i \cap \mathcal{U}_j| \quad (18)$$

$$N = \sum_{u \in \mathcal{U}} 1 = |\mathcal{U}| \quad (19)$$

$$n_{i(nb)} = \sum_{u \in \mathcal{U}_i} r_{u,i} \quad (20)$$

$$n_{ij(nb)} = \sum_{u \in \mathcal{U}_i \cap \mathcal{U}_j} \left( \frac{r_{u,i} + r_{i,j}}{2} \right) \quad (21)$$

$$N_{(nb)} = \sum_{u \in \mathcal{U}} \max_{i \in \mathcal{I}_u} r_{u,i} \quad (22)$$

All of these formulas are for computing item-item similarities. If we wanted to compute user-user similarities, we would have to swap the roles of users and items. Having established these estimates, the adaptation of the term association measures is relatively straightforward, by simply replacing the probabilities by these new expressions. By doing so, we obtained both binarized and non-binarized formulas for all measures.

## 4 EXPERIMENTAL EVALUATION

In this section, we first describe the experimental settings—including the datasets and the evaluation protocol. After that, we present the results of our experiments comparing them to previous techniques.

### 4.1 Datasets

To perform our experiments, we use several datasets from different domains, all of which have explicit feedback data. In particular, we used the MovieLens 100k and MovieLens 1M movie datasets[1], the R3-Yahoo! music dataset[2] and the LibraryThing book dataset. The first three datasets contain integer ratings between 1 and 5. The ratings in the LibraryThing dataset are in the interval between 0.5 and 5.0 with possible values in increments of 0.5. We provide details for each dataset in Table 1. All datasets were randomly partitioned to conduct the experiments, with 80% of the ratings from each user being used for training and the remaining 20% for the test.

### 4.2 Evaluation protocol

To evaluate the recommenders for the top-N recommendation task, we follow the TestItems evaluation approach as described in [3]. In this evaluation methodology, the test items set is defined as all the items that have a rating by any user in the test set. For each

[1] http://grouplens.org/datasets/movielens
[2] http://webscope.sandbox.yahoo.com

**Table 2: User-based recommendation results, showing nDCG@10, Gini@10 and MSI@10 on four different datasets. Binarized and non-binarized measures are subscripted with *bin* and *nb*, respectively. Statistical significant improvements (permutation test with $p < 0.01$) in nDCG@10 and MSI@10 with respect to Pearson and cosine baselines are superscripted with *a* and *b* respectively. Statistical significant improvements of a non-binarized metric with respect to its binarized counterpart are superscripted with ∗; the reverse situations are superscripted with †.**

| Similarity | Metric | Movielens 100k | Movielens 1M | R3-Yahoo | LibraryThing |
|---|---|---|---|---|---|
| Pearson | nDCG@10 | 0.1693 | 0.1076 | 0.0196 | 0.0729 |
| | Gini@10 | 0.0181 | 0.0090 | 0.0225 | 0.0018 |
| | MSI@10 | 8.5662 | 9.5795 | 15.5851 | 18.9782 |
| Cosine | nDCG@10 | 0.3902 | 0.3449 | 0.0261 | 0.1932 |
| | Gini@10 | 0.0549 | 0.0339 | 0.0604 | 0.0180 |
| | MSI@10 | 11.0611 | 12.0341 | 19.5503 | 28.4180 |
| $PMI_{bin}$ | nDCG@10 | $0.3630^{a}$ | $0.3314^{a}$ | $0.0243^{a}$ | $0.1789^{a}$ |
| | Gini@10 | 0.0420 | 0.0319 | 0.0434 | 0.0160 |
| | MSI@10 | $10.4691^{a}$ | $11.9799^{a}$ | $18.5376^{a\dagger}$ | $\mathbf{28.5271}^{a\dagger}$ |
| $PMI_{nb}$ | nDCG@10 | $0.3654^{a}$ | $0.3333^{a*}$ | $0.0238^{a}$ | $0.1817^{a*}$ |
| | Gini@10 | 0.0471 | 0.0323 | 0.0411 | 0.0092 |
| | MSI@10 | $10.7612^{a*}$ | $12.0433^{a*}$ | $18.0530^{a}$ | $25.8058^{a}$ |
| $SMI_{bin}$ | nDCG@10 | $0.3804^{a}$ | $0.3197^{a\dagger}$ | $0.0258^{a}$ | $0.1711^{a}$ |
| | Gini@10 | 0.0518 | 0.0238 | 0.0584 | 0.0127 |
| | MSI@10 | $11.0153^{a}$ | $11.2415^{a}$ | $19.5927^{a\dagger}$ | $26.6928^{a}$ |
| $SMI_{nb}$ | nDCG@10 | $0.3854^{a*}$ | $0.2977^{a}$ | $0.0250^{a}$ | $0.1844^{a*}$ |
| | Gini@10 | 0.0535 | 0.0185 | 0.0386 | 0.0155 |
| | MSI@10 | $11.0586^{a*}$ | $10.8327^{a}$ | $18.4847^{a}$ | $27.5679^{a*}$ |
| $MI_{bin}$ | nDCG@10 | $0.3854^{a}$ | $0.3335^{a\dagger}$ | $0.0259^{a}$ | $0.1767^{a}$ |
| | Gini@10 | 0.0539 | 0.0269 | 0.0488 | 0.0142 |
| | MSI@10 | $11.0742^{a}$ | $11.4706^{a}$ | $18.9018^{a\dagger}$ | $27.1750^{a}$ |
| $MI_{nb}$ | nDCG@10 | $0.3921^{a*}$ | $0.3091^{a}$ | $0.0251^{a}$ | $0.1853^{a*}$ |
| | Gini@10 | 0.0557 | 0.0203 | 0.0380 | 0.0156 |
| | MSI@10 | $11.1256^{ab*}$ | $10.9828^{a}$ | $18.3848^{a}$ | $27.5426^{a*}$ |
| $Dice_{bin}$ | nDCG@10 | $0.3856^{a}$ | $0.3453^{a}$ | $0.0257^{a}$ | $0.1847^{a}$ |
| | Gini@10 | 0.0536 | 0.0347 | 0.0517 | 0.0176 |
| | MSI@10 | $10.9461^{a}$ | $12.0484^{a}$ | $19.0984^{a}$ | $28.2811^{a}$ |
| $Dice_{nb}$ | nDCG@10 | $0.3875^{a}$ | $0.3465^{a}$ | $0.0262^{a}$ | $0.1870^{a*}$ |
| | Gini@10 | 0.0544 | **0.0411** | 0.0653 | 0.0177 |
| | MSI@10 | $10.9769^{a*}$ | $\mathbf{12.5187}^{ab*}$ | $19.9364^{ab*}$ | $28.3074^{a}$ |
| $Jaccard_{bin}$ | nDCG@10 | $0.3860^{a}$ | $0.3457^{a}$ | $0.0258^{a}$ | $0.1853^{a}$ |
| | Gini@10 | 0.0539 | 0.0350 | 0.0518 | 0.0180 |
| | MSI@10 | $10.9598^{a}$ | $12.0668^{ab}$ | $19.1062^{a}$ | $28.3610^{a}$ |
| $Jaccard_{nb}$ | nDCG@10 | $0.3880^{a}$ | $0.3470^{ab}$ | $\mathbf{0.0263}^{a}$ | $0.1878^{a*}$ |
| | Gini@10 | 0.0547 | 0.0350 | 0.0657 | **0.0181** |
| | MSI@10 | $10.9894^{a*}$ | $12.0831^{ab}$ | $19.9575^{ab*}$ | $28.4025^{a}$ |
| $Chi\text{-}square_{bin}$ | nDCG@10 | $0.3948^{ab}$ | $\mathbf{0.3573}^{ab\dagger}$ | $0.0257^{a}$ | $0.1947^{a}$ |
| | Gini@10 | 0.0575 | 0.0385 | 0.0473 | 0.0157 |
| | MSI@10 | $11.1259^{ab}$ | $12.3090^{ab}$ | $18.7855^{a}$ | $27.4074^{a}$ |
| $Chi\text{-}square_{nb}$ | nDCG@10 | $\mathbf{0.3992}^{ab*}$ | $0.3539^{ab}$ | $\mathbf{0.0263}^{a}$ | $\mathbf{0.1979}^{ab*}$ |
| | Gini@10 | **0.0582** | 0.0371 | **0.0709** | 0.0167 |
| | MSI@10 | $\mathbf{11.1519}^{ab*}$ | $12.2443^{ab}$ | $\mathbf{20.2344}^{ab*}$ | $27.6873^{a*}$ |

**Table 3: Item-based recommendation results, showing nDCG@10, Gini@10 and MSI@10 on four different datasets. Binarized and non-binarized measures are subscripted with *bin* and *nb*, respectively. Statistical significant improvements (permutation test with $p < 0.01$) in nDCG@10 and MSI@10 with respect to Pearson and cosine baselines are superscripted with *a* and *b* respectively. Statistical significant improvements of a non-binarized metric with respect to its binarized counterpart are superscripted with ∗; the reverse situations are superscripted with †.**

| Similarity | Metric | Movielens 100k | Movielens 1M | R3-Yahoo | LibraryThing |
|---|---|---|---|---|---|
| Pearson | nDCG@10 | 0.0054 | 0.0005 | 0.0095 | 0.0189 |
| | Gini@10 | 0.0672 | 0.0244 | 0.2319 | 0.1645 |
| | MSI@10 | 47.5752 | 67.4544 | **48.1618** | 62.1193 |
| Cosine | nDCG@10 | **0.3833** | **0.3376** | **0.0273** | 0.2624 |
| | Gini@10 | 0.0901 | 0.0713 | 0.1034 | 0.1200 |
| | MSI@10 | 12.9787 | 14.6225 | 21.7559 | 43.4896 |
| $PMI_{bin}$ | nDCG@10 | 0.0059 | $0.0092^{a\dagger}$ | $0.0176^{a\dagger}$ | $0.1445^{a\dagger}$ |
| | Gini@10 | 0.0391 | 0.0339 | **0.6162** | 0.3505 |
| | MSI@10 | $54.0094^{ab}$ | $\mathbf{73.4936}^{ab\dagger}$ | $41.1525^{b}$ | $63.4519^{ab}$ |
| $PMI_{nb}$ | nDCG@10 | 0.0050 | $0.0054^{a}$ | $0.0133^{a}$ | $0.1366^{a}$ |
| | Gini@10 | 0.0256 | 0.0375 | 0.4583 | **0.3508** |
| | MSI@10 | $\mathbf{56.4564}^{ab*}$ | $70.8676^{ab}$ | $43.6928^{b*}$ | $\mathbf{63.7748}^{ab*}$ |
| $SMI_{bin}$ | nDCG@10 | $0.3504^{a}$ | $0.3107^{a\dagger}$ | $0.0256^{a\dagger}$ | $0.2309^{a\dagger}$ |
| | Gini@10 | 0.0828 | 0.0594 | 0.0563 | 0.0223 |
| | MSI@10 | $13.1697^{b\dagger}$ | $14.8859^{b}$ | 20.7232 | $30.7904^{\dagger}$ |
| $SMI_{nb}$ | nDCG@10 | $0.3710^{a*}$ | $0.2643^{a}$ | $0.0168^{a}$ | $0.1973^{a}$ |
| | Gini@10 | 0.0596 | 0.0903 | 0.3976 | 0.0161 |
| | MSI@10 | 11.4944 | $18.0360^{b*}$ | $42.4918^{b*}$ | 27.2608 |
| $MI_{bin}$ | nDCG@10 | $0.3373^{a}$ | $0.2940^{a\dagger}$ | $0.0258^{a\dagger}$ | $0.2464^{a\dagger}$ |
| | Gini@10 | 0.0851 | 0.0710 | 0.0897 | 0.0289 |
| | MSI@10 | $13.7245^{b\dagger}$ | $16.2571^{b}$ | $24.1635^{b}$ | $33.5735^{\dagger}$ |
| $MI_{nb}$ | nDCG@10 | $0.3712^{a*}$ | $0.2362^{a}$ | 0.0112 | $0.2096^{a}$ |
| | Gini@10 | 0.0613 | 0.0870 | 0.0122 | 0.0177 |
| | MSI@10 | 11.7181 | $18.8748^{b*}$ | $24.1686^{b}$ | 28.2052 |
| $Dice_{bin}$ | nDCG@10 | $0.3607^{a}$ | $0.3225^{a}$ | $0.0259^{a}$ | $0.2614^{a}$ |
| | Gini@10 | 0.0813 | 0.0619 | 0.0701 | 0.1108 |
| | MSI@10 | 12.6802 | $14.1014^{\dagger}$ | 19.7073 | 42.1952 |
| $Dice_{nb}$ | nDCG@10 | $0.3632^{a*}$ | $0.3231^{a}$ | $0.0266^{a*}$ | $0.2615^{a}$ |
| | Gini@10 | 0.0814 | 0.0603 | 0.0764 | 0.1114 |
| | MSI@10 | 12.6822 | 13.9748 | $20.1246^{*}$ | $42.3233^{*}$ |
| $Jaccard_{bin}$ | nDCG@10 | $0.3625^{a}$ | $0.3251^{a}$ | $0.0260^{a}$ | $0.2679^{ab}$ |
| | Gini@10 | 0.0777 | 0.0531 | 0.0643 | 0.1144 |
| | MSI@10 | $12.4758^{\dagger}$ | $13.5672^{\dagger}$ | 19.3526 | 42.8385 |
| $Jaccard_{nb}$ | nDCG@10 | $0.3652^{a}$ | $0.3251^{a}$ | $0.0267^{a}$ | $\mathbf{0.2681}^{ab}$ |
| | Gini@10 | 0.0706 | 0.0526 | 0.0727 | 0.1153 |
| | MSI@10 | 12.2601 | 13.4775 | $19.8563^{*}$ | $42.9928^{*}$ |
| $Chi\text{-}square_{bin}$ | nDCG@10 | $0.3009^{a}$ | $0.2522^{a\dagger}$ | $0.0252^{a\dagger}$ | $0.2576^{a}$ |
| | Gini@10 | **0.1506** | 0.1107 | 0.2155 | 0.1617 |
| | MSI@10 | $17.2052^{b\dagger}$ | $20.1509^{b}$ | $31.0904^{b\dagger}$ | $50.0770^{b\dagger}$ |
| $Chi\text{-}square_{nb}$ | nDCG@10 | $0.3555^{a*}$ | $0.2023^{a}$ | $0.0170^{a}$ | $0.2669^{ab*}$ |
| | Gini@10 | 0.1061 | **0.1446** | 0.0141 | 0.1370 |
| | MSI@10 | $14.0452^{b}$ | $23.2940^{b*}$ | 14.3163 | $46.9330^{b}$ |

user, we rank all the items in the test set excluding those that have been rated by the user in the training set. This protocol allows us to assess how well a recommender system can differentiate relevant from non-relevant items [3].

When evaluating the rating prediction task, error based metrics, such as Mean Absolute Error (MAE) or Root Mean Square Error (RMSE), have traditionally been used [18]. When switching to the top-N recommendation task, these measures are no longer useful, and traditional IR metrics need to be used to assess the effectiveness of the system [9, 20]. To evaluate the effectiveness of the recommendations we use the Normalized Discounted Cumulative Gain (nDCG), using the *standard formulation* as described in [35] with ratings as graded relevance judgements. In our experiments, only items with a rating of 4.0 or higher are considered relevant when evaluating.

While accuracy is probably the most desirable characteristic of a recommender system, some other properties are also important [6, 18]. Being able to recommend from all the catalogue of items, instead of only the more popular ones, is usually an added benefit for a recommender system. We measure the capability of a recommender system to produce diverse recommendations with the complement of the Gini index [15]. When the value of the index is 0 it signifies that a single item is being recommended to all the users. A value of 1 means that all the items are recommended equally to all the users.

Another desirable property is the novelty, the ability of the recommender system to generate unexpected results that the user probably did not know. In other words, the highest the novelty, the highest the probability of the recommender system of producing serendipitous recommendations, usually associated with higher user satisfaction [17]. We use the mean self-information (MSI) [36] to assess the novelty of the recommendations.

We evaluate all metrics with a cut-off of 10. We do this because we are interested in evaluating the quality of the top recommendations. These are the ones the user usually consumes, either because the space for showing the recommendations is limited and few recommendations are presented or because the user only pays attention to the top recommendations.

### 4.3 Results

We tested all the proposed term association measures (both user-based and item-based approaches) on the four datasets. We used k-NN for computing the neighbourhoods and tuned the number of nearest neighbours from $k = 25$ to 125 in steps of 25 neighbours. We used the same measure for neighbourhood computation and rating weighting, as preliminary experiments did not show significant differences between combining metrics and using the same. We compared our proposed measures against cosine similarity and Pearson's coefficient. Tables 2 and 3 show the best results for nDCG@10, and their corresponding values for Gini@10 and MSI@10, for user-based and item-based recommendation, respectively.

Overall, Pearson's Chi-squared measure performs the best in the user-based scenario, outperforming the baselines on every dataset. It also yields some of the best results in novelty and diversity, which makes it a fairly useful similarity metric as an increase in novelty or diversity usually implies a reduction in accuracy [36]. Jaccard index and Dice's coefficient also showed similar results regarding nDCG@10.

In the user-based scenario, non-binarized measures tend to improve the results of the binarized ones: in 10 out of 24 cases, the non-binarized version is significantly better than its binarized counterpart regarding nDCG@10, while only in 3 out of 24 cases the binarized metrics significantly outperform the non-binarized ones. In contrast, in the item-based recommendation, the trend is reversed but not so clear: in 6 out of 24 cases the non-binarized metric significantly improved the results of the binarized ones, whereas in 11 out of 24 cases the binarized metric significantly performed better than the non-binarized one.

However, item-based term association measures show worse performance than in the user-based scenario, generally not improving cosine similarity: only Pearson Chi-squared measure (in its non-binarized version) and Jaccard index (in both versions) significantly outperformed cosine similarity, and only in the LibraryThing dataset. Examining the results in more detail, we can see that Pearson Chi-square measure performed notably worse than in user-based recommendation, being outperformed by Jaccard index and Dice's coefficient most of the time. PMI yields the best novelty results overall, but in exchange for very low precision, as was to be expected due to the trade-off between accuracy and novelty [36].

In summary, we can conclude that Pearson Chi-squared measure is a highly valuable similarity for the user-based scenario, as it improves both precision and novelty w.r.t. cosine. Jaccard index and Dice's coefficient have also proven to be useful metrics. Non-binarized versions should be used in this case, as they tend to perform better. On the other hand, PMI should be discarded as a viable metric, as its low precision renders it unusable. Lastly, none of the proposed metrics seems good enough for the item-based scenario, as its occasional improvements in novelty and diversity do not compensate the loss of precision.

## 5 CONCLUSIONS AND FUTURE WORK

In this work, we have adapted several term association measures for their use in collaborative filtering. On the one hand, we present adaptations where the explicit value of the preference of the user is dropped, using this preferences in a binarized form where only showing interest in an item matters. On the other hand, we also propose versions where the ratings that the user gave to items are taken into account when computing the measures. These measures can be used to estimate user-user and item-item similarities. We tested the performance of the measures by using them with a memory-based recommender, in both user and item based form.

The results of the experiments show that when it comes to the user-based scenario, the Pearson's Chi-squared measure outperforms the baselines regarding accuracy, with a statistically significant difference in 3 out of 4 datasets. That measure can do so while increasing diversity and novelty at the same time in three of the four datasets. Also, in this same scenario, the non-binarized versions of the metrics have proven to perform significantly better in most cases, and should thus be preferred.

As future work, it may be worthwhile to analyse the results produced by these metrics with other memory-based formulations

different from WSR. It can also be interesting to study the performance of these measures with implicit collaborative filtering data.

## ACKNOWLEDGMENTS

This work has received financial support from project TIN2015-64282-R (MINECO/ERDF), project GPC ED431B 2016/035 (Xunta de Galicia) and accreditation ED431G/01 (Xunta de Galicia/ERDF). The third author also acknowledges the support of grant FPU014/01724 (MECD).

## REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2005. 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 (June 2005), 734–749. https://doi.org/10.1109/TKDE.2005.99

[2] Nicholas J. Belkin and W. Bruce Croft. 1992. Information Filtering and Information Retrieval: Two Sides of the Same Coin? *Commun. ACM* 35, 12 (Dec. 1992), 29–38. https://doi.org/10.1145/138859.138861

[3] Alejandro Bellogín, Pablo Castells, and Iván Cantador. 2011. Precision-Oriented Evaluation of Recommender Systems. In *Proceedings of the 5th ACM Conference on Recommender systems (RecSys '11)*. ACM, New York, NY, USA, 333–336. https://doi.org/10.1145/2043932.2043996

[4] Alejandro Bellogin and Javier Parapar. 2012. Using Graph Partitioning Techniques for Neighbour Selection in User-based Collaborative Filtering. In *Proceedings of the Sixth ACM Conference on Recommender Systems (RecSys '12)*. ACM, New York, NY, USA, 213–216. https://doi.org/10.1145/2365952.2365997

[5] Claudio Carpineto and Giovanni Romano. 2012. A Survey of Automatic Query Expansion in Information Retrieval. *ACM Comput. Surv.* 44, 1, Article 1 (Jan. 2012), 50 pages. https://doi.org/10.1145/2071389.2071390

[6] Pablo Castells, Neil J. Hurley, and Saúl Vargas. 2015. Novelty and Diversity in Recommender Systems. In *Recommender Systems Handbook* (2nd ed.), Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer US, Boston, MA, USA, 881–918. https://doi.org/10.1007/978-1-4899-7637-6_26

[7] Kenneth Ward Church and Patrick Hanks. 1990. Word Association Norms, Mutual Information, and Lexicography. *Comput. Linguist.* 16, 1 (March 1990), 22–29. http://dl.acm.org/citation.cfm?id=89086.89095

[8] T.M. Cover and J.A. Thomas. 2012. *Elements of Information Theory*. Wiley. https://books.google.es/books?id=VWq5GG6ycxMC

[9] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of Recommender Algorithms on Top-N Recommendation Tasks. In *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys '10)*. ACM, New York, NY, USA, 39–46. https://doi.org/10.1145/1864708.1864721

[10] Bruce Croft, Donald Metzler, and Trevor Strohman. 2009. *Search Engines: Information Retrieval in Practice* (1st ed.). Addison-Wesley Publishing Company, USA.

[11] W.B. Croft and D.J. Harper. 1979. Using Probabilistic Models of Document Retrieval without Relevance Information. *Journal of Documentation* 35, 4 (1979), 285–295. https://doi.org/10.1108/eb026683

[12] Marco de Gemmis, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro. 2015. Semantics-Aware Content-Based Recommender Systems. In *Recommender Systems Handbook*. 119–159. https://doi.org/10.1007/978-1-4899-7637-6_4

[13] Mukund Deshpande and George Karypis. 2004. Item-based top-N Recommendation Algorithms. *ACM Transactions on Information Systems* 22, 1 (2004), 143–177. https://doi.org/10.1145/963770.963776

[14] Lee R. Dice. 1945. Measures of the Amount of Ecologic Association Between Species. *Ecology* 26, 3 (1945), 297–302. https://doi.org/10.2307/1932409

[15] Daniel Fleder and Kartik Hosanagar. 2009. Blockbuster Culture's Next Rise or Fall: The Impact of Recommender Systems on Sales Diversity. *Management Science* 55, 5 (2009), 697–712. https://doi.org/10.1287/mnsc.1080.0974

[16] Karl Pearson F.R.S. 1900. X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 50, 302 (1900), 157–175. https://doi.org/10.1080/14786440009463897

[17] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. 2010. Beyond Accuracy: Evaluating Recommender Systems by Coverage and Serendipity. In *Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys '10)*. ACM, New York, NY, USA, 257–260. https://doi.org/10.1145/1864708.1864761

[18] Asela Gunawardana and Guy Shani. 2015. Evaluating Recommender Systems. In *Recommender Systems Handbook* (2nd ed.), Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer US, Boston, MA, USA, 265–308. https://doi.org/10.1007/978-1-4899-7637-6_8

[19] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. 1999. An Algorithmic Framework for Performing Collaborative Filtering. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99)*. ACM, New York, NY, USA, 230–237. https://doi.org/10.1145/312624.312682

[20] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems* 22, 1 (2004), 5–53. https://doi.org/10.1145/963770.963772

[21] Adele E. Howe and Ryan D. Forbes. 2008. Re-considering Neighborhood-based Collaborative Filtering Parameters in the Context of New Data. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM '08)*. ACM, New York, NY, USA, 1481–1482. https://doi.org/10.1145/1458082.1458345

[22] Paul Jaccard. 1901. Distribution de la Flore Alpine dans le Bassin des Dranses et dans quelques régions voisines. 37 (01 1901), 241–72.

[23] Yehuda Koren and Robert Bell. 2015. Advances in Collaborative Filtering. In *Recommender Systems Handbook* (2nd ed.), Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer US, Boston, MA, USA, 77–118. https://doi.org/10.1007/978-1-4899-7637-6_3

[24] Xia Ning, Christian Desrosiers, and George Karypis. 2015. A Comprehensive Survey of Neighborhood-Based Recommendation Methods. In *Recommender Systems Handbook* (2nd ed.), Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer US, Boston, MA, USA, 37–76. https://doi.org/10.1007/978-1-4899-7637-6_2

[25] Javier Parapar, Alejandro Bellogín, Pablo Castells, and Álvaro Barreiro. 2013. Relevance-based Language Modelling for Recommender Systems. *Inf. Process. Manage.* 49, 4 (July 2013), 966–980. https://doi.org/10.1016/j.ipm.2013.03.001

[26] C. J. Van Rijsbergen. 1979. *Information Retrieval* (2nd ed.). Butterworth-Heinemann, Newton, MA, USA.

[27] J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. The Adaptive Web. Springer-Verlag, Berlin, Heidelberg, Chapter Collaborative Filtering Recommender Systems, 291–324. http://dl.acm.org/citation.cfm?id=1768197.1768208

[28] T. Sørensen. 1948. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biol. Skr.* 5 (1948), 1–34.

[29] Daniel Valcarce, Javier Parapar, and Álvaro Barreiro. 2016. Additive Smoothing for Relevance-Based Language Modelling of Recommender Systems. In *Proceedings of the 4th Spanish Conference on Information Retrieval (CERI '16)*. ACM, New York, NY, USA, Article 9, 8 pages. https://doi.org/10.1145/2934732.2934737

[30] Daniel Valcarce, Javier Parapar, and Alvaro Barreiro. 2016. Efficient Pseudo-Relevance Feedback Methods for Collaborative Filtering Recommendation. In *Advances in Information Retrieval - 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings*. 602–613. https://doi.org/10.1007/978-3-319-30671-1_44

[31] Daniel Valcarce, Javier Parapar, and Álvaro Barreiro. 2016. Language Models for Collaborative Filtering Neighbourhoods. In *Proceedings of the 38th European Conference on Information Retrieval (ECIR '16)*. Springer, Berlin, Heidelberg, 614–625. https://doi.org/10.1007/978-3-319-30671-1_45

[32] Jun Wang, Arjen P. de Vries, and Marcel J. T. Reinders. 2006. A User-item Relevance Model for Log-based Collaborative Filtering. In *Proceedings of the 28th European Conference on Advances in Information Retrieval (ECIR'06)*. Springer-Verlag, Berlin, Heidelberg, 37–48. https://doi.org/10.1007/11735106_5

[33] Jun Wang, Arjen P. de Vries, and Marcel J. T. Reinders. 2008. Unified Relevance Models for Rating Prediction in Collaborative Filtering. *ACM Trans. Inf. Syst.* 26, 3, Article 16 (June 2008), 42 pages. https://doi.org/10.1145/1361684.1361689

[34] Jun Wang, Stephen Robertson, Arjen P. Vries, and Marcel J. Reinders. 2008. Probabilistic Relevance Ranking for Collaborative Filtering. *Inf. Retr.* 11, 6 (Dec. 2008), 477–497. https://doi.org/10.1007/s10791-008-9060-1

[35] Yining Wang, Liwei Wang, Yuanzhi Li, Di He, Wei Chen, and Tie-Yan Liu. 2013. A Theoretical Analysis of NDCG Ranking Measures. In *Proceedings of the 26th Annual Conference on Learning Theory (COLT '13)*. JMLR.org, 1–30.

[36] Tao Zhou, Zoltán Kuscsik, J.-G. Liu, Matús Medo, Joseph Rushton Wakeling, and Y.-C. Zhang. 2010. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences* 107, 10 (2010), 4511–4515. https://doi.org/10.1073/pnas.1000488107