Understanding Similarity Metrics in Neighbour-based Recommender Systems

Alejandro Bellogín and Arjen P. de Vries
Information Access, Centrum Wiskunde & Informatica
Science Park 123, 1098 XG Amsterdam, The Netherlands
{A.Bellogin, Arjen.de.Vries}@cwi.nl

ABSTRACT

Neighbour-based collaborative filtering is a recommendation technique that provides meaningful and, usually, accurate recommendations. The method's success depends however critically upon the similarity metric used to find the most similar users (neighbours), the basis of the predictions made. In this paper, we explore twelve features that aim to explain why some user similarity metrics perform better than others. Specifically, we define two sets of features, a first one based on statistics computed over the distance distribution in the neighbourhood, and, a second one based on the nearest neighbour graph. Our experiments with a public dataset show that some of these features are able to correlate with the performance up to a 90%.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Relevance feedback, Information Filtering

General Terms

Algorithms, Experimentation, Performance

Keywords

Neighbour selection, Similarity metric, Collaborative Filtering

1. INTRODUCTION

The aim of Recommender Systems (RS) is to assist users in finding their way through huge databases and catalogues, by filtering and suggesting relevant items taking into account or inferring the users' preferences (i.e., tastes, interests, or priorities). Collaborative Filtering (CF) systems can be considered as the earliest and most widely deployed recommendation approach [15], suggesting interesting items to users based on the preferences from "similar" people [19, 1]. Usually, ratings (explicit relevance values given by users to items) are the primary source of evidence upon which this similarity is established. As CF algorithms exploit the active user's ratings to make predictions, no item descriptions are needed to provide recommendations. In this paper, we focus our attention to the memory-based class of CF algorithms that are user-based. These algorithms compute user similarities from the user's item ratings, typically based on distance and correlation metrics [9]; items not yet seen by the active user but rated by users highly similar to the active user (in terms of their item ratings) are then used to produce the recommendations. The "similar" people found (whose preferences are used to predict ratings for the active user) are usually referred to as the active user's *neighbours*.

Neighbour-based recommender systems have some advantages when compared against other types of recommendation techniques [9]. First, they are very intuitive and simple to implement, which allow for richer explanations and justifications of the recommendations, since the interpretation of the results is straightforward. Moreover, it is possible to efficiently compute the recommendations, since the user similarity matrices can be precomputed and stored prior to making the suggestions; in part, this also explains the stability of these methods, in the sense that they are little affected by changes in the domain (e.g., new items or users). On the other hand, these methods also suffer from two main shortcomings mainly due to data sparsity: limited coverage and lower accuracy.

As mentioned before, the choice of the similarity metric is a key aspect of neighbour-based methods. In this context, modifications thereof and alternative similarity functions have been proposed [23, 25], but no principled explanations about why the modifications or the original metrics do or do not work. To better understand the role of these metrics in recommendation, we formulate as follows our research questions: **RQ1)** is it possible to find user similarity properties able to predict the performance of such metrics when integrated in nearest neighbour CF algorithms?, and, in that case, **RQ2)** how should we modify a bad-performing similarity metric to improve its performance?

With these goals in mind, we explore the effect of different features computed for variations of user similarity metrics. These features are classified in two groups: distance-based and graph-based. We have found that some of these features are highly correlated with the success of a similarity metric when used in a neighbour-based recommender system, and thus, they could be used to explain why some similarity metrics perform better than others in a particular recommendation setting. Unfortunately, we have not yet found a principled way to improve the performance of a given similarity function based on these results. We have evaluated a distribution-based normalisation of the similarity values,

ICTIR '13, September 29 - October 02 2013, Copenhagen, Denmark

ACM 978-1-4503-2107-5/13/09 ...$15.00.
http://dx.doi.org/10.1145/2499178.2499186

Table 1: Definition of $\text{sim}(u, v)$ for two functions (cosine and Pearson). The set $I(u, v) \subset \mathcal{I}$ denotes the items involved in the computation of the similarity (see Section 2.2 for more details).

Cosine similarity	Pearson similarity
$\frac{\sum_{i \in I(u,v)} r(u,i) r(v,i)}{\sqrt{\sum_{i \in I(u,v)} r(u,i)^2} \sqrt{\sum_{i \in I(u,v)} r(v,i)^2}}$	$\frac{\sum_{i \in I(u,v)} (r(u,i) - \bar{r}(u))(r(v,i) - \bar{r}(v))}{\sqrt{\sum_{i \in I(u,v)} (r(u,i) - \bar{r}(u))^2} \sqrt{\sum_{i \in I(u,v)} (r(v,i) - \bar{r}(v))^2}}$

such that a bad-performing similarity metric is transformed according to the distribution of a better-performing metric, however the resulting performance remains unaltered with respect to the original, untransformed similarity metric.

2. NEIGHBOUR-BASED RECOMMENDER SYSTEMS

Neighbour-based CF algorithms are based on the principle that a particular user's rating records are not equally useful to make suggestions for all other users [19]. Central aspects in these algorithms are how to identify which neighbours form the best basis to generate item recommendations for the active user, and how to properly account for the information provided by them.

Neighbourhood identification selects the users who are most similar to the active user, according to a given similarity metric. In this context, the similarity of two users generally consists of finding a set of items that both users have interacted with, and examining to what degree the users displayed similar behaviours on these items. It is common practice to set a maximum number of neighbours (or a minimum similarity threshold) to restrict the neighbourhood size either for computational efficiency, or in order to avoid basing recommendations on users who are not similar enough, resulting in noisy outcomes.

Once the target user's neighbours are selected, these neighbour's preferences are usually weighted by their similarity to the active user to produce the final recommendations. A common user-based approach, for example, predicts the relevance of an item for the target user using a linear combination of the neighbours' ratings, weighted by the similarity between the target user and such neighbours [2, 27]:

$$\hat{r}(u, i) = C \sum_{v \in N_k(u)} \text{sim}(u, v) r(v, i) \quad (1)$$

Here, $\hat{r}(u, i)$ represents the predicted rating for user u and item i, C is a normalisation constant, $N_k(u)$ is the user's neighbourhood of size k, $\text{sim}(u, v)$ is a user similarity metric, and $r(v, i)$ denotes the rating given by user v to item i. A similar instantiation of this algorithm takes into account the rating deviations from user's and neighbour's rating means (denoted as $\bar{r}(u)$ for a user u) [26]:

$$\hat{r}(u, i) = \bar{r}(u) + C \sum_{v \in N_k(u,i)} \text{sim}(u, v)(r(v, i) - \bar{r}(v)) \quad (2)$$

Generally, the neighbourhood $N_k(u)$ is defined based on some user similarity metric, either by selecting the top k neighbours more similar to user u, or by choosing a threshold and restricting the neighbours to those whose similarity with u is above such threshold (here we will use the former approach). Other ways to construct the neighbourhood include exploiting the notion of 'trust', by selecting only the most trustworthy users with respect to some trust metric [25], or by using clustering algorithms, to improve the resulting system's scalability [24].

2.1 User similarity metrics

The two most used user similarity functions in the literature are the cosine and Pearson similarities. Table 1 shows their most commonly used definition, when the preferences of users are represented by their item ratings.

Cosine similarity captures the angle between both users when they are represented in the item space, a Euclidean space where dimensions correspond to items. This metric is unaffected by scalar transformations (i.e., multiplication of all the elements of the vector by the same value), although it is not invariant with respect to shifts (i.e., if a constant is added to one of the vectors involved in the computation the cosine similarity may be different).

Pearson similarity captures the correlation in terms of the rating patterns between the two users, again, represented in an item space (each dimension corresponds to an item in the collection). This metric, like the cosine, is invariant to scaling, but differently, it is also invariant to shifts or linear transformations of the data. Note that these two similarities are equivalent when the data is centered on the mean.

Some authors have reported that the performance of recommenders may change depending on which similarity metric is used, specifically, most studies report that Pearson similarity is superior to cosine [5, 20], however, as we shall show in the experimental section, there are other variables that should be considered which may have a large impact on the final performance. Inspired by [6], in the next section we present different variations of these similarity functions based on the space of items in which the users are compared, the default value used when no rating is given by a user, and the filtering threshold to decide when to ignore the similarity value.

2.2 Item selection, imputation, and filtering

In Table 1, we denote as $I(u, v)$ the subset of items $\mathcal{I}$ involved in the computation of the similarity between each pair of users u and v. Usually, this set is equal to the intersection of items rated by both users, that is, $I(u, v) = I(u) \cap I(v)$. This strategy (named as **overlap** from now on) has the advantage that it reduces the computational cost to calculate the similarity; however, as pointed out in [6], it may lead to incorrect normalisations, since it ignores the ratings that were given by only one of the two users being compared, which in sparse situations may lead to similarities computed based on a few item ratings only.

Alternatively, we may try to exploit all the items in the similarity computation, i.e., $I(u, v) = \mathcal{I}$. In this case, a default value should be imputed to those items not rated by a particular user. This strategy of imputed ratings to compute the similarity, although less common than the overlap strategy, is not new in the literature [5, 6, 29]. Varying methods have been deployed to determine the default rating value to be imputed, which can be expressed in general terms as follows: when $r(u, i)$ is unknown then $r(u, i) = \tilde{r}(u, i)$. The imputed value $\tilde{r}(u, i)$ could be modelled as a function of the corresponding user or item, where common strategies include their average rating (denoted as *default voting* in [5])

Table 2: Definition of the features based on the distribution of similarity values. Note that k is the neighbourhood size, $\mathcal{U}$ is the set of users in the community, and $\text{sim}(u,v)$ may be any arbitrary user similarity function.

Distribution features	Definition
Average neighbour similarity, $ans(k)$	$\lvert\mathcal{U}\rvert^{-1}\sum_{u\in\mathcal{U}} k^{-1}\sum_{v\in N_k(u)}\text{sim}(u,v)$
Neighbour similarity ratio, $nsr(r;k)$	$ans(k)^{-1}(ans(k)-ans(k\cdot r))$
Neighbour stability	$\frac{1}{\lvert\mathcal{U}\rvert}\sum_{u\in\mathcal{U}}\frac{\min_{v\in N_k(u)}\text{sim}(u,v)}{\max_{v\in N_k(u)}\text{sim}(u,v)}$
Stability	$\frac{1}{\lvert\mathcal{U}\rvert}\sum_{u\in\mathcal{U}}\frac{\min_{v\in\mathcal{U}\setminus u}\text{sim}(u,v)}{\max_{v\in N_k(u)}\text{sim}(u,v)}$

or a constant value not in the rating scale – i.e., 0 in the common range 1-5 – or a neutral value within that scale – i.e., a 3. Other imputed values would assume the unobserved items to be negative or positive, and thus they are usually avoided. In the following, we shall denote as **Full0** when all the items are used in the similarity computation and the imputed value is 0, as **Full3** when the imputed value is 3, and **FullAvg** when the user's average is used, that is, $\tilde{r}(u,i)=\bar{r}(u)$.

As we have presented in the previous section, similarity scores are typically used as weights in the computation of the predicted rating. However, these functions may return negative values since their range is the interval $[-1,1]$, which may produce negative predictions. To prevent this effect, some authors have proposed to use a filtering threshold τ such that, when $\text{sim}(u,v)<\tau$ the similarity is not considered; for instance, in [6] a value of $\tau=0$ is used to not take into account the negative correlations.

3. ANALYSIS OF USER SIMILARITY METRICS

Considering the variety of choices to be made, even while only looking into the most common case of neighbourhood based recommendation systems based on user-similarity, we would like to find the key characteristics that help understand when these metrics perform better for recommendation. They are defined from two perspectives: firstly, analysing the distribution of the similarity values, and secondly, exploring graph metrics computed over the graph of nearest neighbour users.

3.1 Distance distribution analysis

The distribution of the values returned by a similarity metric contains very important information about the behaviour of that metric, like its range, its most likely output and the smoothness of the transition between these outputs. Now we present some basic properties and others taken from the literature to capture different aspects of a similarity distribution, summarised in Table 2.

We start by defining the **average neighbour similarity**, which gives a notion of how close the top k neighbours are. Besides, we also propose to compute the **neighbour similarity ratio**, that compares the average neighbour similarity of two different neighbourhoods, one of size k and another of size $k\cdot r$. In this way, if the average similarity remains unchanged after enough neighbours have been considered, we could conclude that this metric is not very discriminative.

The rest of the proposed features are adapted from [4], also aiming to deal with how discriminative (or meaningful) a distance distribution is in the context of a nearest neighbour search problem. In [4], Beyer et al. define a stability factor that compares the distance to the nearest neighbour against the farthest point. In that work, the authors prove that the concept of nearest neighbour may not be meaningful in certain situations, which can be measured by this stability factor. In this paper, we adapt this concept and invert its computation since we have similarities instead of distances; we define the **stability** as the average ratio per user between the least similar user and most similar neighbour. We also restrict this definition within a neighbourhood and define the **neighbour stability** in the same way as before where the least similar user is contrained to belong to the neighbourhood (for specific details, see the corresponding formula in Table 2).

Also in [4] the authors use the amount of queries having at least half of the data within a factor of the nearest neighbour as a measure of the contrast (or quality) of the answers obtained by the distance function. Here, we generalise this concept and adapt it to the context of similarity functions, by defining a feature that depends on two parameters: the percentage of data n and a factor f. Then, we compute the **quality** $q(n,f)$ of a similarity function as follows:

$$\begin{aligned} q(n,f) &= \frac{1}{\lvert\mathcal{U}\rvert}\left\lvert\{u\in\mathcal{U}: g_f(u,s_u)\geq n\}\right\rvert \\ g_f(u,s) &= \frac{1}{\lvert\mathcal{U}\rvert}\left\lvert\left\{v\in\mathcal{U}: f<\frac{s}{\text{sim}(u,v)}<f+1\right\}\right\rvert \\ s_u &= \max_{v\in N_k(u)}\text{sim}(u,v) \end{aligned}$$

Therefore, this feature lets us compute the amount of users for which the similarity function has ranked at least n percentage of the whole community within a factor f of the nearest neighbour's similarity. If we are interested in the cumulative percent of users with this property, we could just modify function g_f (removing the restriction that the ratio of similarities should be smaller than $f+1$) or we could simply accumulate this quantity for consecutive values of f.

Table 3: Definition of the features based on the nearest neighbour graph. Note that this graph takes as vertices V the set of users in the system $\mathcal{U}$, and the edges E_k^s depend on the neighbourhood size k and the actual similarity metric s. Besides, $sp(u,v)$ and $\overline{sp}(u)$ denote the shortest path and its average, $cc(u)$ represents the local clustering coefficient for a user, and $\deg^+(u)$ and $\deg^-(u)$ correspond to the out-degree and in-degree of a user. Finally, $M[X]$ denotes the median of the random variable X.

Graph features	Definition
Average graph distance	$\lvert\mathcal{U}\rvert^{-1}\sum_{u\in\mathcal{U}}\overline{sp}(u)$
Clustering coefficient	$\lvert\mathcal{U}\rvert^{-1}\sum_{u\in\mathcal{U}}cc(u)$
Graph density	$\frac{\lvert E_k^s\rvert}{\lvert V\rvert(\lvert V\rvert-1)}$
Graph diameter	$\max_{(u,v)\in\mathcal{U}\times\mathcal{U}}sp(u,v)$
Maximum graph distance	$\max_{u\in\mathcal{U}}\overline{sp}(u)$
Median in-degree	$M_{u\in\mathcal{U}}[\deg^-(u)]$
Median out-degree	$M_{u\in\mathcal{U}}[\deg^+(u)]$

3.2 Nearest neighbour graph analysis

A pure metric representation like the one presented in the previous section has an obvious interpretation from the similarity metric viewpoint. Now, we propose to build the graph associated with the top k nearest neighbours and study its topological properties, not taking into account the specific similarity values (already included in the distance distribution analysis) but exploiting the binary relation of whether a user does or does not belong to a neighbourhood.

In this context, nearest neighbour graphs are defined, given a similarity metric s and a number k, as the directed graph (V, E_k^s) where $V = \mathcal{U}$ and each edge $e \in E_k^s$ connects a user u with every other neighbour in her top k most similar users with respect to s [10, 17]. Obviously, even though the similarity function may be symmetric, the top k nearest neighbour relationship is asymmetric, which explains why we need a directed graph.

Over this graph, we propose to compute standard metrics from link analysis, aiming to summarise the information encoded in it. Among the wide range of metrics available, we propose herein to use those related with the concept of distance in the graph.

The first concept we want to exploit is that of the shortest path in a graph, $sp(x, y)$ between two nodes x and y. We use the average shortest path of a node ($\overline{sp}(x)$) to compute the **maximum graph distance** and the **average graph distance** of a particular nearest neighbour graph, in the first case by taking the maximum of such distances over every user, and in the second by computing the average of these distances. Table 3 contains the specific formulations of these metrics.

The **diameter** of a graph is defined in a similar way to that of the maximum graph distance, but instead of computing the average for every user and then finding the maximum of these values, it takes the maximum shortest path for every pair of users. The graph **density** measures, on the other hand, the number of edges compared to the maximum possible number of edges, and although this metric is not really related with the concept of distance it will serve us as a baseline.

In order to measure how densely connected are the users between them, we use the **clustering coefficient** which measures the probability that two neighbours of a user are neighbours of each other. For this computation we average each local clustering coefficient $cc(x)$ for every node x, this is calculated as the number of actual edges between the neighbours of node x divided by the maximum amount of connections that could exist [30].

Finally, we also include in our analysis the in-degree and out-degree of each node in the graph (denoted as $\deg^-(x)$ and $\deg^+(x)$ respectively). The **out-degree** is constant for every node (i.e., $\deg^+ = k$), and it is included here just as a proof of concept that it should not be meaningful. For the **in-degree**, on the other hand, we compute its median value since the mean in-degree would be constant due to the handshaking lemma. This lemma (also known as degree sum formula) states that $\sum_u \deg^+(u) = \sum_u \deg^-(u)$, and since $\deg^+(u) = k$ for any user u, the mean or average in-degree would be equal to k.

4. EXPERIMENTS

To empirically compare the explanatory power of the proposed user similarity properties, we first show several instantiations of the similarity metrics presented in Section 2 that present different performance results, then, we explore and analyse which of the proposed properties in Section 3 are able to capture the usefulness of such similarity metrics when used as the main components of user-based recommendation algorithms. With these experiments, we aim to answer the two research questions stated at the beginning of this paper; more specifically, Section 4.2 addresses **RQ1** and Section 4.3 deals with **RQ2**.

The results reported in this section have been obtained using the publicly available dataset[1] called *MovieLens 1M*. This dataset contains 6,040 users, 3,900 items and 1,000,209 ratings. We performed a 5-fold cross validation retaining 80% of the data for training, and the rest for testing. Some preliminary experiments were carried out in the smaller dataset called *MovieLens 100K* and we obtained comparable results.

The methodology used in our evaluation corresponds to the one described by Koren in [22], where for each user a number of not relevant items (unrated by this user in the training and testing sets) is randomly selected (100 in our case), and then, for each highly relevant item in the testing split (i.e., those rated as 5), a ranking is generated by predicting a score for both this item and the other (not relevant) items. Then, the performance of this ranking is measured using, in this case, the *trec_eval* program[2]. In this way, standard retrieval metrics such as precision, normalised Discounted Cumulative Gain (nDCG) or Mean Reciprocal Rank (MRR) could be used. In our experiments we use MRR because each ranking only contains one relevant item, as described before.

We have used the JUNG library[3] for most of the computations related with the metrics based on nearest neighbour graphs.

4.1 Performance comparison of user similarity metrics

Table 4 shows the MRR values of different variations of the cosine and Pearson similarity metrics when used inside a standard user-based recommendation technique (denoted as UB and corresponding to the Equation 1) and when the user and neighbour deviations are considered (UBMeans, Equation 2), in both cases using 50 neighbours. Here, we have experimented with the different imputation strategies described in Section 2.2. We carried out experiments with filtered correlations (thresholding by τ, see Section 2.2) as well, but they produced results very similar to the unfiltered ones (probably because the neighbourhood size is too small to observe a significant change), and thus these results will be omitted from discussion in the paper.

A first observation we can derive from Table 4 is that similarity metrics have an equivalent behaviour on different recommendation techniques. This makes sense since both techniques only change the way they aggregate the rating and similarity information. However, we can observe that the results are slightly different for each method. In particular, the trend is consistent for both settings, although the values of UBMeans are slighly lower than those of UB. We find that we cannot easily determine which similarity metric performs better, since any Pearson variation is better than cosine when using overlap, while full cosine with an imputed value of 0 (Full0) is better than Pearson; and, using overlap leads to inferior results for all settings. We should emphasise that the overlap strategy for computing the similarities saves computation time and space, and perhaps this explains why

[1]Available at http://www.grouplens.org/node/73
[2]Available at http://trec.nist.gov/trec_eval
[3]Available at http://jung.sourceforge.net

Table 4: Performance results (mean reciprocal rank) for different combinations of recommendation methods, similarity metrics and imputation techniques. ↑ and ↓ denote the best and worst combinations for each pair of recommender and similarity metric.

		Imputation			
Recommender	Similarity	Full0	Full3	FullAvg	Overlap
UB	Cosine	0.511 ↑	0.392	0.333	0.187 ↓
	Pearson	0.451	0.443	0.456 ↑	0.220 ↓
UBMeans	Cosine	0.471 ↑	0.303	0.269	0.156 ↓
	Pearson	0.371	0.368	0.431 ↑	0.192 ↓

Table 5: Spearman correlation values between the similarity features and the MRR values of the different combinations of recommendation methods, similarity metrics, and imputation techniques. All values are statistically significant ($p < 0.05$).

Distribution features	Correlation	Graph features	Correlation
Average neighbour similarity	−0.97	Average graph distance	−0.77
Neighbour similarity ratio, $nsr(10)$	0.88	Clustering coefficient	−0.21
Neighbour stability	−0.92	Graph density	0.29
Stability	−0.33	Graph diameter	0.70
Quality, $q(0.5, 1)$	−0.74	Maximum graph distance	−0.51
Quality, $q(0.5, 2)$	0.26	Median in-degree	0.85
Quality, $q(0.5, 3)$	0.32	Median out-degree	NA

it has been the preferred method in the literature (see [9, 19, 1]) and in some public implementations (like Mahout[4] and MyMediaLite[5]), even though it was shown in [6] already that overlapping distances do not properly capture the similarity between the profiles, thus obtaining compatible results with those presented here.

4.2 Performance analysis of user similarity metrics

In this section, we analyse the features defined in Section 3 by measuring the correlation between the values obtained by these features and the performance of each combination of similarity and imputation strategy. To focus on the top most similar users, we build the neighbour graphs and distributions considering only the top 5 users more similar (i.e., $k = 5$).

Table 5 summarises the correlations between the performance values and the properties of all the methods described before using Spearman's coefficient, which is a well-known correlation function able to capture non-linear relationships between the variables of interest, in this case the similarity features and their corresponding performance.

The results presented in Table 5 evidence that some of these features are very correlated with the final performance of the recommender that uses a particular similarity metric. For instance, the *average neighbour similarity* and the *average graph distance* correlate negatively with performance, representing that the higher these features the worse the performance. As described in Section 3, this is related, in the first case, to the average neighbours' similarity, and in the second to the average shortest path length.

Having short paths in the nearest neighbour graph is related with a high value of the *quality* feature, since most of the users are within a small factor of the neighbour's distance (this explains the similar correlation obtained for quality, in particular for the case $q(0.5, 1)$). The parameters of the quality feature correspond to those suggested in [4], where the authors examine the percentage of queries in which at least half the data points ($n = 0.5$) were within some factor f of the nearest neighbor.

A small average neighbours' similarity could be produced by either a metric that deliberately outputs low similarity values, or by a metric whose values distinguish strongly within the neighbourhood (a very high value for the very top neighbours, and much lower values for the rest). Thus, we need to look at the rest of the features to distinguish between these situations.

The *stability* feature and its constrained version (*neighbour stability*) also show negative correlations. These results are consistent with the theoretical analysis developed in [4], where the authors state that when the stability is high it represents that the distance can provide plenty of contrast between the nearest and the farthest object. In our context, this rationale should be inverted, since we deal with similarities instead of distances (note its definition in Table 2). Thus, larger values (closer to 1.0) correspond to situations where the farthest and the closest neighbours have an equivalent similarity value, whereas when the stability is closer to 0 or negative it means the similarity metric is more discriminative. Lower stability values thus correspond to better performing similarity metric, as reflected in the negative correlations observed.

Other features such as the *graph density*, *graph diameter*, *median in-degree*, and *neighbour similarity ratio* correlate positively with performance. As expected, the predictive power of the density metric is not strong, because it is not actually related with the concept of distance in the graph. Note that correlations for the mean and median out-degree are not applicable since this feature is constant for every user (the out-degree for every user is exactly k).

Now let us illustrate these results with a detailed example of the distributions for a well and a bad performing similarity metrics. Figure 2 shows an example of each situation, we can observe here that the graph corresponding to the cosine with overlap (right, bad performing) hardly discriminates between the neighbours, since most of its values are closer to 1, whereas the full cosine with an imputed value of 0 (left, better performing) shows a limited amount of very similar points. This specific characteristic of the distribution is captured to some extent, as we described in Section 3.1, by most of the proposed features analysed herein, such as average neighbour similarity (0.36 for the better performing, 0.99 for the other), neighbour stability (0.88 and 0.99), and graph diameter (31 and 10).

[4]Available at http://mahout.apache.org
[5]Available at http://mymedialite.net

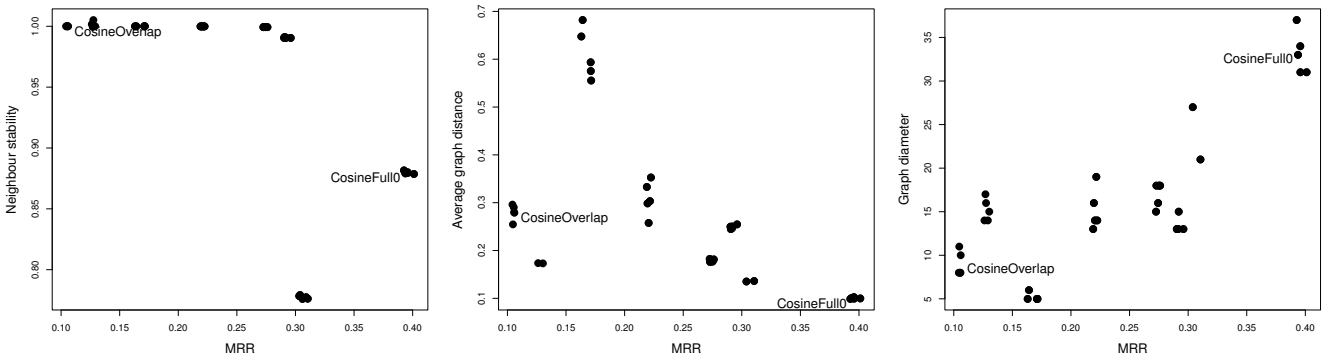

Figure 1: Scatterplot based on the performance results (MRR) of different versions of recommendation methods (with $k = 5$) and three of the similarity features presented before (neighbour stability, average neighbour distance, and graph diameter). Note that these plots correspond to the correlations presented in Table 5, and the MRR values are slightly different to those in Table 4 since a different k is used.

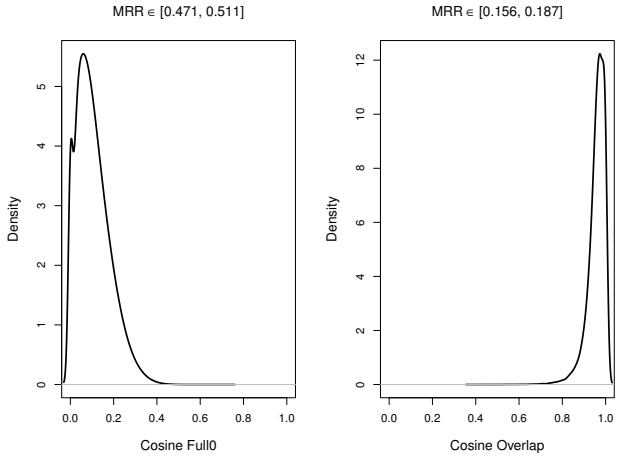

Figure 2: Distance distribution for two similarity metrics with different performance results. Note how the cosine similarity with an imputed value of 0 (left, good performance) is more discriminative than the distribution for the cosine with overlap, since it produces a smaller amount of high similarities.

A more general trend is observed in Figure 1, which shows the scatterplots for three of these similarity features. Note how easy we can distinguish a bad performing similarity metric (cosine with overlap) from a better metric (cosine Full0) by looking at the values of these features. This is in contrast with the typical results obtained in query performance literature; we discuss this aspect in Section 5.

4.3 Adjusting similarity metric distributions

Based on the results presented before, we aim to improve the performance of user similarity metrics by adjusting their values. As a first approach, we transform their distributions to the ones that show better results. For this purpose, we consider a distribution-based normalisation [13, 14] to transform the values of similarity functions based on some ideal distributions (i.e., those that show better performance).

The rationale behind this idea is that, according to the correlation values found in the previous section, if we are able to build a user similarity function with a high neighbour similarity ratio, very few neighbours within a factor of 1 of the nearest neighbour, and a small average distance between them – by taking just three of the strongest features –, such similarity function may show a good performance.

Table 6: Performance results (using MRR) for a UB recommender when similarity metrics are normalised according to the best performing similarity function (Cosine Full0). Two neighbourhood sizes are tested: 5 and 50 neighbours. No statistically difference observed ($p < 0.01$).

	Top 5		Top 50	
	Orig	Norm	Orig	Norm
Cosine Full0	0	-	0.511	-
Cosine Full3	0.285	0.283	0.392	0.388
Cosine FullAvg	0.184	0.182	0.333	0.328
Cosine Overlap	0.108	0.108	0.187	0.181
Pearson Full0	0.425	0.426	0.451	0.445
Pearson Full3	0.351	0.351	0.443	0.443
Pearson FullAvg	0.384	0.382	0.456	0.454
Pearson Overlap	0.160	0.160	0.220	0.217

However, it is very difficult to build such a function in a formal, principled way. Because of this, we propose to normalise an (already existing) bad-performing similarity function and transform its values based on the distribution of a better-performing (or ideal) similarity function. In this way, the new similarity metric should behave more like the ideal distribution; although this transformation by itself does not guarantee that the normalised similarity inherits all the features from the ideal similarity. Results reported in [13, 14] demonstrate that this method can, in principle, transform the scores automatically without manual intervention and lead to improved results.

Table 6 shows the performance of the original and of the transformed similarity metrics using as ideal distribution the best performing combination presented in Table 4, that is, the full cosine metric with an imputed value of 0. We also show the results for two different neighbourhood sizes: 5 because the feature analysis (Section 4.2) was focused on the closest neighbours and thus fewer neighbours were considered (i.e., $k = 5$), and 50 because this is a typical neighbourhood size used in previous work [9] and it also corresponds to the size used in the results reported in Section 4.1.

We can observe in Table 6 that the performance of the transformed similarity functions remains almost unchanged, although the results are better when only the top 5 neighbours are considered. This is probably because, as mentioned before, the transformation does not ensure that the corresponding distribution features computed over the transformed values have the desired values, such as a lower average neighbour similarity. One possible reason for this is

Table 7: Performance results (using MRR) for UB recommender when the Pearson similarity is normalised according to the best performing Pearson similarity function (Pearson FullAvg).

	Top 5		Top 50	
	Orig	Norm	Orig	Norm
Pearson FullAvg	0.384	-	0.456	-
Pearson Full0	0.425	0.385	0.451	0.431
Pearson Full3	0.351	0.332	0.443	0.412
Pearson Overlap	0.160	0.159	0.220	0.210

that the distribution-based normalisation is transforming a global aspect of the similarity function without considering other *local effects* (e.g., per user) such as the average neighbour similarity or the stability; moreover, it is difficult for such approach to modify the original features based on the nearest neighbour graph, which, as already discussed, are also related to the final performance of the metric.

As an additional check, and since it seems cosine similarities decrease their performance when normalised using a cosine variation as ideal distribution, we have normalised the Pearson similarities with respect to the best Pearson similarity metric reported in Table 4, which corresponds to the full Pearson similarity with the user's average as imputed value. In this case the results are worse than before (see Table 7), except for Pearson overlap and top 5, which suggests that the target and ideal distributions should be very different in order to not have at least a negative effect, like in the case of the overlap strategy (either with cosine or Pearson as ideal distributions).

5. RELATED WORK

An empirical comparison of different strategies to build neighbourhoods in collaborative filtering was developed in [19]. In addition, an analysis of the performance of similarity metrics was presented in [28] and [6]. However we should note that the evaluation metrics and methodologies have changed with respect to these papers where only error-based metrics – not ranking-based like here – were used. Besides, although in this paper we present an empirical analysis of different metrics, this is not its main goal, in contrast with the aforementioned papers.

Regarding the analysis of the user similarity metrics, this paper has connections with previous theoretical papers in the fields of databases [4] (from which we adapted the concept of stability and quality) and [21, 8, 11]. Nearest neighbour graphs have also been used in semantic image search [17, 18], where some properties of such graphs are analysed from a topological point of view.

This work may also be observed from the perspective of performance prediction in Information Retrieval [7, 16] and recommendation [3], where different functions have been proposed to predict the final performance of the query (in retrieval) or the target user (in recommendation). In our case, the features we have defined are assumed to be inherent to the components of the recommender system, and thus, they measure a global property of the algorithm and its corresponding similarity metric.

Furthermore, measurements of performance predictors are usually defined at a query or user level, whereas the features analysed in this paper are global measurements related to the whole recommender system. This difference between the techniques may be the key factor for obtaining so different results – whereas in query performance the features hardly ever correlate, here we have found very strong correlations.

6. CONCLUSIONS AND FUTURE WORK

The performance of neighbour-based recommender systems changes depending on the user similarity metric used to build its neighbourhood and how it is computed. In this paper, we have explored this issue first by computing different variations of two similarity metrics (cosine and Pearson) and then by proposing several features computed on the output of each similarity variation, either by exploiting its distribution or the corresponding nearest neighbour graph.

We have found that some of these features present a strong predictive power with respect to the performance of a neighbour-based recommender system using such similarity metric. More specifically, the more successful features are those dealing with a comparison of the specific similarity value between top and farthest neighbours (like average neighbour similarity and neighbour stability) or top neighbours and less similar users (such as average graph distance, neighbour similarity ratio, and quality). These results are compatible with others found in the database literature where the stability of a metric is related with its ability to properly discriminate good from bad neighbours. To put these insights to the test, we have transformed some bad-performing similarities such that their distribution is more similar to that of better-performing metrics, but the results are not conclusive (the performance barely changes) and more effort is needed in this direction.

In the future, we aim to develop a general methodology to analytically and experimentally diagnose the weaknesses of a neighbour-based recommender system, similar to what was proposed recently in Information Retrieval [12]. We plan to explore other similarity metrics like the Spearman correlation coefficient, that traditionally has been observed as a bad performing metric, which could be misleading since most of the variations presented here, to the best of our knowledge, were not tested for this metric. Furthermore, we want to emphasise that the evaluation metrics and methodologies have changed with respect to the classic papers where most of these observations were first proposed, which mainly used error-based metrics – instead of ranking-based. Additionally, we also plan to analyse the effect of variations in similarity metrics for item-based recommender systems, to check whether those features having a strong predictive power in the user-based scenario are also useful in the item-based context.

7. ACKNOWLEDGEMENTS

This work was carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no.246016, and has partially been supported by the Dutch national program COMMIT.

8. REFERENCES

[1] 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.

[2] C. C. Aggarwal, J. L. Wolf, K.-L. Wu, and P. S. Yu. Horting hatches an egg: a new graph-theoretic approach to collaborative filtering. In *the fifth ACM SIGKDD international conference*, ECSCW'01, pages 201–212, New York, New York, USA, 1999. ACM Press.

[3] A. Bellogín, P. Castells, and I. Cantador. Predicting the Performance of Recommender Systems: An Information Theoretic Approach. In *ICTIR*, volume 6931 of *Lecture Notes in Computer Science*, pages 27–39, Berlin, Heidelberg, 2011. Springer Berlin / Heidelberg.

[4] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? In C. Beeri and P. Buneman, editors, *Database Theory - ICDT'99*, volume 1540 of *Lecture Notes in Computer Science*, chapter 15, pages 217–235. Springer Berlin Heidelberg, Berlin, Heidelberg, Jan. 1999.

[5] J. S. Breese, D. Heckerman, and C. Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 43–52, July 1998.

[6] M. Clements, A. P. de Vries, J. A. Pouwelse, J. Wang, and M. J. T. Reinders. Evaluation of Neighbourhood Selection Methods in Decentralized Recommendation Systems. In *Workshop on Large Scale Distributed Systems for Information Retrieval (LSDS-IR)*, 2007.

[7] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '02, pages 299–306, New York, NY, USA, 2002. ACM.

[8] A. P. de Vries, N. Mamoulis, N. Nes, and M. L. Kersten. Efficient k-NN search on vertically decomposed data. In *SIGMOD Conference*, pages 322–333. ACM, 2002.

[9] C. Desrosiers and G. Karypis. A Comprehensive Survey of Neighborhood-based Recommendation Methods. In *Recommender Systems Handbook*, chapter 4, pages 107–144. Springer, Boston, MA, 2011.

[10] D. Eppstein, M. Paterson, and F. F. Yao. On nearest-neighbor graphs. *Discrete & Computational Geometry*, 17(3):263–282, 1997.

[11] R. Fagin, R. Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *SIGMOD Conference*, pages 301–312, 2003.

[12] H. Fang, T. Tao, and C. Zhai. Diagnostic Evaluation of Information Retrieval Models. *ACM Trans. Inf. Syst.*, 29, Apr. 2011.

[13] M. Fernández, D. Vallet, and P. Castells. Probabilistic Score Normalization for Rank Aggregation. In *28th European Conference on Information Retrieval (ECIR 2006)*, pages 553–556. Springer Verlag Lecture Notes in Computer Science, Vol. 3936, Apr. 2006.

[14] M. Fernández, D. Vallet, and P. Castells. Using historical data to enhance rank aggregation. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 643–644, New York, NY, USA, Aug. 2006. ACM.

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

[16] C. Hauff, D. Kelly, and L. Azzopardi. A comparison of user and system query performance predictions. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 979–988, New York, NY, USA, 2010. ACM.

[17] D. Heesch and S. Rüger. NN^k networks for Content-Based image retrieval. In *Advances in Information Retrieval*, volume 2997 of *Lecture Notes in Computer Science*, pages 253–266. Springer Berlin Heidelberg, 2004.

[18] D. Heesch and S. Rüger. Image browsing: Semantic analysis of NN^k networks. In *Image and Video Retrieval*, volume 3568 of *Lecture Notes in Computer Science*, pages 609–618. Springer Berlin Heidelberg, 2005.

[19] J. Herlocker, J. A. Konstan, and J. Riedl. An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms. *Inf. Retr.*, 5(4):287–310, Oct. 2002.

[20] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. 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, pages 230–237, New York, NY, USA, 1999. ACM.

[21] A. Hinneburg, C. C. Aggarwal, and D. A. Keim. What is the nearest neighbor in high dimensional spaces? In *VLDB*, pages 506–515, 2000.

[22] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 426–434, New York, NY, USA, 2008. ACM.

[23] H. Ma, I. King, and M. R. Lyu. Effective missing data prediction for collaborative filtering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 39–46, New York, NY, USA, 2007. ACM.

[24] M. O'Connor and J. Herlocker. Clustering items for collaborative filtering. In *ACM SIGIR Workshop on Recommender Systems*, 1999.

[25] J. O'Donovan and B. Smyth. Trust in recommender systems. In *Proceedings of the 10th international conference on Intelligent user interfaces*, IUI '05, pages 167–174, New York, NY, USA, 2005. ACM.

[26] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *CSCW*, pages 175–186, 1994.

[27] U. Shardanand and P. Maes. Social information filtering: algorithms for automating "word of mouth". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 210–217, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[28] P. Symeonidis, A. Nanopoulos, A. Papadopoulos, and M. Y. Collaborative filtering : Fallacies and insights in measuring similarity. In *Proceedings of the 10th PKDD Workshop on Web Mining (WEBMine'2006)*, pages 56–67, Berlin, 2006.

[29] J. Wang, A. P. de Vries, and M. J. T. Reinders. Unified relevance models for rating prediction in collaborative filtering. *ACM Trans. Inf. Syst.*, 26(3):1–42, June 2008.

[30] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998.