Improving collaborative filtering's rating prediction coverage in sparse datasets by exploiting the 'friend of a friend' concept

Dionisis Margaris

Department of Informatics and Telecommunications,
University of Athens,
Athens, Greece
Email: margaris@di.uoa.gr

Costas Vassilakis*

Department of Informatics and Telecommunications,
University of the Peloponnese,
Tripoli, Greece
Email: costas@uop.gr
*Corresponding author

Abstract: In collaborative filtering users with highly similar tastes are termed 'near neighbours' and recommendations for a user are based on her 'near neighbours' ratings. However, for a number of users no near neighbours can be found, a problem termed as the 'grey sheep' problem. This problem is more intense in sparse datasets. In this work, we propose an algorithm for alleviating this problem by exploiting the friend of a friend (FOAF) concept. The proposed algorithm, CFfoaf, has been evaluated against eight widely used sparse datasets and under two widely used collaborative filtering correlation metrics, namely the Pearson Correlation Coefficient and the Cosine Similarity and has been proven to be particularly effective.

Keywords: collaborative filtering; recommender systems; big data; sparse datasets; similarity transitivity; friend-of-a-friend; Pearson correlation coefficient; cosine similarity; prediction accuracy; coverage; evaluation.

Reference to this paper should be made as follows: Margaris, D. and Vassilakis, C. (2020) 'Improving collaborative filtering's rating prediction coverage in sparse datasets by exploiting the 'friend of a friend' concept', *Int. J. Big Data Intelligence*, Vol. 7, No. 1, pp.47–57.

Biographical notes: Dionisis Margaris is a Postdoc Researcher at the Department of Informatics and Telecommunications of the University of Athens and an Adjunct Assistant Professor in the Department of Digital Systems at the University of the Peloponnese, in the area of Information Systems. He received his Master's degree and his PhD from the Department of Informatics and Telecommunications of the University of Athens. He has published more than 30 papers in international journals and conferences and his research interests include recommender systems, information systems, computer programming, databases systems and social networks.

Costas Vassilakis is a Professor at the Department of Informatics and Telecommunications of the University of the Peloponnese, in the area of Information Systems. He received his degree and his PhD from the Department of Informatics, University of Athens. He has published more than 170 papers in international journals and conferences. He has also participated in more than 30 international and national projects, mainly involving information and software systems. His research interests include information systems, semantic web, recommender systems, cultural informatics, service-oriented architectures as well as information presentation aspects.

This paper is a revised and expanded version of a paper entitled 'Improving collaborative filtering's rating prediction coverage in sparse datasets by exploiting user dissimilarity' presented at DataCom 2018, Athens, Greece, 12–15 August 2018.

1 Introduction

Collaborative filtering (CF) computes personalised recommendations, by taking into account users' past likings and tastes, in the form of ratings entered in the CF ratings database. User-user CF algorithms firstly identify people having similar tastes, by examining the resemblance of already entered ratings; for each user *u*, other users having highly similar tastes with *u* are designated as *u's nearest neighbours* (NNs). Afterwards, in order to predict the rating that u would give to an item *i* that *u* has not reviewed yet, the ratings assigned to item *i* by *u*'s NNs are combined (Balabanovic and Shoham, 1997), under the assumption that users are highly likely to exhibit similar tastes in the future, if they have done so in the past as well (Ekstrand et al., 2011; Yu et al., 2004). Analogous practices are followed in item-item CF algorithms, where the first step is to locate items that are similarly rated by users.

Formally, the likeness of ratings between two users is expressed using a correlation coefficient, the range of which is typically either [–1, 1] or [0, 1], with higher values denoting greater likeness.

Under this common scheme, only direct resemblance between two users is considered in the formulation of the set of NNs; therefore, users that have not rated any item in common cannot be used as NNs with each other. In this paper, we explore the exploitation of the transitivity concept in user similarity, in order to broaden the set of NNs that contribute in the formulation of rating predictions and recommendations for each user, and thus tackle the 'grey sheep' problem. This exploitation of the transitivity concept is analogous to the transitive usage of the 'knows' relationship in FOAF ontologies (Brickley, 2005) or the use of the 'trusts' relationship in trust networks (Jøsang and Pope, 2005). Effectively, we start from the observation that NNs are grouped together by the semantics of *homophily* (Lazarsfeld and Merton, 1954; McPherson and Smith-Lovin, 1987), i.e., bonding between NNs is established due to their similarity. Subsequently, taking into account that homophily-based relationships have been shown to be transitive ('a friend of a friend is a friend') (Hoff, 2008; De la Haye et al., 2011), we investigate models of NN transitivity, according to which NN relationships can be derived and used in the rating prediction process.

To this end, in this paper:

1 we introduce the concept of *FOAF NN* (NNF), i.e., users that are considered as NNs due to the fact that they are NNs to some third user *u'*; furthermore

2 we investigate how we can incorporate the NNF information to the rating prediction computation process, in order to improve the prediction coverage of CF recommender systems (RSs), while maintaining the quality of rating predictions. The incorporation of the NNF aspect into CF leads to a novel algorithm, termed as CF_{foaf}.

To validate our approach, we present an extensive evaluation, comparing the presented algorithm against the one presented in Margaris and Vassilakis (2018a), which incorporates the concept of negative neighbours, i.e., users with negative correlation between them, in the recommendation process, using both the Pearson correlation coefficient (PCC) as well as the cosine similarity (CS) as similarity metrics (Herlocker et al., 2004).

It is worth noting that the proposed technique can be combined with other algorithms that have been proposed for improving rating prediction accuracy, recommendation quality or prediction coverage in CF-based systems, including clustering techniques (Gong, 2010; Margaris et al., 2015) of social network data (Bakshy et al., 2012; Margaris et al., 2016, 2017), pruning of old user ratings (Margaris and Vassilakis, 2016; 2017) or hybrid filtering algorithms (Vozalis et al., 2009).

The rest of the paper is structured as follows: Section 2 overviews related work, while Section 3 presents the proposed technique. Section 4 reports on the methodology for tuning the algorithm operation, while Section 5 evaluates the proposed technique using seven contemporary datasets. Finally, Section 6 concludes the paper and outlines future work.

2 Related work

While the accuracy of CF-based systems is a topic that has attracted considerable research efforts (Yu et al., 2004; Dias and Fonseca, 2013; Margaris et al., 2018), research on CF-based systems' coverage is relatively limited. Vozalis et al. (2009) present 'Item HyCov', a filtering algorithm which combines the strengths of two popular CF approaches, item-based CF and user-based CF, into a feature combination hybrid. 'Item HyCov' deals with low prediction coverage, a problem especially present in sparse datasets; however, it has been tested using only one MovieLens dataset (MovieLens, 2018; Harper and Konstan, 2015), namely the 'MovieLens 100K' dataset, whose density index (computed as #ratings/#users/#items) is 6.3%, being 1.5 to 38 times more dense than other MovieLens datasets (MovieLens, 2018), while in relation to contemporary datasets, e.g., the Amazon datasets (McAuley et al., 2015a, 2015b), its density index is 2-4 orders of magnitude higher. Moreover, the 'Item HyCov' algorithm needs extra storage space and extra preprocessing time for the hybrid combination step, as well as continuous updates.

The work in Poirier et al. (2010) reduces the sparsity of the user-item rating matrix by computing virtual ratings based on textual user opinions, through the aggregation of the sentiments of all of the opinion words in each review. Similarly, in Moshfeghi et al. (2011), emotions are extracted from textual reviews and are used to determine the probability that a user will like an item. Both these approaches increase coverage; however they necessitate the existence of textual reviews, which are not always available; on the contrary, the algorithm presented in this paper does not impose this requirement.

Pham et al. (2011) present a clustering approach to CF recommendation technique that instead of using rating data, they use social relationship between users to identify their neighbourhoods. A complex network clustering technique is applied on the social network of users to find the groups of similar users and after that, the traditional CF algorithms can be used to efficiently generate the recommendations. Although this approach improves CF coverage, it is based on information sourced from a social network, which is not always available.

Matrix factorisation techniques constitute an alternative approach to computing rating predictions for users. As noted in Margaris and Vassilakis (2017), matrix factorisation-based techniques *always* produce a prediction for a user's *u* rating on an item *i*, through the formula (Koren et al., 2009)

$$\hat{r}_{ui} = q_i^T * p_u \tag{1}$$

where q_i^T captures the relationship between item *i* and the vector of latent factors identified by the matrix decomposition process and p_u reflects the relationship between user *u* and the latent factors. However, users who rate only a small portion of items may not get proper recommendations, and items with few ratings may not be recommended well (Wen et al., 2014), since predictions involving users or items having very few ratings degenerate to a dataset-dependent constant value (Margaris and Vassilakis, 2017). This is reflected into the rating prediction accuracy of the algorithm. To address this issue, Guan et al. (2017) have proposed an enhanced SVD model, which incorporates the classic matrix factorisation algorithms with ratings completion inspired by active learning. In the same paper, the multi-layer ESVD is introduced, which learns the model iteratively to further improve the prediction accuracy.

Margaris and Vassilakis (2018a) propose an algorithm that incorporates, in the rating prediction computation process, users with negative correlation to the user for whom the rating prediction is being computed, in order to improve coverage in sparse datasets, archiving coverage increases ranging 5.1% to 14.8%, with an average of 11.6%, while attaining at the same time an average MAE improvement of 0.74%.

The present paper advances the state-of-the-art regarding coverage increase in the context of sparse datasets, by introducing an algorithm that significantly leverages CF coverage, while at the same time reducing CF rating prediction errors; this behaviour is proven consistent under both correlation metrics and has been validated using eight contemporary and widely used datasets.

3 The proposed algorithm

In CF, predictions for a user *U* are computed based on *U*'s NNs, i.e., a set of users that have rated items similarly to *U*. The similarity metric between two users *U* and *V* is typically based on the PCC, which is expressed as:

$$sim\_p(U,V) = \frac{\sum_k \left(r_{U,k} - \overline{r_U}\right) * \left(r_{V,k} - \overline{r_V}\right)}{\sqrt{\sum_k \left(r_{U,k} - \overline{r_U}\right)^2 * \sum_k \left(r_{V,k} - \overline{r_V}\right)^2}} \tag{2}$$

where *k* ranges over items that have been rated by both *U* and *V*, while $\overline{r_U}$ and $\overline{r_V}$ are the mean values or ratings entered by users *U* and *V*, respectively. Then, for user *U*, his NN users NN_U are selected, out of the users with whom a positive similarity has been computed.

Similarly, the cosine similarity metric is expressed as:

$$sim\_cs(U,V) = \frac{\sum_k r_{U,k} * r_{V,k}}{\sqrt{\sum_k \left(r_{U,k}\right)^2} * \sqrt{\sum_k \left(r_{V,k}\right)^2}} \tag{3}$$

As noted in McAuley (2015b), CS between users does not perform well when only non-negative ratings are available; in this case ratings in the similarity computation process must be transformed so that values are negative for ratings below the centre of the rating scale and positive for ratings above the centre. In this work, we adopt this approach.

Subsequently, for computing a rating prediction $p_{U,i}$ for the rating that user *U* would assign to item *i*, the following formula is applied:

$$p_{U,i} = \overline{r_u} + \frac{\sum_{V \in NN_u} sim(U,V) * \left(r_{V,i} - \overline{r_V}\right)}{\sum_{V \in NN_u} sim(U,V)} \tag{4}$$

The proposed algorithm introduces the concept of FOAF NN (NNF), where two users *W* and *X* are designated as NNF by virtue of being NNs to a third user *Y*. More specifically, we target the case where users *W* and *X* have no ratings in common, i.e., *ratings*(*W*) ∩ *ratings*(*X*) = ∅ and for these users there exists a third user *Y* for which *Y* ∈ *NN*(*W*) ∧ *Y* ∈ *NN*(*X*).

Then, we can consider *W* and *X* as *NNFs* and consequently we can assign a similarity metric *NNF_sim* to users *W* and *X*, and use this metric in place of the (uncomputable due to the lack of common ratings between *W* and *X*) *sim*(*U*, *V*) metric in formula (4). In this respect, we can exploit the ratings of user *W* in the process of computing prediction ratings for user *X*, and vice versa. In this respect, we can effectively broaden the NN set of users through the inclusion of NNFs, and this broadening leads in turn to the limitation of the 'grey sheep; phenomenon. The rationale behind this approach is based on the transitivity of the 'similar' relationship, according to which *similar*(*W*, *Y*) ∧ *similar*(*Y*, *X*) ⇒ *similar*(*W*, *X*).

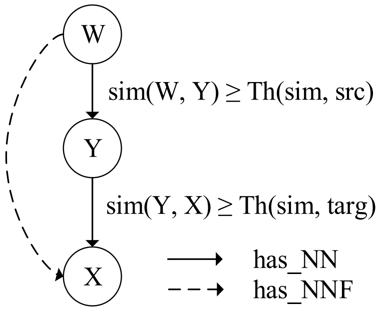
For the application of this algorithm, the following parameters need to be determined:

1 Are there any conditions –other than the existence of a common near neighbour– that must hold, in order for two users *X* and *W* to be considered as NNFs? In this paper, we consider the following aspects:

   - A similarity threshold *Th*(*sim*, *src*) between users *W* and *Y*. Under this condition, user *W* having user *Y* as a NN is considered a NNF with a user *X* ∈ *NN*(*Y*) only if *sim*(*W*, *Y*) ≥ *Th*(*sim*, *src*). The element 'src' in the name of the threshold corresponds to the fact that user *W* is the source of the FOAF relationship.
   - A similarity threshold *Th*(*sim*, *targ*) between users *X* and *Y*. Under this condition, a user *W* being an NN of user *Y*, is considered a NNF with a user *X* ∈ *NN*(*Y*) only if *sim*(*X*, *Y*) ≥ *Th*(*sim*, *targ*). The element 'targ' in the name of the threshold corresponds to the fact that user *W* is the target of the FOAF relationship.

Figure 1 illustrates how the source and target similarity thresholds are applied in the context of FOAF relationship establishment (i.e., relationships of type NNF).

Figure 1 Source and target similarity thresholds in the context of FOAF relationship establishment



- A similarity threshold *Th*(*sim*, *endpoints*) for the similarity between users *W* and *X*. Since users *W* and *X* have no ratings in common, their similarity is computed indirectly, as the product of *sim*(*W*, *Y*) and *sim*(*X*, *Y*). Under this condition, user *W* having user *Y* as a NN is considered a NNF with a user $X \in NN(Y)$ only if $sim(X, Y) * sim(X, W) \geq Th(sim, endpoints)$.
- A number of common ratings threshold *Th*(*cr*, *src*), which corresponds to the minimum number of items that *W* and *Y* have rated in common. Under this condition, user *W* having user *Y* as a NN is considered a NNF with a user $X \in NN(Y)$ only if $|ratings(W) \cap ratings(Y)| \geq Th(cr, src)$.
- A number of common ratings threshold *Th*(*cr*, *targ)* which corresponds to the minimum number of items that *X* and *Y* have rated in common. Under this condition, user *W* having user *Y* as a NN is considered a NNF with a user $X \in NN(Y)$ only if $|ratings(X) \cap ratings(Y)| \geq Th(cr, targ)$.
- A combined threshold regarding the number of common ratings threshold *Th*(*cr*, *comb*), pertaining to the total number of common ratings that both *W* and *X* have with *Y*. Under this condition, user *W* having user *Y* as a NN is considered a NNF with a user $X \in NN(Y)$ only if $(|ratings(W) \cap ratings(Y)| + |ratings(X) \cap ratings(Y)|) \geq Th(cr, comb)$.

These conditions can be applied either individually or in conjunctive fashion. Note that if $(Th(sim, src) = Th(sim, targ) \wedge Th(cr, src) = Th(cr, targ))$ then the NNF relation is symmetric, i.e., $X \in NNF(W) \Leftrightarrow W \in NNF(Y)$.

2 Which is the optimal method for computing the value of *NNF_sim*? In this paper, we considered the following options:

- $NNF\_sim(W, X) = sim(W, Y) * sim(Y, X)$
- $NNF\_sim(W, X) = \min(sim(W, Y), sim(Y, X))$
- $NNF\_sim(W, X) = \max(sim(W, Y), sim(Y, X))$
- $NNF\_sim(W, X) = sim(W, Y)$
- $NNF\_sim(W, X) = sim(Y, X)$
- $NNF\_sim(W, X) = mean(sim(W, Y), sim(Y, X))$.

Both parameters should be determined in a fashion that:

a maximises coverage

b minimises the prediction error.

In the next section, we investigate these two aspects, in order to identify the optimal settings for the algorithm parameters.

4 Tuning algorithm operation

In this section, we report on the methodology followed to:

a determine the optimal values for parameters $Th_s(W)$, $Th_s(X)$, $Th_c(W)$, $Th_c(X)$ and $Th_c(W, X)$

b determine the optimal computation method for the *NNF_sim* metric.

The methodology consists of a set of experiments, where the relevant aspects varied, in order to gain insight on the effect that each parameter setting has on the coverage and the quality of prediction ratings (as this is quantified through the MAE and RMSE metrics), and also choose the optimal setting for each parameter.

For our experiments we used a machine equipped with six Intel Xeon E7 – 4830 @ 2.13 GHz CPUs, 256 GB of RAM and one 900 GB HDD with a transfer rate of 200 MBps, which hosted the datasets and ran the rating prediction algorithms.

In the following paragraphs, we report on our experiments regarding eight datasets. Seven of these datasets are obtained from Amazon (McAuley et al., 2015a, 2015b) and one from MovieLens (MovieLens, 2018; Harper and Konstan, 2015); the Amazon datasets are relatively sparse, while the MovieLens datasets are relatively dense. Our research targets mainly on the former type (i.e., sparse datasets), however, we include the (more) dense dataset, in order to gain insight on the behaviour of the proposed algorithm in the context of datasets having higher density and verify that it does not deteriorate the coverage and accuracy metrics. The eight datasets used in our experiments are summarised in Table 1 and have the following characteristics:

- They are up to date (published between 1996 and 2016) and are widely used as benchmarking datasets in CF research.
- They vary with respect to the type of dataset item domain (video games, movies, music, books, office products and grocery and gourmet food) and size (from 1.4MB to 227MB in text format).

Table 1 Datasets summary

Dataset name	*#users*	*#ratings*	*#items*	*Avg. #ratings/user*	*Density*	*DB size (in text format)*
Amazon 'Videogames' (McAuley et al., 2015a)	8.1K	157K	50K	19.6	0.0039%	3.8 MB
Amazon 'CDs and Vinyl' (McAuley et al., 2015a)	41.2K	1.3M	486K	31.5	0.0065%	32 MB
Amazon 'Movies and TV' (McAuley et al., 2015a)	46.4K	1.3M	134K	29.0	0.0209%	31 MB
Amazon 'Books' (McAuley et al., 2015a)	295K	8.7M	2.33M	29.4	0.0001%	227 MB
Amazon 'Digital Music' (McAuley et al., 2015a)	6.2K	86K	35K	13.9	0.0040%	1.9 MB
Amazon 'Office Supplies' (McAuley et al., 2015a)	3.7K	66K	25K	17.8	0.0714%	1.4 MB
Amazon 'Grocery and Gourmet Food' (McAuley et al., 2015a)	9K	184K	65K	20.4	0.0314%	4.2 MB
MovieLens 'Latest 100K – Recommended for education and development' (MovieLens, 2018)	700	100K	9K	143	1.5873%	2.19 MB

In each dataset, users initially having less than 10 ratings were dropped, since users with few ratings are known to exhibit low accuracy in predictions computed for them (Ekstrand et al., 2011). This procedure did not affect the MovieLens dataset, because it contains only users that have rated 20 items or more. In the following paragraphs, we report on our findings regarding the performance of the CFfoaf algorithm proposed in this work, versus the *negNNs* algorithm presented in (Margaris and Vassilakis, 2018a).

To compute the MAE, the RMSE and the algorithm's coverage, we employed a 10-fold evaluation. We opted to use 10 folds instead of less ones, in order to maintain the number of ratings per user as high as possible, to avoid inaccuracies due to limited number of (non-hidden) ratings for users. To further validate our results, we also performed an evaluation employing the 'hide one' technique (Yu et al., 2004): each time we hid one rating in the database and then predicted its value based on the ratings of other non-hidden items; this procedure was repeated for all ratings in the database. While the absolute magnitudes of the MAE and RMSE of the two experiments expectedly differed, since the 'hide-one' technique is known to overestimate accuracy (Rao et al., 2008), the relative accuracy improvements observed in both experiments (i.e., the ratio $\frac{\textit{accuracy improvement}}{\textit{initial accuracy}}$) were in close agreement, with the deviations observed being always less than 1.2%. Taking this into account, we present only the findings of the 10-fold evaluation.

4.1 Determining the CF_{foaf} threshold parameters

The first experiment is aimed at determining the criteria which a user *X* must fulfill in order to be considered as a NNF of user *W*, besides the fundamental property that a user *Y* must exist for which $Y \in NN(W) \wedge X \in NN(Y)$. Recall from Section 3 that the relevant conditions explored in this paper are:

- *sim*(*W*, *Y*) ≥ *Th*(*sim*, *src*): the similarity *sim*(*W*, *Y*) between *X* and his immediate near neighbour *Y* must surpass the *Th*(*sim*, *src*) threshold.
- *sim*(*X*, *Y*) ≥ *Th*(*sim*, *targ*): the similarity *sim*(*X*, *Y*) between *W*'s immediate near neighbour and the target of the FOAF relation must meet or exceed the *Th*(*sim*, *targ*) threshold.
- *sim*(*W*, *Y*) ∗ *sim*(*Y*, *X*) ≥ *Th*(*sim*, *endpoints*): the similarity between the endpoints of the FOAF relation, i.e., users *W* and *X*, indirectly computed as the product of the similarities *sim*(*X*, *Y*) and *sim*(*Y*, *W*), must meet or exceed the *Th*(*sim*, *endpoints*) threshold.
- *common_ratings*(*W*, *Y*) ≥ *Th*(*cr*, *src*)*:* the number of items that users *W* and *Y* have rated in common must meet or exceed the *Th*(*cr*, *src*) threshold.
- *common_ratings*(*X*, *Y*) ≥ *Th*(*cr*, *targ*)*:* the number of items that users *X* and *Y* have rated in common must meet or exceed the *Th*(*cr*, *targ*) threshold.
- *common_ratings*(*W*, *Y*) + *common_ratings*(*X*, *Y*) ≥ *Th*(*cr*, *comb*)*:* the number of items rated in common by users *W* and *Y* plus the number items rated in common by users *X* and *Y* must meet or exceed the *Th*(*cr*, *comb*) threshold.

Obviously, it holds that *Th*(*cr*, *src)* ≥ 1 since if *W* did not have any common ratings with *Y*, then *Y* could not be a NN for *W*; and similarly, *Th*(*cr*, *targ*) ≥ 1 and *Th*(*cr*, *comb*) ≥ 2. Regarding the values of *Th*(*sim*, *src*) and *Th*(*sim*, *targ*) we consider only values that are greater than zero, under the rationale that it is only meaningful to include NNFs that are positively correlated under the employed similarity metric.

In order to find the optimal setting for the above listed conditions, in our first experiment, we examined different combination of values for the *Th*(*sim*, *src*), *Th*(*sim*, *targ*), *Th*(*sim*, *endpoints*), *Th*(*cr*, *src*), *Th*(*cr*, *targ*) and *Th*(*cr*, *comb*). Overall, more than 40 value combinations were examined, however, for conciseness purposes, we will report only the most indicative ones. For each of them, we present the coverage increase and the improvement of rating prediction accuracy achieved (measured in terms of the MAE and RMSE metrics reduction, as described above). These findings are depicted in Figure 2.

Table 2 Statistical figures for criteria combination performance

Criteria combination	*Coverage increase*			*MAE reduction*			*RMSE reduction*		
	Rank	*Average*	*StdDev*	*Rank*	*Average*	*StdDev*	*Rank*	*Average*	*StdDev*
all = min	1	42.31%	0.1826	1	1.77%	0.0094	1	3.35%	0.0161
Th(sim, src) = 0.5	2	42.20%	0.1824	2	1.76%	0.0093	2	3.31%	0.0159
Th(sim, targ) = 0.5	3	42.10%	0.1819	3	1.68%	0.0089	3	3.28%	0.0152
Th(sim, src) = 0.5 && Th(sim, targ) = 0.5	4	42.08%	0.1816	4	1.65%	0.0086	4	3.25%	0.0155
Th(cr, src) = 2	6	35.06%	0.1514	5	1.57%	0.0084	5	3.02%	0.0144
Th(cr, targ) = 2	5	39.14%	0.169	6	1.19%	0.0063	6	2.31%	0.0109
Th(cr, sim) = 2 && Th(cr, targ) = 2	8	29.96%	0.1294	7	1.03%	0.0055	7	1.85%	0.0094
Th(cr, comb) = 5	7	34.04%	0.147	8	0.74%	0.004	8	1.50%	0.0068

Furthermore, we ran the same experiment for all the datasets listed in Table 1; the results were consistent across all datasets, in the sense that the ranking of criteria combinations was the same for all datasets, hence in this section we only present the mean values of the respective metrics for all datasets. Detailed figures for each dataset and relevant discussions are presented in Section 5. Table 2 lists statistical properties on the behaviour of each of the criteria combinations depicted in Figure 2 across all datasets listed in Table 1 and under the PCC similarity metric. The behaviour under the CS similarity metric follows the same pattern. More specifically, for each performance metric (coverage increase, MAE reduction and RMSE reduction) and for each criteria combination, Table 2 includes the following sub-columns:

- *rank*, which indicates the relative ranking of the specific criteria combination among all criteria combinations. The criteria combination achieving the biggest performance metric improvement is ranked first, and the one attaining the lowest improvement is ranked last. Note that the rank is the same across all datasets, e.g., the criteria combination ‘all = min’ was ranked first regarding coverage increase for each of the datasets in Table 1.
- *average*, indicating the average performance metric improvement achieved while using the specific criteria combination across all datasets.
- *stdDev*, depicting the standard deviation of the performance metric improvement achieved while using the specific criteria combination across all datasets.

Regarding the computation of the *NNF_sim*(*W*, *X*) metric, at this phase it was calculated using the formula

$$NNF\_sim(W, X) = sim(W, Y) * sim(Y, X) \tag{5}$$

However, we also validated that the results remain consistent with the other *NNF_sim* computation formulas listed in Section 3, by performing the experiment at random samples of the search space; in all cases, the results were in close agreement with those obtained when using the formula in equation (5), in the sense that the ranking of the methods’ performance was the same under all NNF computation methods. Hence, we will confine ourselves in presenting and discussing only the results obtained when using the formula in equation (5), which additionally was proven to be optimal by our second experiment.

Figure 2 Coverage increase and prediction error reduction under different NNF conditions

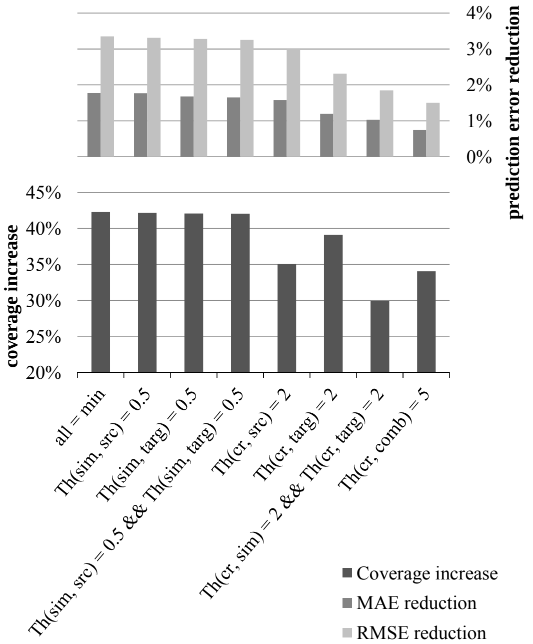


Figure 2 depicts the coverage increase (bottom half of the chart) and the rating prediction error reduction (upper half of the chart) under different NNF inclusion criteria, when considering the PCC similarity metric. The results for the SC similarity metric follow the same pattern.

We can observe that using the minimum values for all thresholds (setting labelled ‘all = min’) leads to the maximum increase in coverage (42.31%) and the greatest reduction in the MAE (1.77%) and RMSE (3.35%). The fact that the RMSE improvement is higher than the corresponding MAE reduction indicates that the proposed algorithm corrects prediction with high errors, rather than predictions with small deviations. The setting where *Th(sim, src)* is set to 0.5 and all other thresholds are set to the minimum value follows closely, with a coverage increase equal to 42.20%, while the MAE and RMSE improvements are equal to 1.76% and 3.31%, respectively. Two more settings lag behind by small margins, namely:

a the setting where *Th(sim, targ)* is set to 0.5 and all other thresholds are set to their minimum values (coverage increase: 42.10%; MAE decrement: 1.68%; RMSE improvement: 3.28%)

b the setting where both similarity thresholds are set to 0.5 and all other thresholds are set to their minimum values (coverage increase: 42.08%; MAE decrement: 1.65%; RMSE improvement: 3.25%).

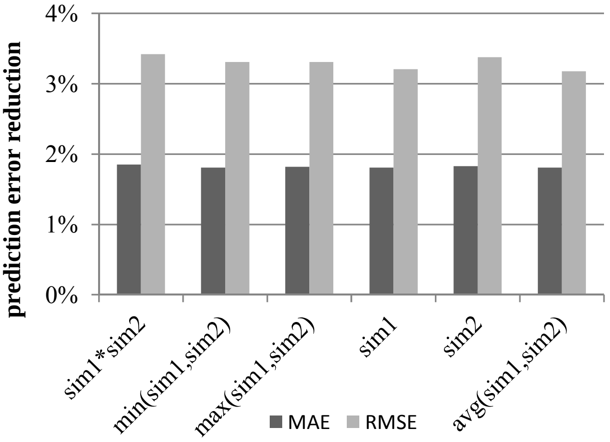
When the values of thresholds related to the minimum number of commonly rated items are increased, coverage improvement declines; this effect is sharper when the increment pertains to the *Th*(*cr*, *src*).

Considering the above results, in the rest of this paper we adopt the setting where all thresholds are set to their minimum values.

4.2 Determining the CF$_{foaf}$ transitive user similarity computation method

Our second experiment, targets to the method for calculating the *NNF_sim* value in the recommendation algorithm. Figure 3 depicts the MAE reduction achieved under different *NNF_sim* computation methods.

Figure 3 Prediction error reduction under different NNF similarity (NNF_sim) computation methods



We can observe that the calculation of *NNF_sim* as the product of the two individual similarities is the optimal one, achieving a MAE reduction of 1.77% on average across all datasets, while the respective RMSE improvement is equal to 3.35%; this behaviour is consistent for each individual dataset, i.e., the multiplication method is ranked first, regarding the MAE and the RMSE reduction achieved, for each of the considered datasets. In Figure 3, we can also observe that the variations in the reduction of the MAE are very small (1.15%–2.25%), while the variations in the reduction of the RMSE metric are higher (1.15%–7.15%).

5 Performance evaluation

After having determined the optimal parameters for the operation of the CF$_{foaf}$ algorithm (values for the different thresholds and method for computing the *NNF_sim* metric), we proceed in evaluating the algorithm's performance in terms of coverage and prediction accuracy. More specifically, we measure:

a coverage, i.e., the percentage of the cases for which a personalised prediction can be computed (Burke, 2002)

b prediction accuracy, i.e., the closeness of the algorithms' predictions to the actual values that are known to have been entered by the users.

For quantifying prediction accuracy, we employed two well-established error metrics, namely the mean absolute error (MAE), and the root mean squared error (RMSE) that 'punishes' big mistakes more severely. For both aspects, the performance of the plain CF algorithm is used as a baseline.

Besides obtaining absolute metrics regarding improvements in coverage and accuracy achieved by the CF$_{foaf}$ algorithm, we compare the performance of CF$_{foaf}$ with the performance of other, state-of-the-art algorithms targeting the increase of coverage. In particular we compare the proposed algorithm to the *negative NNs algorithm* (*negNNs*) introduced in Margaris and Vassilakis (2018a), which is the most recently published one, does not necessitate any additional information (e.g., user relationships sourced from social networks), and achieves considerable improvements in coverage while maintaining (and slightly improving) the quality of rating predictions. Note that the comparison with the negative NNs algorithm is only performed for the case that the Pearson correlation coefficient is used as a similarity metric, because the publication introducing the negative NNs algorithm (Margaris and Vassilakis, 2018a) does not provide information on how the algorithm could be adapted for use with the cosine similarity metric.

In the following, we initially report on the results obtained from our experiments on the seven sparse datasets listed in Table 1, since the proposed algorithm targets this dataset category. The results obtained from the dense dataset (MovieLens 'Latest 100K') are discussed separately, so as to gain insight on the effect of the algorithm mainly on the prediction accuracy, since for dense datasets coverage is already at high levels.

For conducting these experiments, we used the machine described in Section 4.

5.1 Experiments using the Pearson correlation coefficient as a similarity metric

Figure 4 depicts the performance metrics regarding the increase in coverage when similarity between users is measured using the Pearson correlation coefficient.

We can observe that the proposed algorithm achieves an average coverage increase over all datasets equal to 48.08%, surpassing the corresponding improvement achieved by the negative NNs algorithm (11.87%) by four times. At individual dataset level, the performance edge of the CF_{foaf} algorithm against the negative NNs one ranges from 2.78 times higher for the ‘Amazon Office’ dataset to 7.22 times higher, observed for the ‘Amazon Grocery and Gourmet Food’ dataset. The lowest increase is observed for the ‘Amazon Movies and TV’ dataset; interestingly this dataset has the highest (#ratings/#items) ratio among the seven sparse datasets; while the results in other datasets do not concur that there is a direct relationship between the coverage increase and the (#ratings/#items) ratio, this is a noteworthy element and will be further investigated in the context of our future work.

Figure 4 Coverage increase for the different datasets, when using the PCC as a similarity metric

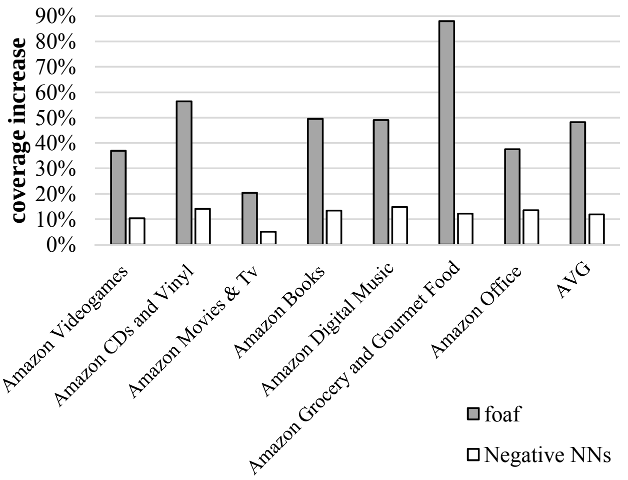


Figure 5 MAE reduction for the different datasets, when using the PCC as a similarity metric

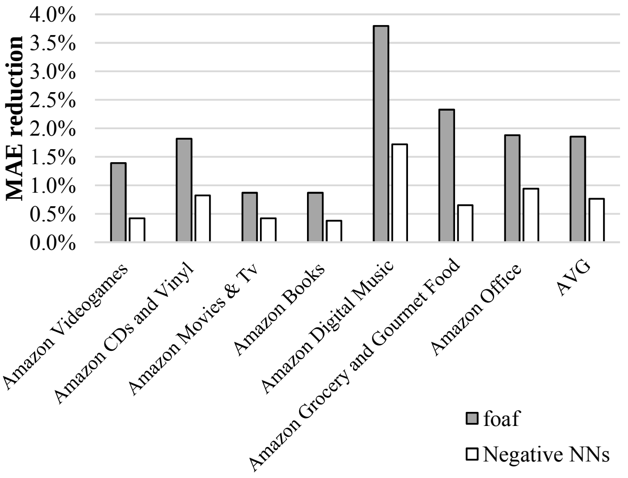
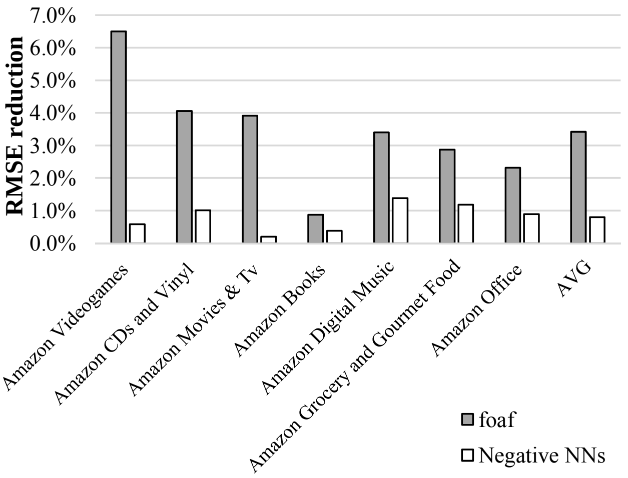


Figure 5 illustrates the performance metrics regarding the MAE reduction when similarity between users is measured using the Pearson correlation coefficient.

We can observe that the proposed algorithm achieves an average MAE reduction over all datasets equal to 1.83%, surpassing by approximately 2.5 times the corresponding improvement achieved by the negative NNs algorithm (0.77%). At individual dataset level, the performance edge of the CF_{foaf} algorithm against the negative NNs one ranges from 2.0 times higher for the ‘Amazon Office’ dataset to 3.58 times higher, observed for the ‘Amazon Grocery and Gourmet Food’ dataset. The lowest MAE improvement for the CF_{foaf} algorithm is observed for the ‘Amazon Movies and TV’ and the ‘Amazon Books’ datasets (0.87%), which have the highest (#ratings/#items) ratio among the seven sparse datasets (9.7 and 3.73, respectively). In this case, the results for the other datasets are inline with this observation, i.e., when the (#ratings/#items) ratio increases, the achieved MAE reduction drops. Further investigation of this aspect is again part of our future work.

Figure 6 demonstrates the performance metrics regarding the RMSE reduction when similarity between users is measured using the Pearson correlation coefficient.

Figure 6 RMSE reduction for the different datasets, when using the PCC as a similarity metric

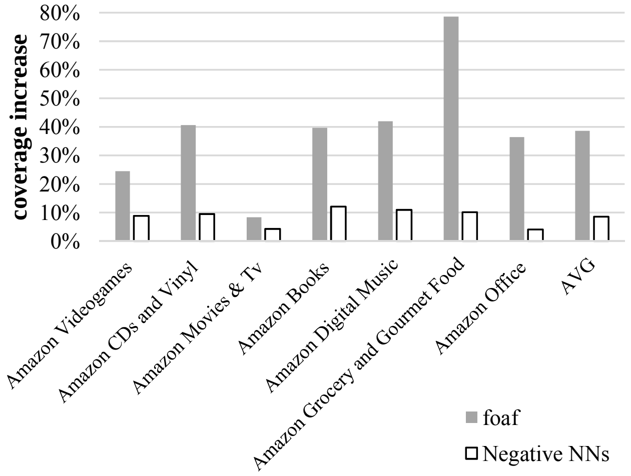


We can observe that the proposed algorithm achieves an average RMSE reduction over all datasets equal to 3.46%, surpassing by approximately 4.25 times the corresponding improvement achieved by the negative NNs algorithm (0.78%). At individual dataset level, the performance edge of the CF_{foaf} algorithm against the negative NNs one ranges from 2.29 times higher for the ‘Amazon Books’ dataset to 19.55 times higher, observed for the ‘Amazon Movies and TV dataset’. The lowest RMSE improvement for the CF_{foaf} algorithm is observed for the ‘Amazon Books’ datasets (0.86%), while the highest improvement is achieved for the ‘Amazon Videogames dataset’ (6.53%). The fact that the RMSE metric improvement is higher than the corresponding MAE improvement indicates that the CF_{foaf} algorithm achieves to correct some prediction errors with high absolute magnitudes, since the RMSE metric is known to penalise high errors, while the MAE metric takes into account all error magnitudes with an equal weight.

5.2 Experiments using the cosine similarity metric

Figure 7 illustrates the performance metrics regarding the increase in coverage when similarity is measured using the cosine similarity metric.

Figure 7 Coverage increase for the different datasets, when using the CS as a similarity metric



We can observe that the proposed algorithm achieves an average coverage increase over all datasets equal to 38.57%, ranging from 8.28% (observed for the 'Amazon Movies and TV' dataset) to 78.69% (observed for the 'Amazon Grocery and Gourmet Food'. These performance metrics surpass the corresponding improvements achieved by the negative NNs algorithm (8.47%) by four times. We can notice that the average coverage improvement achieved when the Cosine Similarity metric is employed lags behind the corresponding improvement obtained when using the Pearson correlation coefficient by approximately 10%; this can be attributed to the fact that the plain CF cosine-based similarity algorithm achieves higher coverage than the corresponding Pearson correlation coefficient-based ones (Margaris and Vassilakis, 2018b), therefore the cosine-based CF_{foaf} algorithm has smaller improvement margins.

Figure 8 MAE reduction for the different datasets, when using the CS as a similarity metric

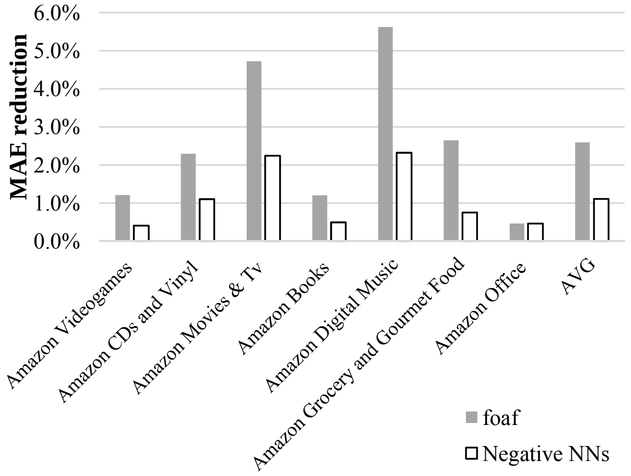
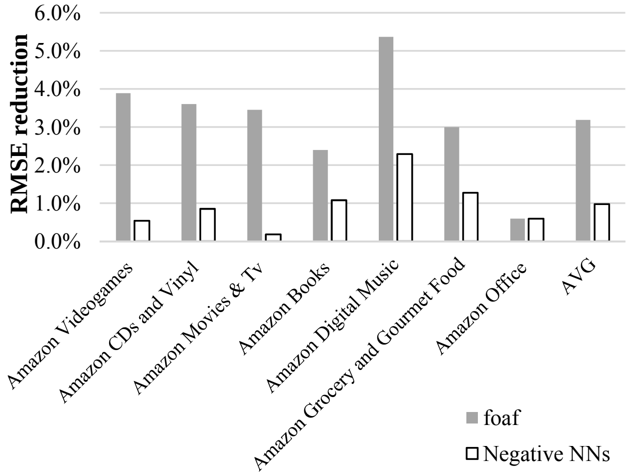


Figure 8 depicts the performance metrics regarding the MAE reduction when similarity is measured using the cosine similarity metric.

We can observe that the proposed algorithm achieves an average MAE reduction over all datasets equal to 2.62%, ranging from 0.47% (for the 'Amazon Office' dataset) to 5.64% (for the 'Amazon Digital Music'). On average, the MAE improvement achieved by the CF_{foaf} algorithm surpasses that of the negative NNs algorithm (1.05%) by 2.5 times. The average improvement is 1.40 times higher than the corresponding improvement obtained when the Pearson correlation coefficient is used.

Figure 9 RMSE reduction for the different datasets, when using the CS as a similarity metric



Finally, Figure 9 pictures the performance metrics regarding the RMSE reduction when similarity between users is measured using the Cosine Similarity.

We can observe that the proposed algorithm achieves an average RMSE reduction over all datasets equal to 3.17%, ranging from 0.58% (for the 'Amazon Office' dataset) to 5.35% (for the 'Amazon Digital Music'). On average, the RMSE improvement achieved by the CF_{foaf} algorithm surpasses that of the negative NNs algorithm (0.95%) by 3.2 times. Again, the RMSE reduction is higher than the MAE reduction, indicating that the algorithm corrects some errors with high absolute magnitudes. In this case, the average RMSE improvement obtained when using the two similarity metrics (PCC and CS) is almost equal (PCC: 3.40%, CS: 3.21%).

5.3 The MovieLens 'Latest 100K – recommended for education and development' dataset

In this subsection, we discuss the results obtained from applying the CF_{foaf} algorithm on the MovieLens 'Latest 100K' dataset. The density of this dataset is 1.59%, which is considerably higher than the density of the other datasets (from 22 to 16.000 times higher), and the coverage achieved by the plain CF algorithm is 97.31%, as far as the PCC is concerned, hence the coverage improvement opportunity is severely limited. Even in this case, the negNNs algorithm increases the coverage by 0.45%, while the $\mathrm{CF}_{\mathrm{foaf}}$ algorithm, presented in this paper achieves an increment of 1.9%.

Considering rating prediction quality, when using the negNNs algorithm the MAE drops by 0.78%, while the RMSE drops by 1.86%; the proposed $\mathrm{CF}_{\mathrm{foaf}}$ algorithm surpasses all these performance measures, achieving drops by 1.38% and 2.57%, respectively.

Regarding the case where the CS metric is used, the coverage of the plain CF algorithm is 95.68%. In this context, both the $\mathrm{CF}_{\mathrm{foaf}}$ and the negNNs algorithms increase coverage by 1.2%. The $\mathrm{CF}_{\mathrm{foaf}}$ algorithm achieves to reduce the MAE and the RMSE by 0.58% and 0.85%, respectively, having thus a performance edge over the negNNs algorithm, which leaves MAE intact (no reduction or increment), while reducing the RMSE by 0.28%.

We can observe that even in dense datasets, the proposed algorithm offers a small coverage increase and additionally achieves a considerable improvement in rating prediction quality, surpassing the negNNs algorithm proposed in Margaris and Vassilakis (2018a).

Considering the matrix factorisation-based algorithm proposed by Guan et al. (2017) achieves on the same dataset an RMSE reduction ranging from 1.43% (basic ESVD approach) to 1.52 (four-layer ESVD approach). Additionally, two proposed extensions of ESVD proposed in the same paper, namely item-wise ESVD (IESVD) and user-wise ESVD (UESVD) are applied on subsets of the Movielens 1M datasets, achieving RMSE reductions of 1.78% and 0.97%, respectively. We can observe that the improvements achieved by the proposed algorithm under the PCC similarity metric surpass those achieved by the algorithm proposed by Guan et al. (2017), while under the CS similarity metric the situation is reversed. The comparison between the proposed algorithm and the one proposed by Guan et al. (2017) is limited to their performance on the MovieLens 'Latest 100K – Recommended for education and development' dataset because the work in Guan et al. (2017) presents results only on the MovieLens and Netflix datasets; notably, the improvements achieved by the algorithm proposed in Guan et al. (2017) on the MovieLens dataset are superior to those achieved by the same algorithm on the Netflix dataset.

6 Conclusions and future work

In this paper we have introduced a novel CF algorithm for improving prediction coverage in sparse datasets. The proposed algorithm has been experimentally verified using eight datasets and compared with the *negNNs* algorithm (Margaris and Vassilakis, 2018a), as far as prediction accuracy and coverage are concerned. The $\mathrm{CF}_{\mathrm{foaf}}$ algorithm, presented in this paper, has been found to consistently outperform the *negNNs* algorithm, which is a state-of-the-art algorithm targeting increase of coverage, in all tested datasets. The evaluation results have shown that for sparse datasets the proposed algorithm provides a substantial increase in coverage, ranging from 20.46% to 88.04%, with an average of 48%, as far as the Pearson Correlation Coefficient is concerned, and from 8.32% to 78.73%, with an average of 38.57%, as far as the Cosine Similarity metric is concerned.

At the same time, the proposed algorithm offers considerable improvements regarding rating prediction quality; the MAE decreases by 1.83% and the RMSE by 3.46% on average, as far as the Pearson Correlation Coefficient is concerned, and by 2.62% and 3.17%, respectively, as far as the cosine similarity metric is concerned.

Furthermore, in the context of dense datasets, the proposed algorithm has been found to offer small to negligible improvements in coverage and recommendation quality. This indicates that the algorithm can be employed across all datasets, regardless of their density, hence a CF system implementation may employ the proposed algorithm, even without examining the properties of the used dataset, since using this algorithm will either improve the performance of the CF system or – in the worst case – have small positive effects on it.

Our future work will focus on exploring alternative techniques for increasing coverage and/or reducing recommendation error in sparse CF datasets. Furthermore, we are planning to examine these techniques in more correlation metrics, such as the Euclidian distance, the Manhattan distance and the Spearman coefficient (Herlocker et al., 2004); a deeper analysis of the dataset factors affecting the algorithm performance will be explored. Adaptation of the proposed approaches for use with matrix factorisation techniques (Koren et al., 2009) is also considered. Finally, the combination of the proposed method with other techniques, such as exploiting social network data for improving the quality of recommendations (Margaris et al., 2017) will be investigated.

References

Bakshy, E., Rosenn, I., Marlow, C. and Adamic, L. (2012) 'The role of social networks in information diffusion', *Proceedings of the 21st International Conference on World Wide Web*, pp.519–528, ACM.

Balabanovic, M. and Shoham, Y. (1997) 'Fab: content-based, collaborative recommendation', *Communications of the ACM*, Vol. 40, No. 3, pp.66–72.

Brickley, D. (2005) *The Friend of a Friend (FOAF) Project* [online] http://www.foafproject.org/, (accessed May 2019).

Burke, R. (2002) 'Hybrid recommender systems: survey and experiments', *User Modeling and User-Adapted Interaction*, Vol. 12, No. 4, pp.331–370, Kluwer Academic Publishers Hingham, MA, USA.

De la Haye, K., Robins, G., Mohr, P. and Wilson, C. (2011) 'Homophily and contagion as explanations for weight similarities amongst adolescent friends', *Journal of Adolescent Health*, Vol. 49, No. 4, pp.421–427, Elsevier.

Dias, R. and Fonseca, M.J. (2013) 'Improving music recommendation in session-based collaborative filtering by using temporal context', *Proceedings of the 25th IEEE International Conference on Tools with Artificial Intelligence*, pp.783–788, IEEE.

Ekstrand, M.D., Riedl, J.T. and Konstan, J.A. (2011) 'Collaborative filtering recommender systems', *Foundations and Trends in Human-Computer Interaction*, Vol. 4, No. 2, pp.81–173, IEEE.

Gong, S. (2010) 'A collaborative filtering recommendation algorithm based on user clustering and item clustering', *Journal of Software*, Vol. 5, No. 7, pp.745–752.

Guan, X., Li, C. and Guan, Y. (2017) 'Matrix factorization with rating completion: an enhanced SVD model for collaborative filtering recommender systems', *Access*, Vol. 5, pp.27668–27678, IEEE.

Harper, F.M. and Konstan, J.A. (2015) 'The MovieLens Datasets: history and context', *ACM Transactions on Interactive Intelligent Systems*, Vol. 5, No. 4, pp, 19:1–19:19, Article 19 [online] https://dl.acm.org/citation.cfm?id=2827872.

Herlocker, J.L., Konstan, J.A., Terveen, L.G. and Riedl, J.T. (2004) 'Evaluating collaborative filtering recommender systems', *ACM Transactions of Information Systems*, Vol. 22, No. 1, pp.5–53, ACM.

Hoff, P. (2008) 'Modeling homophily and stochastic equivalence in symmetric relational data', *Advances in Neural Information Processing Systems 20*, pp.657–664, MIT Press.

Jøsang, A. and Pope, S. (2005) 'Semantic constraints for trust transitivity', Proceedings *of the 2nd Asia-Pacific conference on Conceptual Modelling*, Vol. 43, pp.59–68.

Koren, Y., Bell, R. and Volinsky, C. (2009) 'Matrix factorization techniques for recommender systems', *IEEE Computer*, Vol. 42, No. 8, pp.42–49, IEEE.

Lazarsfeld, P.F. and Merton, R.K. (1954) 'Friendship as a social process: a substantive and methodological analysis', *Freedom and Control in Modern Society*, pp. 8–66, Van Nostrand, New York.

Margaris, D. and Vassilakis, C. (2016) 'Pruning and aging for user histories in collaborative filtering', *Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence*, IEEE, pp.1–8.

Margaris, D. and Vassilakis, C. (2017) 'Enhancing user rating database consistency through pruning', *Transactions on Large-Scale Data and Knowledge-Centered Systems*, Vol. 34, pp.33–64, Springer, Berlin, Heidelberg.

Margaris, D. and Vassilakis, C. (2018a) 'Improving collaborative filtering's rating prediction coverage in sparse datasets by exploiting user dissimilarity', *Proceedings of the 4th IEEE International Conference on Big Data Intelligence and Computing*, pp.1054–1059, IEEE.

Margaris, D. and Vassilakis, C. (2018b) 'Enhancing rating prediction quality through improving the accuracy of detection of shifts in rating practices', *Transactions on Large-Scale Data- and Knowledge-Centered Systems*, Vol. 37, pp.151–191, Springer, Berlin, Heidelberg.

Margaris, D., Georgiadis, P. and Vassilakis, C. (2015) 'A collaborative filtering algorithm with clustering for personalized web service selection in business processes', *Proceedings of the 9th International Conference on Research Challenges in Information Science*, pp.169–180, IEEE.

Margaris, D., Vassilakis, C. and Georgiadis, P. (2016) 'Recommendation information diffusion in social networks considering user influence and semantics', *Springer Social Network Analysis and Mining*, Vol. 6, No. 1, pp.1–22, Springer, Vienna.

Margaris, D., Vassilakis, C. and Georgiadis, P. (2017) 'Knowledge-based leisure time recommendations in social networks', *Current Trends on Knowledge-Based Systems: Theory and Applications*, pp.23–48, Springer, Cham.

Margaris, D., Vassilakis, C. and Georgiadis, P. (2018) 'Query personalization using social network information and collaborative filtering techniques', Future Generation of Computer Systems, Vol. 78, No. 1, pp. 440-450, Elsevier.

McAuley, J.J., Pandey, R. and Leskovec, J. (2015a) 'Inferring networks of substitutable and complementary products', *Proceedings of the 21st ACM SIGKDD Conference*, pp.785–794, ACM.

McAuley, J.J., Targett, C. and Hengel, J. (2015b) 'Image-based recommendations on styles and substitutes', *Proceedings of the 38th International ACM SIGIR Conference*, pp.43–52, ACM.

McPherson, J.M. and Smith-Lovin, L. (1987) 'Homophily in voluntary organizations: status distance and the composition of face-to-face groups', *American Sociological Review*, Vol. 52, pp.370–379.

Moshfeghi, Y., Piwowarski, B. and Jose, J.M. (2011) 'Handling data sparsity in collaborative filtering using emotion and semantic based features', *Proceedings of 34th International ACM SIGIR Conference*, pp.625–634, ACM.

MovieLens (2018) MovieLens [online] http://grouplens.org/datasets/movielens/ (accessed January 2019).

Pham, M.C., Cao, Y., Klamma, R. and Jarke, M. (2011) 'A clustering approach for collaborative filtering recommendation using social network analysis', *Journal of Universal Computer Science*, Vol. 17, No. 4, pp.583–604.

Poirier, D., Fessant, F. and Tellier, I. (2010) 'Reducing the cold-start problem in content recommendation through opinion classification', *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pp.204–207.

Rao, R.B., Fung, G. and Rosales, R. (2008) 'On the dangers of cross-validation. An experimental evaluation', *Proceedings of the 2008 SIAM International Conference on Data Mining*, pp.588–596, ASA.

Vozalis, M.G., Markos, A.I. and Margaritis, K.G. (2009) 'A hybrid approach for improving prediction coverage of collaborative filtering', *Artificial Intelligence Applications and Innovations*, Vol. 296, pp.491–498, Springer, Boston, MA.

Wen, H., Ding, G., Liu, C. and Wang, J. (2014) 'Matrix factorization meets cosine similarity: addressing sparsity problem in collaborative filtering recommender system', *Proceedings of APWeb 2014, LNCS*, vol. 8709, pp.306–317, Springer, Cham.

Yu, K., Schwaighofer, A., Tresp, V., Xu, X. and Kriegel, H.P. (2004) 'Probabilistic memory-based collaborative filtering', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, No. 1, pp.56–69.