Recommending Collaborative Filtering algorithms using subsampling landmarkers

Tiago Cunha[1], Carlos Soares[1], and André C.P.L.F. de Carvalho[2]

[1] INESC-TEC/FEUP, Porto, Portugal
{tiagodscunha,csoares}@fe.up.pt
[2] ICMC - USP São Carlos, São Paulo, Brazil
andre@icmc.usp.br

Abstract. Recommender Systems have become increasingly popular, propelling the emergence of several algorithms. As the number of algorithms grows, the selection of the most suitable algorithm for a new task becomes more complex. The development of new Recommender Systems would benefit from tools to support the selection of the most suitable algorithm. Metalearning has been used for similar purposes in other tasks, such as classification and regression. It learns predictive models to map characteristics of a dataset with the predictive performance obtained by a set of algorithms. For such, different types of characteristics have been proposed: statistical and/or information-theoretical, model-based and landmarkers. Recent studies argue that landmarkers are successful in selecting algorithms for different tasks. We propose a set of landmarkers for a Metalearning approach to the selection of Collaborative Filtering algorithms. The performance is compared with a state of the art systematic metafeatures approach using statistical and/or information-theoretical metafeatures. The results show that the metalevel accuracy performance using landmarkers is not statistically significantly better than the metafeatures obtained with a more traditional approach. Furthermore, the baselevel results obtained with the algorithms recommended using landmarkers are worse than the ones obtained with the other metafeatures. In summary, our results show that, contrary to the results obtained in other tasks, these landmarkers are not necessarily the best metafeatures for algorithm selection in Collaborative Filtering.

Keywords: Metalearning, Subsampling Landmarkers, Collaborative Filtering

1 Introduction

Recommender Systems (RSs) recommend potentially interesting items to users in order to deal with the information overload problem [1]. Collaborative Filtering (CF) is the most popular of the available recommendation strategies. Despite the large amount of research dedicated to this topic, there are still several challenges that need to be addressed. One of them is how to choose the best CF algorithm for a given dataset. Since training and evaluating all algorithms for the new

dataset requires a prohibitive amount of time and resources, automatic solutions based on prior knowledge are of the utmost importance. Metalearning (MtL) is an approach useful for that purpose [7].

MtL is concerned with discovering patterns in data and understanding the effect on the behavior of algorithms [30]. It has been extensively used for algorithm selection [6,27,28]. MtL casts the algorithm selection problem as a learning task. For such, it uses a metadataset, where each meta-example corresponds to a problem. For each meta-example, the predictive features are characteristics (metafeatures) extracted from the corresponding problem and the target represents the performance of algorithms when applied to the problem (metatarget) [5].

Metafeatures are regarded as the most important element in a MtL task [5]. It is essential for them to be representative of the problem at hand. The metafeatures used must contain information that discriminates the performance of different algorithms in such a way that the patterns found are useful for future applications. However, this is not a trivial task. The research in this topic has originated several different types of metafeatures, such as statistical and/or information-theoretical, model-based and landmarkers, which are related to the dataset, model and performance properties, respectively [29, 30].

The algorithm selection task for CF has received considerable attention recently [2,7,10,14,23]. Related work has investigated the effect of different statistical and information-theoretical metafeatures with positive performances. However, none has investigated the merits of landmarkers as metafeatures. Since these metafeatures use simple estimates of performance to predict the actual performance of algorithms, its efficacy in solving the algorithm selection problem is not only expected but has been demonstrated in various other tasks [3, 11, 17, 18, 20, 21, 25]. Therefore, it is important to understand if their effect is similarly positive in selecting CF algorithms.

Hence, the main contribution of this paper is the proposal of several subsampling landmarkers and their experimental validation in terms of their merits to select CF algorithms. To do so, this paper provides an extensive collection of baselevel datasets, algorithms and evaluation measures similarly to the ones found in the state of the art [7]. The subsampling landmarkers are proposed and analyzed as relative landmarkers. Such landmarkers look not only towards the absolute performance estimations, but also to the relative performance between landmarkers. Our motivation lies in ensuring a proper exploration of the landmarkers concept for the CF scope. All different metafeatures are compared to the state of the art approach in statistical and information-theoretical metafeatures [7] in terms of metalevel accuracy and impact on the baselevel performance. The results show that landmarkers are not statistical significantly better than the statistical and/or information-theoretical metafeatures.

This document is organized as follows: Section 2 presents related work on CF, MtL and algorithm selection for CF; Section 3 presents the approach used for subsampling landmarkers and relative landmarkers and explains the experimental setup. In Section 4, several aspects of the proposed approach are evaluated and discussed. Section 5 presents the conclusions and directions for future work.

2 Related Work

2.1 Collaborative Filtering

RSs were proposed to complement Information Retrieval systems, providing an alternative to solve the problem of information overload and recommend potentially interesting items to users [4]. RSs are inspired by human social behavior, where it is common to take into account the tastes, opinions and experiences of acquaintances when making decisions [4]. Several strategies are used in such systems, such as: 1) recommend items that similar users find relevant, 2) recommend items with similar characteristics, 3) recommend items depending on the user's context, 4) recommend items based on social relationships and 5) recommend items using knowledge about the user's behavior. From the several strategies available, Collaborative Filtering (CF) is the most popular.

CF recommendations are based on the premise that a user will probably like the items favored by a similar user. CF employs the feedback from each individual user to recommend items to similar users [33]. The feedback is a numeric value, proportional to the user's appreciation of an item. Most feedback is based on a rating scale, although other variants such as like/dislike actions and clickstream are also suitable. The data structure used in CF is named rating matrix R. It is usually described as $R^{U \times I}$, representing a set of users U, where $u \in \{1, ..., N\}$ and a set of items I, where $i \in \{1, ..., M\}$. Each element of this matrix (R_{ui}) is the feedback provided by user u for item i. Figure 1 presents such matrix.

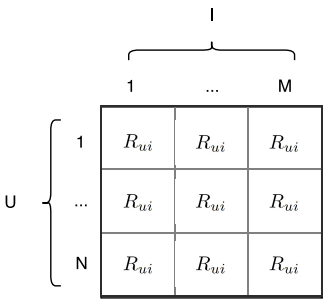

Fig. 1: Rating matrix.

CF algorithms can be organized in two major groups: memory-based and model-based [4]. Memory-based algorithms apply heuristics to a rating matrix to compute recommendations, whereas model-based algorithms induce a model from this matrix. Most memory-based algorithms adopt Nearest Neighbor strategies, while the model-based ones are mostly based on Matrix Factorization [33].

The evaluation of RSs is usually performed by procedures that split the dataset into training and testing subsets (using sampling strategies, such as k-fold cross-validation [16]) and assesses the performance of the trained model on

the testing dataset. Different evaluation metrics exist [22]: for rating accuracy, error measures such as Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE); for classification accuracy, one uses Precision/Recall or Area Under the Curve (AUC); for ranking accuracy, common measures are Normalized Discounted Cumulative Gain (NDCG) and Mean Reciprocal Rank (MRR).

2.2 Metalearning

MtL addresses the algorithm selection problem similarly to a traditional learning process (see Figure 2). First, the problems are characterized by a set of measurable characteristics (i.e., metafeatures) and the compared algorithms are evaluated according to their performance in the learning task. This creates a metadataset, where each meta-example has as predictive attributes the characteristics extracted for the problem and the target attribute is usually the algorithm that obtained the best performance in the specific dataset. Next, a learning algorithm is trained using the metadataset. The trained model represents patterns in the data that relate the metafeatures with the best performing algorithms. Hence, it can be used to predict the best algorithm for a new problem [29].

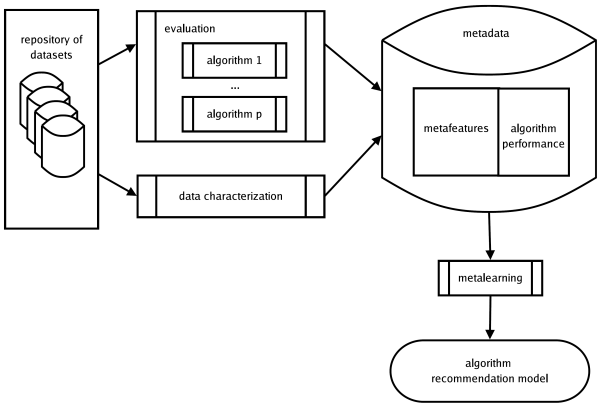

Fig. 2: Metalearning process [5].

As in any other learning problem, the success of a MtL approach depends on the information contained in the independent variables, i.e. the metafeatures. The MtL literature divides metafeatures into three main groups [5, 29, 30]: statistical and/or information-theoretical, model-based and landmarkers.

Statistical and/or information-theoretical metafeatures describe the dataset characteristics using a set of measures from statistics and information theory. These metafeatures assume that there are patterns in the data which can be related to the best algorithms. Examples include simple measures such as the

number of examples and features in the dataset to more advanced measures such as entropy and kurtosis of features and even correlation between features [5].

Model-based characteristics are properties extracted from models induced from the dataset. They refer, for instance, to the number of leaf nodes in a decision tree [5]. The rationale is that there is a relationship between model characteristics and algorithm performance which are dataset-independent. Then, it is expected that these characteristics are able to discriminate among algorithms.

Finally, landmarkers are fast estimates of the algorithm performance on the dataset. There are two different types of landmarkers: those obtained from the application of fast and simple algorithms on complete datasets and those which are achieved by using complete models for samples of datasets, also known as subsampling landmarkers [5]. Such metafeatures rely on the assumption that by estimating the performance of fast and simple models or by using samples of the data, the performance estimates will correlate well with the best algorithms, hence enabling future predictions. In fact, these metafeatures have proven successful on the selection of algorithms for various tasks [3, 11, 17, 18, 20, 21, 25].

2.3 Algorithm selection for CF

Related work in algorithm selection for CF has studied the problem using only statistical and/or information-theoretical metafeatures. These have focused on different aspects of the data distributions [2, 10, 14, 23], the matrix structure [23] and neighborhood statistics [14]. A more recent work has combined the majority of the metafeatures used previously in a single framework [7]. This extensive set of metafeatures (referred to here as Systematic) are used in our experimental study in order to properly compare statistical and/or information-theoretical metafeatures with the set of subsampling landmarkers proposed here.

In order to understand the systematic metafeatures, one must consider first the framework used to generate them. It requires three main elements: object o, function f and a post-function pf. The framework applies the function f to the object o and, afterwards the post-function pf to the outcome of the function f in order to derive the final metafeature. Thus, any metafeature can be represented using the following notation: $\{o.f.pf\}$ [26]. For instance, the metafeature *column.maximum.mean* refers to the mean value of all the maximum values in all columns in the dataset.

Consider now a rating matrix R, with rows (i.e., users) U and columns (i.e., items) I. The objects to be used in the framework are R, U and I. The functions f considered to characterize these objects are: original ratings (ratings), count the number of elements (count), mean value (mean) and sum of values (sum). The post-functions pf are maximum, minimum, mean, standard deviation (sd), median, mode, entropy, Gini index, skewness and kurtosis. Additionally, we consider 4 simple metafeatures: number of users, items, ratings and matrix sparsity. This results in 74 metafeatures which were reduced by correlation feature selection, ending up with the following set: *D.ratings.kurtosis*, *D.ratings.sd*, *I.count.kurtosis*, *I.count.minimum*, *I.mean.entropy*, *I.sum.skewness*, *nusers*, *sparsity*, *U.mean.minimum*, *U.sum.kurtosis*, *U.mean.skewness* and *U.sum.entropy*.

3 Subsampling landmarkers for Collaborative Filtering

This section presents our proposal of subsampling landmarkers for the selection of CF algorithms and the experimental procedure used to validate them. Our motivation for using landmarkers is that, although they have been successfully applied to the algorithm selection problem in other learning tasks [3, 11, 17, 18, 20, 21, 25], they were never adapted for selecting CF algorithms. Since there are no fast/simple CF algorithms, which can be used as traditional landmarkers, we have followed the alternative approach of developing subsampling landmarkers, i.e. applying the complete CF algorithms on samples of the data.

3.1 Subsampling landmarkers

Subsampling landmarkers are based on the estimation of the performance of algorithms on random samples from the original datasets. This means that for each CF dataset, random samples are extracted. Then, CF algorithms are trained on these samples and their performance assessed using different metrics. The outcome is a subsampling landmarker for each pair algorithm/evaluation measure. In order to properly validate the impact of subsampling landmarkers, we recur to different ways to take advantage of these metafeatures, also known as relative landmarkers [11]:

- Absolute: this is the most straightforward approach since it does not operate any transformation on the subsampling landmarkers. It uses the estimated performance values as the metafeature.
- Ranking: this approach is based on the ranking of the landmarkers $L = \{l_1, l_2, ..., l_n\}$. Therefore, the metafeatures are now the rank of the landmarker, where 1 indicates the best landmarker and n the worst.
- Pairwise: this approach performs pairwise comparison for all pairs of landmarkers. Consider two landmarkers l_i and l_j. If the performance of l_i is greater, equal or worse than l_j, then the final metafeature values are 1, 0 or -1, respectively. Such comparisons are performed for all pairs of landmarkers. Thus $n \times (n-1)$ new metafeatures are added for each evaluation measure.
- Ratio: this approach also performs pairwise comparisons. However, it does so by using the ratios of the performances instead of assigning 1, 0 or -1 values. Given two landmarkers l_i and l_j, a metafeature with the value l_i/l_j is created.

As an example, let us consider two CF algorithms, A and B, and the NMAE performance measure. Given a data sample, they are applied to it and the corresponding NMAE score is computed. Table 1 illustrates such values and all the corresponding subsampling landmarkers. Notice Absolute is equal to the original NMAE, Ranking assigns the ranking of the algorithms, Pairwise assigns 1 to the best algorithm and -1 to the worst and Ratio presents the ratios of NMAE. It should be noted that the process is repeated for each evaluation measure.

Table 1: Example of relative landmarkers.

Algorithm	NMAE	Absolute Ranking		Pairwise Ratio
A	0.73	0.73	1	1	0.839
B	0.87	0.87	2	-1	1.192

3.2 Experimental procedure

The experimental setup used in this work is divided into baselevel and metalevel, referring, respectively, to the CF and classification stages of the process.

Baselevel The baselevel setup is concerned with the CF datasets, algorithms and measures used to evaluate the performance of CF algorithms on those datasets. The 38 datasets used come from different domains, namely Amazon Reviews [24], BookCrossing [36], Flixter [35], Jester [13], MovieLens [15], MovieTweetings [9], Tripadvisor [31], Yahoo! [32] and Yelp [34]. It is important to observe that each domain can contain more than one dataset.

The experiments were carried out with MyMediaLite, a software library for recommender systems [12]. Two CF tasks were addressed: Rating Prediction (RP) and Item Recommendation (IR). While RP aims to predict the rating an user would assign to a new instance, in IR the goal is to recommend a ranked list of items in terms of user preference. Since the tasks are different, so are the algorithms and evaluation measures. The following CF algorithms were used for RP: Matrix Factorization (MF), Biased MF (BMF), Latent Feature Log Linear Model (LFLLM), SVD++, 3 variants of Sigmoid Asymmetric Factor Model (SIAFM, SUAFM and SCAFM), User Item Baseline (UIB) and Global Average (GA). Regarding IR, the algorithms used are BPRMF, Weighted BPRMF (WBPRMF), Soft Margin Ranking MF (SMRMF), WRMF and Most Popular (MP). In IR, the algorithms are evaluated using NDCG, while in RP the algorithms are evaluated using NMAE. All experiments use 10-fold cross-validation.

Metalevel The metalevel is first characterized by the construction of the metafeatures. This work applies the statistical and/or information-theoretical metafeatures (described in Section 2.2) to all 38 CF datasets to extract the metafeatures for the Systematic approach. In order to extract the subsampling landmarkers (see Section 3.1), random samples of 10% for each of the original 38 CF datasets are extracted. Next, all algorithms are trained on said samples and their performance assessed via suitable evaluation metrics. This allows the extraction of what are referred as the Original relative landmarkers. Afterwards, the remaining relative landmarkers (Ranking, Pairwise and Ratio) are computed based on the values for the Original relative landmarker, as explained previously in Section 3.1. The entire process creates 5 different sets of metafeatures.

Two baselevel measures (NMAE and NDCG) are used to create two separate metatargets. The best algorithm, and consequently the target variable for each dataset, depends on the evaluation measures. For each pair dataset/evaluation

measure, the best algorithm is chosen as the target variable. Hence, we study the algorithm selection problem for 2 different metatargets. The final metadatabases, consisting of combinations of all different metafeatures and metatargets, are the experimental basis for the algorithm selection problem addressed here.

Since the model selection problem is approached here as a classification task, 11 classification algorithms from the `caret` package [19] representing several biases were chosen to address it: ctree, C4.5, C5.0, kNN, LDA, Naive Bayes, SVM (linear, polynomial and radial kernels), random forest and a baseline algorithm: Majority Vote. The Majority Vote does not take into account any metafeatures and always predicts the class which appears more often. Since the metadatasets have a reduced number of examples, the accuracy of the metalevel algorithms was estimated using a leave one out strategy.

Meta-level performance is measured in two ways. First, the accuracy of the meta-level prediction is assessed, i.e. whether the best algorithm is selected or not. However, in MtL it is also important to understand the impact on the baselevel performance of the meta-level prediction. It assesses how the algorithms recommended by the metamodels actually affect the baselevel performance. It is based on the comparison of baselevel performance between the algorithm selected by the metamodel and the best possible algorithm. The goal is to understand what is the actual cost of failing in the prediction of the best algorithm in terms of baselevel performance.

Consider a dataset D and the performance of n algorithms on D, $P_D = \{p_1, p_2, ..., p_n\}$, according to a specific evaluation measure. It is possible to create a ranking $R_D = \{a_1, a_2, ..., a_n\}$ in decreasing order of those performance values. This means that a_1 is the best algorithm on D, with a performance of p_1. Consider now that $\hat{a} = a_q$ is the algorithm predicted by a metamodel for dataset D, $q \in \{1, \ldots, n\}$. The impact at the baselevel of using the metamodel for algorithm selection is assessed by comparing p_q, the performance of the selected algorithm, with p_1, the performance of the best algorithm. In this work, this comparison is done in three ways: performance (PE), error (ER) and ranking (RK), which are given by: $PE(\hat{a}, D) = p_q$, $ER(\hat{a}, D) = p_1 - p_q$ and $RK(\hat{a}, D) = q$.

The three measures are computed for all datasets and averaged. The comparisons average performance (AP), average error (AE) and average rankings (AR) for a set of M datasets are defined as follows:

$$\begin{aligned} AP(\hat{a}_i) &= \frac{\sum_{i=1}^{M} PE(\hat{a}_i, D_i)}{M} \\ AE(\hat{a}_i) &= \frac{\sum_{i=1}^{M} ER(\hat{a}_i, D_i)}{M} \\ AR(\hat{a}_i) &= \frac{\sum_{i=1}^{M} RK(\hat{a}_i, D_i)}{M} \end{aligned} \tag{1}$$

where $\hat{a}_i$ is the algorithm selected for dataset D_i.

4 Results and Discussion

4.1 Metalevel evaluation

The metalevel accuracy performance for all strategies evaluated in this experimental study can be seen in Figure 3. For readability purposes, only the performance of the best metamodel is presented. After manual inspection, the choice fell on SVM with polynomial kernel. Two baseline methods are included for fair comparison. The Majority Vote baseline assesses if the MtL approach is finding any useful patterns. The Systematic metafeatures baseline assesses if there is any advantage in using the proposed subsampling landmarkers in the CF scenario.

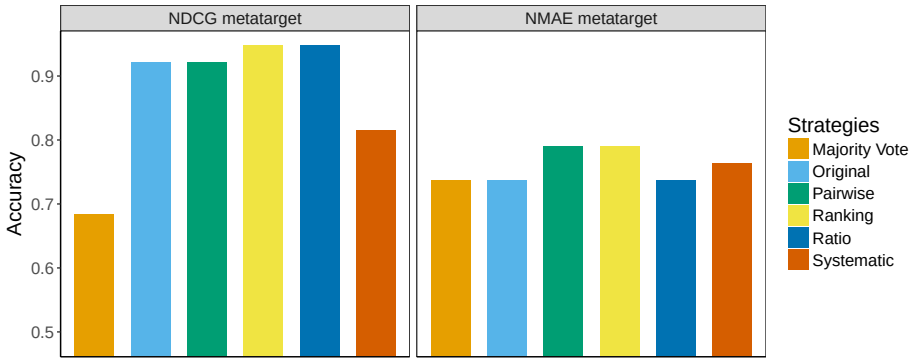

Fig. 3: Metalevel accuracy for all relative landmarkers and baselines.

Several observations can be made:

- Most landmarkers outperform the Majority Vote baseline. The exceptions are the Original and Ratio relative landmarkers in the NMAE metatarget.
- Landmarkers are better than the Systematic metafeatures in the NDCG metatarget.
- Landmarkers have slightly better performance than the Systematic metafeatures in the NMAE metatarget: this happens for the Ranking and Pairwise relative landmarkers.

The observations seem to indicate that 1) the metafeatures proposed are better than the baseline in terms of metalevel accuracy and 2) they seem to have slightly better performances than the Systematic metafeatures. To validate this assessment, we employ statistical significance tests using Critical Difference (CD) diagrams [8]. CD diagrams plot the average rank for each strategy and calculate the CD interval. Strategies connected by a CD line cannot be considered to perform differently. On the other hand, if two strategies are not connected by a CD line, they obtain, in fact, different performance, i.e. one strategy is ranked

higher than the other. To apply this framework, we combine the performances of all relative landmarkers and compare it with the baselines. The statistical validation confirms the observations made here (see Figure 4).

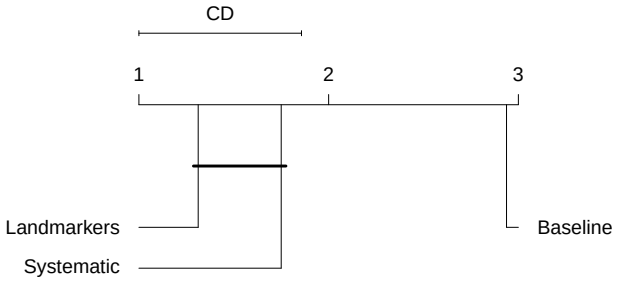

Fig. 4: CD diagram for the comparison of metafeature strategies.

4.2 Baselevel performance analysis

Figure 5 presents the baselevel performance analysis with regards to the Average Performance (discussed in Section 3.2). The oracle represents an ideal system that always predicts the best algorithm, and, thus, achieves the best possible performance. The performance of the methods were scaled such that it is represented as a percentage, where the oracle corresponds to 100%. As before, the Majority Vote and MtL with Systematic metafeatures are used as reference baselines.

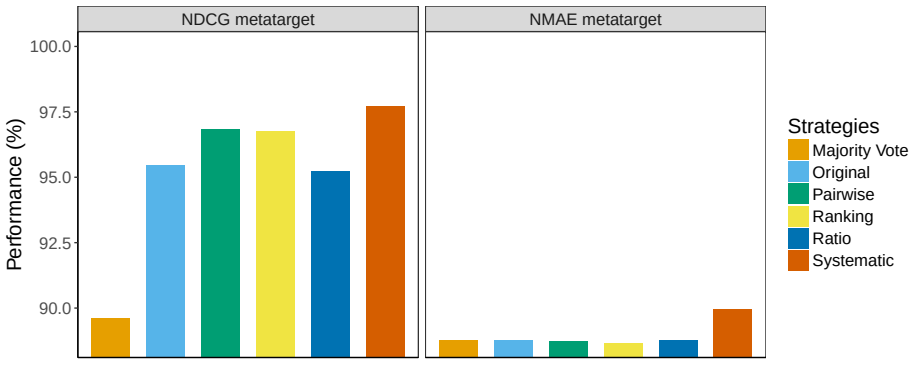

Fig. 5: Baselevel performance analysis regarding Average Performance.

The results show that the MtL approach using landmarkers:

– outperforms the Majority Voting baseline on the NDCG metatarget, but not on the NMAE metatarget.
– never beats the Systematic approach on either metatarget.

The results on the baselevel performance show that, although the landmarkers perform better in terms of metalevel accuracy, the same is not true for the baselevel performance analysis in terms of Average Performance. This shows that in spite of correctly predicting the best algorithm more often, the performance of the selected algorithms in terms of the baselevel evaluation measure is worse, on average. Thus, when the landmarkers fail to predict the correct best algorithm, they usually choose an algorithm with worse performance than when the systematic metafeatures fail to predict the best algorithm.

To validate this analysis, we performed the baselevel performance analysis, based on the Average Error (discussed in Section 3.2). The results are presented in Figure 6. It shows that the error obtained by the Systematic approach has a smaller difference to the best error on both metatargets, hence confirming our previous observation.

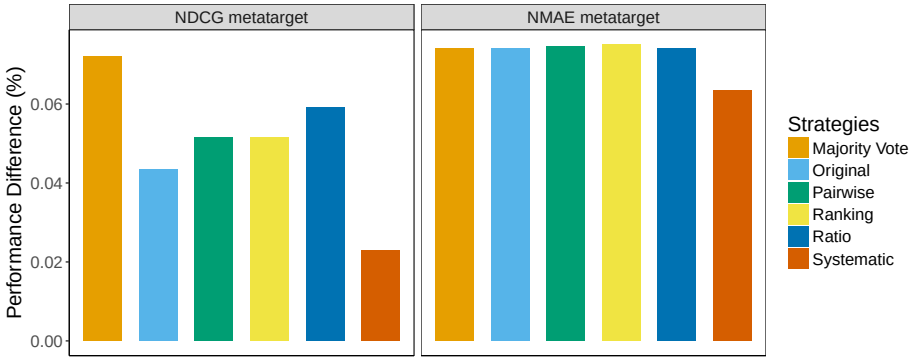

Fig. 6: Baselevel performance analysis regarding Average Error.

In another analysis, we looked towards the Average Ranking (discussed in Section 3.2). The results are presented in Figure 7. The baselines Majority Vote and Systematic are included for comparison with the landmarkers. The following observations regarding the landmarkers can be made:

– They rarely outperform the baseline Majority Voting: this only happens in 3 relative landmarkers in the NMAE metatarget.
– They are always worse than the Systematic metafeatures.

This analysis confirms the reason for the poor performance of landmarkers in terms of baselevel performance: the average ranking for the predicted CF algorithms is always higher than the Systematic approach. This means that the metamodels trained with landmarkers tend to recommend on average the second best

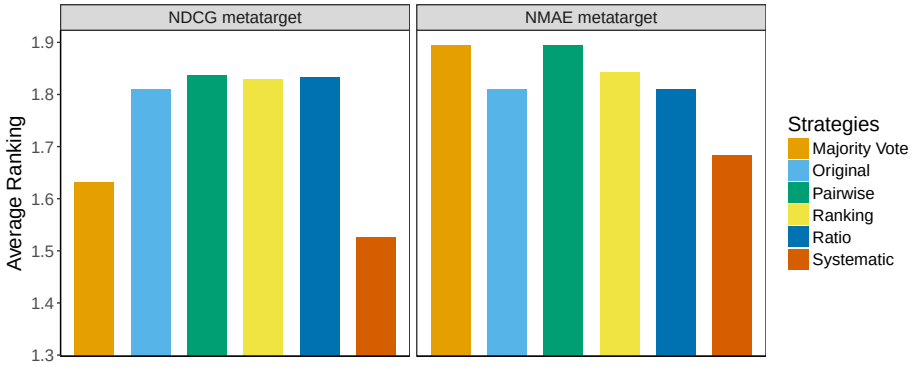

Fig. 7: Baselevel performance analysis regarding Average Ranking.

CF algorithm. When we consider the difference in terms of baselevel performance presented in Figure 6, one understands how costly these misclassifications are. These are surprising results, as they contradict the results in other tasks, where landmarkers are typically better than statistical and/or information-theoretical measures [3, 11, 17, 18, 20, 21, 25].

4.3 Metaknowledge

Metaknowledge is the knowledge about learning processes acquired through experience with past learning episodes [30]. It explains how specific metafeatures influence which one is the best algorithm. Such knowledge is typically embedded in the metamodels built and sometimes it is difficult to access and/or interpret. Furthermore, considering the vast amount of metamodels built and analyzed so far, it is difficult to discuss all the knowledge potentially obtained with this study. Here, we address this problem simply by analyzing metafeature importance.

We analyze all different strategies in terms of feature importance across all metatargets studied. To do so, we build Random Forest models and take advantage of its inbuilt mechanism for feature importance. We use the implementation available in the `caret` package [19], which computes an importance score for each feature. We average the importance percentages across all models which share the same metafeatures and present the results in Figure 8. Features with average importance below 10% were discarded.

The results show that the Systematic strategy contains the most influential metafeatures throughout. Special attention goes to the number of users and the skewness of the distribution of the sum of ratings per item. The remaining metafeatures focus on the kurtosis and entropy of the distribution of the sum of ratings of users. In terms of landmarkers, the Original relative landmarker highlights the importance of NMAE for SCAFM and LFLLM, while in the Ranking relative landmarker, the NDCG for MP is essential. In terms of relative landmarkers which focus on the comparison of landmarkers, the results show that

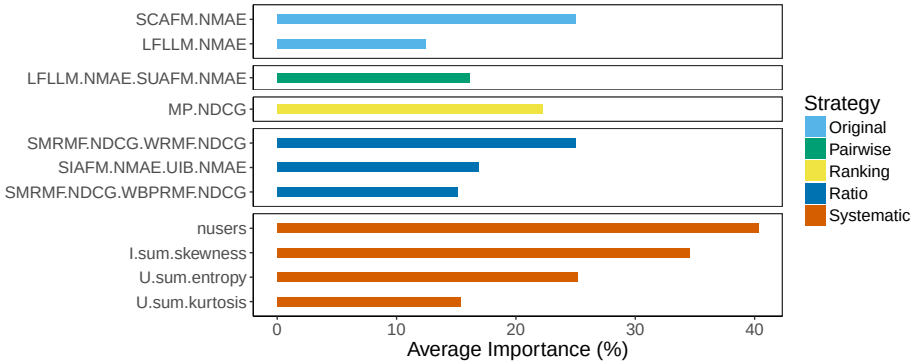

Fig. 8: Feature importance.

the comparison of NMAE performances of LFLLM and SUAFM algorithms are quite important among the Pairwise relative landmarkers. In the Ratio relative landmarkers, the ratios in terms of NDCG performance between SMRMF and both WRMF and WBPRMF and the ratio of NMAE between SIAFM and UIB are the most important ones. Although this analysis lacks some depth in terms of patterns found in the metamodels, it highlights two very important issues: 1) which are the most influential metafeatures and 2) since we are using landmarkers, which algorithms and evaluation measures are essential for the problem. Both are essential for future CF algorithm selection works.

5 Conclusions and Future Work

Landmarkers have been reported as a successful way to characterize problems in Metalearning approaches to algorithm selection in several tasks. In this work, we propose a set of subsampling landmarkers for Collaborative Filtering (CF) methods. The landmarkers were compared with the state of the art systematic metafeatures, based on statistical and/or information-theoretical measures, both in terms of metalevel accuracy and baselevel performance analysis. Somewhat surprisingly, in our experiments, their performance was not statistical significantly better than the systematic approach, in terms of metalevel accuracy. Furthermore, the impact on the baselevel performance produces worse results when using landmarkers in terms of average performance, average error and average rankings. Thus, the major contributions of this work are: 1) to propose subsampling landmarkers for CF tasks and 2) showing that the widely accepted assumption that landmarkers are better than statistical and/or information-theoretical metafeatures may not be true in CF. Future work includes the adaptation of other types of landmarkers for CF, using for instance different sampling strategies and the extension of the experimental procedures in order to allow more generic conclusions regarding the impact of metafeatures of different natures on the CF algorithm selection problem.

Acknowledgments This work is financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 under the Portugal 2020 Partnership Agreement, and through the Portuguese National Innovation Agency (ANI) as a part of project «FASCOM | POCI-01-0247-FEDER-003506».

References

1. Adomavicius, G., Tuzhilin, A.: 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 (2005)
2. Adomavicius, G., Zhang, J.: Impact of data characteristics on recommender systems performance. ACM Transactions on Management Information Systems 3(1), 1–17 (2012)
3. Bensusan, H., Kalousis, A.: Estimating the Predictive Accuracy of a Classifier. European Conference on Machine Learning pp. 25–36 (2001)
4. Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A.: Recommender systems survey. Knowledge-Based Systems 46, 109–132 (Jul 2013)
5. Brazdil, P., Giraud-Carrier, C., Soares, C., Vilalta, R.: Metalearning: Applications to Data Mining. Springer, 1 edn. (2009)
6. Brazdil, P., Soares, C., da Costa, J.: Ranking Learning Algorithms : Using IBL and Meta-Learning on Accuracy and Time Results. Machine Learning 50(3), 251–277 (2003)
7. Cunha, T., Soares, C., de Carvalho, A.C.: Selecting Collaborative Filtering algorithms using Metalearning. In: European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 393–409 (2016)
8. Demšar, J.: Statistical Comparisons of Classifiers over Multiple Data Sets. Journal of Machine Learning Research 7, 1–30 (2006)
9. Dooms, S., De Pessemier, T., Martens, L.: MovieTweetings: a Movie Rating Dataset Collected From Twitter. In: CrowdRec at RecSys 2013 (2013)
10. Ekstrand, M., Riedl, J.: When Recommenders Fail: Predicting Recommender Failure for Algorithm Selection and Combination. ACM Conference on Recommender Systems pp. 233–236 (2012)
11. Fürnkranz, J., Petrak, J., Bradzil, P., Soares, C.: On the use of fast subsampling estimates for algorithm recommendation. Tech. rep. (2002)
12. Gantner, Z., Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: MyMediaLite: A Free Recommender System Library. In: ACM Conference on Recommender Systems. pp. 305–308 (2011)
13. Goldberg, K., Roeder, T., Gupta, D., Perkins, C.: Eigentaste: A Constant Time Collaborative Filtering Algorithm. Information Retrieval 4(2), 133–151 (2001)
14. Griffith, J., O'Riordan, C., Sorensen, H.: Investigations into user rating information and predictive accuracy in a collaborative filtering domain. In: ACM Symposium on Applied Computing. pp. 937–942 (2012)
15. GroupLens: MovieLens datasets (2016), `http://grouplens.org/datasets/movielens/`
16. Herlocker, J.L., Konstan, J.a., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems 22(1), 5–53 (Jan 2004)

17. Kanda, J., de Carvalho, A., Hruschka, E., Soares, C., Brazdil, P.: Meta-learning to select the best meta-heuristic for the Traveling Salesman Problem: A comparison of meta-features. Neurocomputing 205, 393–406 (2016)
18. Kück, M., Crone, S.F., Freitag, M.: Meta-Learning with Neural Networks and Landmarking for Forecasting Model Selection - An Empirical Evaluation of Different Feature Sets Applied to Industry Data Meta-Learning with Neural Networks and Landmarking for Forecasting Model Selection. In: International Joint Conference on Neural Networks. pp. 1499–1506 (2016)
19. Kuhn, M.: caret: Classification and Regression Training (2016), `https://CRAN.R-project.org/package=caret`, r package version 6.0-73
20. Ler, D., Koprinska, I., Chawla, S.: Utilizing regression-based landmarkers within a meta-learning framework for algorithm selection. Tech. rep., School of Information Technologies University of Sydney (2005)
21. Ler, D., Koprinska, I., Chawla, S.: Utilizing regression-based landmarkers within a meta-learning framework for algorithm selection. In: Proceedings of the ICML-2005 Workshop on Metalearning. pp. 44–51 (2005)
22. Lü, L., Medo, M., Yeung, C.H., Zhang, Y.C., Zhang, Z.K., Zhou, T.: Recommender systems. Physics Reports 519(1), 1–49 (Oct 2012)
23. Matuszyk, P., Spiliopoulou, M.: Predicting the Performance of Collaborative Filtering Algorithms. In: International Conference on Web Intelligence, Mining and Semantics. pp. 38:1—-38:6 (2014)
24. McAuley, J., Leskovec, J.: Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. In: ACM Conference on Recommender Systems. pp. 165–172 (2013)
25. Pfahringer, B., Bensusan, H., Giraud-Carrier, C.: Meta-Learning by Landmarking Various Learning Algorithms. In: International Conference on Machine Learning. pp. 743–750 (2000)
26. Pinto, F., Soares, C., Mendes-Moreira, J.: Towards automatic generation of Metafeatures. In: Pacific Asia Conference on Knowledge Discovery and Data Mining. pp. 215–226 (2016)
27. Prudêncio, R.B., Ludermir, T.B.: Meta-learning approaches to selecting time series models. Neurocomputing 61, 121–137 (Oct 2004)
28. Rossi, A.L.D., de Carvalho, A.C.P.D.L.F., Soares, C., de Souza, B.F.: MetaStream: A meta-learning based method for periodic algorithm selection in time-changing data. Neurocomputing 127, 52–64 (Mar 2014)
29. Serban, F., Vanschoren, J., Bernstein, A.: A survey of intelligent assistants for data analysis. ACM Computing Surveys V(212), 1–35 (2013)
30. Vanschoren, J.: Understanding machine learning performance with experiment databases. Ph.D. thesis, Katholieke Universiteit Leuven (2010)
31. Wang, H., Lu, Y., Zhai, C.: Latent Aspect Rating Analysis Without Aspect Keyword Supervision. In: ACM SIGKDD. pp. 618–626. KDD '11, ACM (2011)
32. Yahoo!: Webscope datasets (2016), `https://webscope.sandbox.yahoo.com/`
33. Yang, X., Guo, Y., Liu, Y., Steck, H.: A survey of collaborative filtering based social recommender systems. Computer Communications 41, 1–10 (Mar 2014)
34. Yelp: Yelp Dataset Challenge (2016), `https://www.yelp.com/dataset_challenge`
35. Zafarani, R., Liu, H.: Social computing data repository at ASU (2009), `http://socialcomputing.asu.edu`
36. Ziegler, C.N.C., McNee, S.M.S., Konstan, J.a.J., Lausen, G.: Improving recommendation lists through topic diversification. In: Proceedings of the 14th international conference on World Wide Web WWW 05. p. 22. WWW '05, ACM (2005)