Context-Aware Movie Recommendations: An Empirical Comparison of Pre-filtering, Post-filtering and Contextual Modeling Approaches

Pedro G. Campos[1,2], Ignacio Fernández-Tobías[2], Iván Cantador[2], and Fernando Díez[2]

[1] Departamento de Sistemas de Información
Universidad del Bío-Bío 4081112 Concepción, Chile
[2] Escuela Politécnica Superior
Universidad Autónoma de Madrid 28049 Madrid, Spain
pgcampos@ubiobio.cl,
{pedro.campos,i.fernandez,ivan.cantador,fernando.diez}@uam.es

Abstract. Context-aware recommender systems have been proven to improve the performance of recommendations in a wide array of domains and applications. Despite individual improvements, little work has been done on comparing different approaches, in order to determine which of them outperform the others, and under what circumstances. In this paper we address this issue by conducting an empirical comparison of several pre-filtering, post-filtering and contextual modeling approaches on the movie recommendation domain. To acquire confident contextual information, we performed a user study where participants were asked to rate movies, stating the time and social companion with which they preferred to watch the rated movies. The results of our evaluation show that there is neither a clear superior contextualization approach nor an always best contextual signal, and that achieved improvements depend on the recommendation algorithm used together with each contextualization approach. Nonetheless, we conclude with a number of cues and advices about which particular combinations of contextualization approaches and recommendation algorithms could be better suited for the movie recommendation domain.

Keywords: Context-aware recommender systems, pre-filtering, post-filtering, contextual modeling, time context, social context.

1 Introduction

Recommender systems (RS) suggest items to users relying on preferences –usually expressed in the form of numeric ratings– of similar-minded people. Context-Aware Recommender Systems (CARS) additionally take into consideration contextual information (e.g. time, location, social companion, and mood) associated to the collected preferences. In this way, CARS can discriminate the interest a user may have in a particular item within different contexts and situations.

Several approaches have been proposed to properly deal with contextual information. Adomavicius et al. [1, 2] distinguish three main types of CARS: those

C. Huemer and P. Lops (Eds.): EC-Web 2013, LNBIP 152, pp. 137–149, 2013.

based on *contextual pre-filtering*, which prune the available user preference data according to the target recommendation context, prior to applying a recommendation algorithm; those based on *contextual post-filtering*, which apply a recommendation algorithm on the original preference data, and afterwards adjust the generated recommendations according to the target recommendation context; and those based on *contextual modeling*, which incorporate contextual information into the model used for generating recommendations.

In the literature, pre-filtering, post-filtering and contextual modeling have been proven to improve the performance of recommendations in a wide array of domains and applications. Despite individual improvements, little work has been done on comparing different approaches, in order to determine which of them outperform the others, and under what circumstances. In this paper we address this issue by conducting an empirical comparison of several pre-filtering, post-filtering and contextual modeling approaches on the movie recommendation domain. Specifically, we frame the problem as a multi-label classification task, where recommender systems are required to properly classify a given test pattern (composed of user preference, item attribute and/or contextual data) with a class label corresponding to certain rating value. This lets us to directly use well known Machine Learning algorithms for contextual modeling, and compare pre-/post-filtering with context modeling.

A major difficulty for evaluating CARS is the lack of availability of context-enriched datasets. Obtaining contextual information imposes an extra effort from the user to explicitly state or describe the current context, or system/device requirements to automatically infer the current context, e.g. by capturing time and location signals, or by analyzing the user's interactions with the system. This fact makes it difficult to gain access to contextual data really valuable for evaluation. Addressing this problem, in order to acquire confident contextual information, we performed a user study where participants were asked to rate movies, stating the time and social companion with which they preferred to watch the rated movies.

In the study we aimed to address the following research questions: **RQ1**, which CARS approaches –pre-filtering, post-filtering or contextual modeling– are able to better predict the rating a user would assign to a movie in a particular context? And **RQ2**, which contextual signal –time or social companion (or a combination of both)– provides more useful information for predicting the above rating?

The results of our evaluation show that there is neither a clear superior contextualization approach nor an always best contextual signal, and that achieved improvements depend on the underlying recommendation algorithm used together with each contextualization approach. Nonetheless, we conclude with a number of cues and advices about which particular combinations of contextualization approaches and recommendation algorithms could be better suited for the movie recommendation domain.

The reminder of the paper is organized as follows. In Section 2 we discuss related work. In Section 3 we describe the analyzed contexts, and the evaluated contextualization and recommendation approaches. In Section 4 we describe the experiments conducted, and report the results obtained. Finally, in Section 5 we provide some conclusions and future research directions of our work.

2 Related Work

Context is a multifaceted concept that has been studied in different research disciplines, and thus has been defined in multiple ways [2]. Quoting [3], "context is any information that can be used to characterize the situation of an entity." In the case of RS, an entity can be a user, an item, or an experience the user is evaluating [4]. Hence, any information signal –e.g. location, time, social companion, device, and mood– regarding the situation in which a user experiences an item can be considered as context.

Generally speaking, the recommendation problem relies on the notion of *rating* as a mechanism to capture user preferences for different items. Two common strategies to RS are *content-based* (CB) recommendations, which recommends items similar to those preferred by the user in the past, and *collaborative filtering* (CF), which recommends items preferred in the past by similar-minded people. *Hybrid* recommenders combine CB and CF in order to overcome particular limitations of each individual strategy. For any of the above strategies, recommendation approaches can be classified as *heuristic-based* or *model-based*. Heuristic-based approaches utilize explicit formulas that aggregate collected user preferences to compute item relevance predictions. Model-based approaches, in contrast, utilize collected user preferences to build (machine learning) models that, once built, provide item relevance predictions [5].

Traditional RS exploit only user and item profile data associated to past ratings in order to predict ratings of unseen items [1], and they do not take any contextual information into account. Extending the rating notion, Adomavicius et al. [1] incorporate additional dimensions assuming that the context can be represented as a set of contextual dimensions. By using this formulation, CARS can be classified as *contextual pre-filtering*, *contextual post-filtering*, and *contextual modeling* systems [1, 2]. In contextual pre-filtering the target recommendation context –i.e., the context in which the target user expects to consume the recommended items– is used to filter user profile data relevant to such context before rating prediction computation. In contextual post-filtering rating predictions are adjusted according to the target context after being computed (on entire user profiles). In both cases traditional non-contextualized recommendation algorithms can be utilized, as the contextualization involves independent pre- or post-processing computations. On the other hand, contextual modeling incorporates context information directly into the model used to estimate rating predictions.

Different pre-filtering, post-filtering and contextual modeling approaches can be found in the literature. For instance, Adomavicius and colleagues [1] propose a pre-filtering based on pruning all ratings irrelevant to the target context. Baltrunas and Amatriain [6] created contextual micro-profiles, each of them containing ratings in a particular context, as a pre-filtering strategy aimed to better detect the user's preferences for specific time contexts. Baltrunas and Ricci [7, 8] proposed a pre-filtering technique called *Item Splitting*. This technique divides (i.e., splits) preference data for items according to the context in which such data were generated, assuming that there exist significant differences in the user preferences received by items among contexts. Panniello and colleagues [9] present a post-filtering strategy that penalizes the recommendation of items with few ratings in the target context.

One of the first contextual modeling approaches is presented in [10], where several contextual dimensions including time, social companion, and weather were incorporated into a Support Vector Machine model for recommendation. In [11] Karatzoglou and colleagues used Tensor Factorization to model *n*-dimensional contextual information. They called their approach as multiverse recommendation because of its ability to bridge data pertaining to different contexts (universes of information) into a unified model. Another example is given in [12], where Factorization Machines were used to combine different types of contextual information.

Although different approaches and algorithms have been developed for exploiting contextual information, little work has been done on comparing them, in order to better understand the circumstances that affect their performance. As noted by [2], context-aware recommendation is a relatively unexplored area, and still needs a much better comprehension. The most notable work in comparing CARS approaches correspond to the series of studies from Panniello et al. [9, 13–15]. They compare CARS approaches using heuristic-based CF algorithms. Differently from that work, we evaluate CARS using model-based as well as heuristic based CF algorithms, and moreover we include a hybrid approach that exploits CB user preferences in a CF fashion, providing a more diverse set of configurations and enabling a broader analysis of existing CARS approaches.

3 Evaluating Context-Aware Recommendation

We compare several pre-filtering, post-filtering and contextual modeling RS, using different contextual signals. In this section we describe the analyzed contextual signals and acquired information, and detail the evaluated CARS.

3.1 Analyzed Contextual Signals

We focus on two types of contextual signals: Time context and social context (i.e., the user's current companion). Exploiting time context has been proved to be an effective approach to improve recommendation performance, as shown e.g. in the Netflix Prize competition. Additionally, social context has also been found as a source for improving CARS performance [1, 2].

Among the existing contextual dimensions, time context –i.e., contextual attributes related to time, such as *time of the day*, *day of the week*, and *current time/date*– can be considered as the most versatile one. Time can be represented both as continuum information (e.g. current date/time), and as periodic, discrete information (e.g. day of the week). This lets classify Time-aware Recommender Systems (TARS) according to the way they model time information: *continuous TARS* –which model time context information as a continuous variable– and *categorical TARS* –which model time as one or more categorical variables [16]. Interestingly, when timestamps are available, both continuous and categorical context information can be extracted and exploited.

In general, collecting time information of user interactions with a system does not require additional user effort nor impose strict system/device requirements. Moreover, it has been used as a key input for achieving significant improvements on recommendation accuracy [17]. Hence, the timestamps of collected user preferences are valuable, easy-to-collect data for improving recommendations. Due to these benefits, recent years have been prolific in the research and development of TARS. However, it is important to note that if a RS collects ratings instead of usage/consumption data, the collected timestamps do not necessarily correspond to item usage/consumption time, and thus may not be considered as the context in which the user prefers to use/consume the item.

Some other contextual signals can be inferred with appropriated devices, such as location or weather, by means e.g. of mobile devices with GPS. In contrast, for other contextual signals there may not exist devices to automatically infer them (or they may be unfeasible due to cost or physical constraints), such as mood or social (companion) context, but may represent important signals for determining user preferences. In particular, social context has been proved as a key factor for the users' actions [18, 19]. One way to obtain social context signals is to take advantage of online social networks such as Facebook[1] and Twitter[2], which have given raise to social network-based recommender systems [19]. However, the context information obtained in this way is used to find general preferences of related users (those connected in the social network), and generally does not correspond to the item usage/consumption context of the target user.

Thus, in order to count with confident context signals related to user preferences, we collected a movie ratings dataset, including time and social context information, as described in the next subsection.

3.2 Acquired Contextual Information

We collected a dataset of user preferences for movies. Since we were interested in the effect of time and social context on user interests, we built our own Web application, and asked users (recruited via social networks) for using it to provide personal ratings for movies they had watched. Specifically, participants rated a freely chosen set of movies by using a rating scale from 1 to 5 (1 representing no user interest, and 5 for a maximum user interest). The final dataset used in our study consisted of 481 ratings from 67 users given to 174 movies. The rating distribution of the dataset was 2.7%, 7.7%, 19.1%, 44.7%, and 25.8% for ratings values of 1, 2, 3, 4, and 5 respectively. This non-uniform distribution is important to take into account when analyzing the results reported in Section 4.

In addition to ratings, participants stated which time of the day (*morning*, *afternoon*, *night*, and *indifferent*), which period of the week (*working day*, *weekend*, and *indifferent*), and with whom (*alone*, *with my couple*, *with my family*, *with friends*, and *indifferent*) they would prefer to watch the rated movies.

In order to gain a first insight about the context influence on user preference, we analyze the differences in ratings between movie genres and contexts. Figure 1 shows the average movie rating value computed over the different contexts in our study,

[1] `http://www.facebook.com`
[2] `http://www.twitter.com`

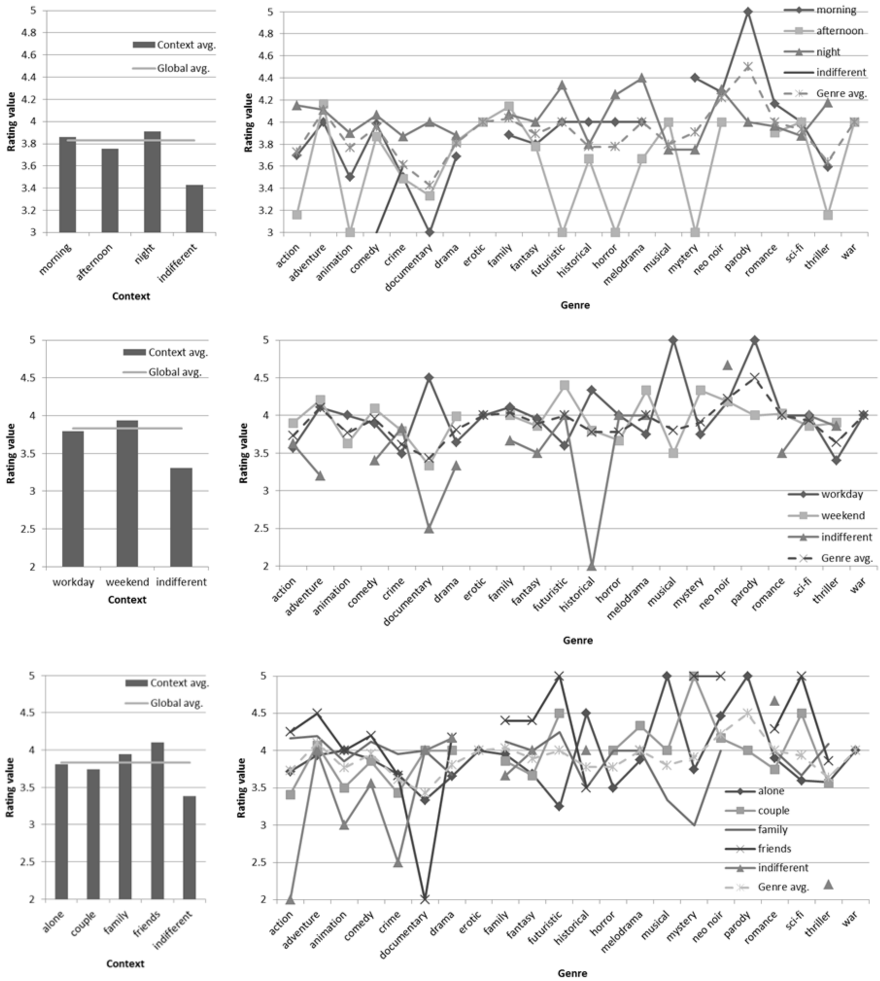

Fig. 1. Average movie rating values computed over different contexts and movie genres on the context-enriched dataset collected in our study

globally and per movie genre. As shown in the figure, there are important variations in average rating values between different contexts. These results show that time and social context information has an impact on user preferences in the movies domain, and thus, can be useful in the rating prediction task.

3.3 Evaluated Context-Aware Recommender Systems

We evaluated several pre-filtering, post-filtering and contextual modeling approaches. In the pre-filtering case, we used the exact pre-filtering strategy suggested by

Adomavicius and colleagues [1], and the Item Splitting technique proposed by Baltrunas and Ricci [4, 7, 8]. In the post-filtering case, we used the filtering strategy presented by Panniello and colleagues in [9]. Finally, in the contextual modeling case, we evaluated several classifiers developed by the Machine Learning community, including Naïve Bayes, Random Forest, MultiLayer Perceptron (MLP), and Support Vector Machine (SVM) algorithms [20, 21]. All the classifiers were built with vectors of content-based attributes corresponding to user and item genre information, and different contextual signals.

In exact pre-filtering (PeF), only ratings relevant to the target context are used to compute rating predictions with a context-unaware recommendation algorithm. Specifically, the k-nearest neighbor (kNN) algorithm [22] was used as underlying recommendation algorithm.

Item Splitting (IS) is a variant of context pre-filtering. This method divides (i.e., splits) preference data for items according to the context in which such data were generated, in cases where there exist significant differences in the user preferences received by items among contexts. In order to determine whether such differences are significant, an impurity criterion is used. When an item is split, two new (artificial) items are created, each one with a subset of the preference data from the original item, according to the associated context value. One of these new items corresponds with the preferences generated on one contextual condition, and the other (artificial) item corresponds with the remainder preferences. The original item is removed from the dataset, and afterwards, any non-contextualized recommendation algorithm is performed on the modified dataset.

In order to decide whether to split the set of ratings given to an item i, we utilized several impurity criteria, based on Baltrunas and Ricci's findings [4]. An impurity criterion $ic(i, s)$ returns a score of the differences between the ratings given to an item i in a split $s \in S$, where S represents the set of possible contextual splits.

The selected impurity criteria were: $ic_{IG}(i, s)$, which measures the information gain given by s to the knowledge of item i rating; $ic_M(i, s)$, which estimates the statistical significance of the difference in the means of ratings associated to each context in s using the t-test; and $ic_P(i, s)$, which estimates the statistical significance of the difference between the proportion of high and low ratings in each context of s using the two-proportion z-test. A set of item ratings is split if the corresponding criterion returns a score above certain threshold. If several splits obtain a score above the threshold, the split with highest score is used. Note that by using this heuristic, when more than one context variable is used for splitting (e.g. *time of the day* and *period of the week*), the impurity score lets select dynamically the best context variable for performing the split of a given item –the one that maximizes the differences in item rating patterns among contextual conditions. We used kNN and matrix factorization (MF) [17] collaborative filtering algorithms separately as recommendation strategies after IS.

In contextual post-filtering (PoF), rating predictions are generated by a context-unaware algorithm in a first stage, and then the predictions are contextualized according to the target context. We used the same kNN rating prediction algorithm used with pre-filtering approaches. The contextualization of rating predictions was performed by a filtering strategy presented in [9], which penalizes the recommendation of items that are not relevant in the target context as follows. The relevance of an item i for the target user u in a particular context c is approximated by the probability $P_c(u,i,c) = \frac{|U_{u,i,c}|}{k}$, where k is the number of neighbors used by kNN and $U_{u,i,c} = \{v \in N(u) | r_{v,i,c} \neq \emptyset\}$, that is, the user's neighbors v in the neighborhood of u, $N(u)$, who have rated/consumed item i in context c. The item relevance is determined by a threshold value τ_{P_c} (set to 0.1 in our experiments) that is used to contextualize the ratings as follows:

$$F(u,i,c) = \begin{cases} F(u,i) & \text{if} \quad P_c(u,i,c) \geq \tau_{P_c} \\ F(u,i) - 0.5 & \text{if} \quad P_c(u,i,c) < \tau_{P_c} \end{cases}$$

where $F(u,i)$ denotes the context-unaware rating prediction given by a RS, and $F(u,i,c)$ denotes the context-aware rating prediction.

The Machine Learning algorithms used for contextual modeling provide a score distribution for a rating (class label) in the space of rating values 1, 2, 3, 4 and 5. These algorithms were trained with a set of patterns composed of attributes describing user and item characteristics, and attributes containing contextual information. The algorithms exploit these patterns to compute score distributions. In this way, preferences of individual users were exploited in a collaborative way. The analyzed user and item characteristics correspond to movie genres. For each user u, the value of attribute a_m was the number of u's liked/preferred items with genre m. For each item i, the value of attribute a_n was 1 if i had the genre n, and 0 otherwise.

4 Experiments and Results

To determine which contextualization approach performs the best, we evaluated the CARS described in Section 3.3 on the context-enriched dataset collected in our study, and using the contextual information described in Section 3.2. In this section we detail the followed experimental setting, and discuss the obtained results.

4.1 Experimental Setting

We performed 10-fold cross-validation in all the experiments. In the pre-filtering and post-filtering cases, we used the kNN and MF implementations provided by the Apache Mahout project[3], with $k = 30$ and the Pearson Correlation for kNN, and 60 factors for the MF algorithm. To obtain full coverage, in cases where an algorithm was unable to compute a prediction, the average dataset rating was provided as prediction. In the contextual modeling cases, we used the classifier implementations provided in Weka[4].

We computed the accuracy of the evaluated recommendation approaches in terms of the correct classification rate for each rating value (*acc1*, *acc2*, *acc3*, *acc4*, and *acc5*), and the weighted overall correct classification rate (*acc*) [23]. We also computed the Area under the Curve (*AUC*) metric [24]. These metrics allow us to observe the performance of the tested approaches taking the pattern's class distribution into account.

[3] `http://mahout.apache.org/`

[4] `http://www.cs.waikato.ac.nz/ml/weka/`

4.2 Results

Table 1 shows the best results obtained for each of the tested approaches on our context-enriched dataset. The results are grouped according to the contextualization approach (pre- and post-filtering or contextual modeling), and the type of profile data provided to

Table 1. Performance values obtained by the pre-filtering, post-filtering and contextual modeling-based recommender systems built with different profile types. Global top values are in bold, and best values for each profile type are underlined.

	Profile type	Classifier	acc1	acc2	acc3	acc4	acc5	acc	AUC
Contextual Pre- and Post-Filtering	***user and item genres***	*kNN*	23.077	5.405	6.522	87.442	8.871	43.659	0.494
		MF	0.000	21.622	23.913	67.442	30.645	44.283	0.626
	user and item genres + time contexts	*PeF*	7.692	0.000	1.087	99.070	0.000	44.699	0.466
		IS_ic_{IG} + *kNN*	23.077	2.703	4.348	87.442	8.871	43.035	0.493
		IS_ic_{M} + *kNN*	23.077	5.405	4.348	86.047	10.484	43.035	0.514
		IS_ic_{P} + *kNN*	23.077	5.405	3.261	88.372	8.871	43.451	0.504
		IS_ic_{IG} + *MF*	0.000	21.622	23.913	66.512	31.452	44.075	0.625
		IS_ic_{M} + *MF*	0.000	21.622	25.000	66.977	32.258	44.699	0.636
		IS_ic_{P} + *MF*	0.000	18.919	25.000	66.977	33.065	44.699	0.635
		PoF	23.077	5.405	6.522	88.372	8.871	44.075	0.510
	user and item genres + social context	*PeF*	0.000	0.000	1.087	95.814	1.613	43.451	0.468
		IS_ic_{IG} + *kNN*	0.000	2.703	5.435	88.837	9.677	43.451	0.508
		IS_ic_{M} + *kNN*	23.077	5.405	6.522	87.442	8.871	43.659	0.494
		IS_ic_{P} + *kNN*	7.692	2.703	5.435	85.581	6.452	41.372	0.486
		IS_ic_{IG} + *MF*	0.000	21.622	25.000	66.512	29.839	43.867	0.625
		IS_ic_{M} + *MF*	0.000	21.622	23.913	67.442	29.839	44.075	0.626
		IS_ic_{P} + *MF*	0.000	**24.324**	22.826	67.907	32.258	44.906	0.639
		PoF	23.077	5.405	6.522	86.512	8.871	43.243	0.493
	user and item genres + all contexts	*PeF*	0.000	0.000	0.000	**100.000**	0.000	44.699	0.462
		IS_ic_{IG} + *kNN*	0.000	2.703	4.348	88.372	8.871	42.827	0.510
		IS_ic_{M} + *kNN*	23.077	5.405	4.348	86.047	10.484	43.035	0.514
		IS_ic_{P} + *kNN*	7.692	2.703	3.261	88.372	4.839	41.788	0.489
		IS_ic_{IG} + *MF*	0.000	21.622	25.000	66.047	29.839	43.659	0.625
		IS_ic_{M} + *MF*	0.000	21.622	25.000	66.977	32.258	44.699	0.636
		IS_ic_{P} + *MF*	0.000	21.622	22.826	68.372	33.871	45.322	0.642
		PoF	23.077	5.405	6.522	86.977	8.871	43.451	0.499
Contextual Modeling	***user and item genres***	*Naïve Bayes*	**38.462**	0.000	6.522	73.488	31.452	43.243	0.615
		Random Forest	0.000	21.622	25.000	62.791	51.613	**47.817**	0.669
		MLP	0.000	13.514	**29.348**	59.070	46.774	45.114	0.646
		SVM	0.000	16.216	20.652	54.884	37.903	39.501	0.554
	user and item genres + time contexts	*Naïve Bayes*	**38.462**	0.000	8.696	72.093	32.258	43.243	0.613
		Random Forest	15.385	13.514	23.913	61.395	48.387	45.946	0.649
		MLP	0.000	8.108	**29.348**	54.419	43.548	41.788	0.648
		SVM	23.077	16.216	21.739	59.535	40.323	43.035	0.573
	user and item genres + social context	*Naïve Bayes*	**38.462**	0.000	6.522	71.628	33.871	43.035	0.619
		Random Forest	0.000	16.216	20.652	60.930	**54.032**	46.362	**0.672**
		MLP	7.692	13.514	23.913	57.674	41.935	42.412	0.631
		SVM	7.692	10.811	18.478	59.070	41.129	41.580	0.563
	user and item genres + all contexts	*Naïve Bayes*	**38.462**	0.000	8.696	71.163	33.871	43.243	0.617
		Random Forest	7.692	13.514	22.826	63.721	44.355	45.530	0.666
		MLP	7.692	18.919	21.739	57.209	44.355	42.827	0.631
		SVM	15.385	13.514	17.391	63.721	37.903	43.035	0.568

each recommendation algorithm. In the IS approaches, we tested different threshold values for the considered impurity criteria. We finally used 0.8, 2.1 and 1.2 as threshold values for ic_{IG}, ic_M and ic_P respectively. We also tested different settings for specific parameters of each classifier used in contextual modeling, obtaining similar results.

We observe that the results of the ***AUC* metric** are close and above 0.5 for most of the approaches, with the exception of kNN and PeF, which got the worst performance. Moreover, the results obtained by PeF are worse than those obtained by kNN without contextualization in all cases. We also observe that **IS pre-filtering** improves the results provided by the underlying recommendation algorithm, particularly when it is used with the ic_P impurity criterion and the MF recommender. When using kNN, the ic_M impurity criterion improves the base recommendation algorithm. PoF shows a slightly better AUC than kNN. The Random Forest contextual modeling method obtains the best values of AUC, followed by MLP. The latter results are similar to those obtained by the IS + MF method.

For the ***acc* metric**, we observe that the **contextual modeling** approaches in general obtain the best values, although this may be due to the accuracy of the classifiers, as can be observed from the results using only genre profile data. On the other hand, the IS approach is not useful for improving kNN results. We observe that for PeF the good results are related with an almost perfect result on *acc4* metric. This is due to the low coverage induced by PeF, which forces to present the dataset average rating (3.83) as prediction in many cases, which is associated to the class label 4, but with near zero accuracy for the other rating values. On the other hand, PoF and contextual modeling approaches show a better balance of accuracy among the different rating values, as contextual modeling approaches also do.

Regarding the contribution of the contextual signals, we observe that **the evaluated CARS take advantage differently from each type of context information**. IS pre-filtering shows better performance by using all contextual signals. PoF, differently, shows better performance when it uses only time context information. In the case of the contextual modeling approaches, Naïve Bayes and Random Forest algorithms show better *AUC* when exploiting social context, although *acc* is not improved when using such contextual signal. SVM, on the other hand, shows better performance when it uses time context information, and MLP obtains only a slight improvement on *AUC* from using time context information. Interestingly, using all contextual signals does not lead to consistent improvements of the contextual modeling approaches.

One possible reason for the low performance obtained when using all the contextual signals is the increased dimensionality introduced by the additional information that must be handled by the CARS. This higher dimensionality is traduced in increased data sparsity in the case of PeF-based CARS (because PeF uses rating data only from the same context), and overfitting in the case of the Machine Learning-based contextual modeling CARS analyzed here, due to the increased number of pattern attributes.

Summing up, based on the reported results, we could conclude that **there is no unique superior CARS for improving rating predictions on the movie domain**, and that performance improvements have a strong dependency with the underlying recommendation algorithm used with the contextualization approach. Moreover, **no contextual signal seems to be more informative than other** for all the evaluated

CARS. Similarly to findings in previous research comparing some CARS approaches on e-commerce applications [9], the identification of the best performing approach requires a time-consuming evaluation and comparison of several CARS on the target data. Finally, we could also conclude that **using larger number of contextual signals does not necessarily lead to better CARS performance**, and the contribution given to a contextual signal depends on the particular combination of contextualization approach and recommendation algorithm used.

5 Conclusions and Future Work

In this paper we have compared diverse CARS, including various pre-filtering, post-filtering and contextual modeling approaches. To address the lack of available context-enriched data, we conducted a user study, and collected a dataset of movie ratings and information about the time and the social company preferred by the users for watching the rated movies.

The results obtained in our experiments show that there is not a CARS clearly superior to others, since performance values depend to a large extent on the particular combination of the contextualization approach and the underlying recommendation algorithm used to instantiate the approach. We observed that an **Item Splitting pre-filtering using Matrix Factorization**, as well as a **Random Forest-based contextual modeling** had a general good performance on the collected dataset, independently of the contextual information used, and thus, may represent good choices for the movie domain when different contextual signals are available (**RQ1**).

The analysis of contextual information also showed that the highest contribution is not given consistently by any of the signals alone, nor their combination. Thus, we conclude that **using all available context information does not have to be the best solution**, due to the higher dimensionality introduced by the context information (the "curse of dimensionality" [20]). Despite this fact, the **Item Splitting-based approach was able to properly deal with the combination of context signals**, possibly due to its ability of not discarding rating data, but splitting them according to the context only in cases where a significant difference is observed (**RQ2**).

The study reported in this paper has some limitations. In particular, the used dataset have a limited number of ratings, and experiments with a much larger dataset (and additional datasets) should be conducted, in order to test whether results obtained in this work are general or not. Nonetheless, we remind that in our dataset (and differently to publicly available datasets with rating timestamps), the contextual information associated to each rating corresponds to the actual context in which users watched movies (at least as informed by them), and thus, represent confident contextual signals.

Apart from using more experimental data, next steps in our research will consider analyzing additional contextual signals, and evaluating more complex contextual modeling strategies, particularly those that are able to take advantage of combinations of contextual signals.

Acknowledgements. This work was supported by the Spanish Government (TIN2011-28538-C02) and the Regional Government of Madrid (S2009TIC-1542).

References

1. Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. ACM Transactions on Information Systems 23, 103–145 (2005)
2. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 217–253 (2011)
3. Dey, A.K.: Understanding and using context. Personal and Ubiquitous Computing 5, 4–7 (2001)
4. Baltrunas, L., Ricci, F.: Experimental evaluation of context-dependent collaborative filtering using item splitting. User Modeling and User-Adapted Interaction (2013)
5. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering 17, 734–749 (2005)
6. Baltrunas, L., Amatriain, X.: Towards time-dependant recommendation based on implicit feedback. In: Proceedings of the 2009 Workshop on Context-Aware Recommender Systems (2009)
7. Baltrunas, L., Ricci, F.: Context-based splitting of item ratings in collaborative filtering. In: Proceedings of the 3rd ACM Conference on Recommender Systems, pp. 245–248 (2009)
8. Baltrunas, L., Ricci, F.: Context-dependent items generation in collaborative filtering. In: Proceedings of the 2009 Workshop on Context-Aware Recommender Systems (2009)
9. Panniello, U., Tuzhilin, A.: Experimental comparison of pre- vs. post-filtering approaches in context-aware recommender systems. In: Proceedings of the 3rd ACM Conference on Recommender Systems, pp. 265–268 (2009)
10. Oku, K., Nakajima, S., Miyazaki, J., Uemura, S.: Context-aware SVM for context-dependent information recommendation. In: Proceedings of the 7th International Conference on Mobile Data Management, pp. 109–109 (2006)
11. Karatzoglou, A., Amatriain, X., Baltrunas, L., Oliver, N.: Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In: Proceedings of the 4th ACM Conference on Recommender Systems, pp. 79–86 (2010)
12. Rendle, S., Gantner, Z., Freudenthaler, C., Schmidt-Thieme, L.: Fast context-aware recommendations with factorization machines. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information, pp. 635–644 (2011)
13. Panniello, U., Gorgoglione, M., Palmisano, C.: Comparing pre-filtering and post-filtering approach in a collaborative contextual recommender system: an application to e-commerce. In: Proceedings of the 10th International Conference on E-Commerce and Web Technologies, pp. 348–359 (2009)
14. Panniello, U., Gorgoglione, M.: Incorporating context into recommender systems: an empirical comparison of context-based approaches. Electronic Commerce Research 12, 1–30 (2012)
15. Panniello, U., Tuzhilin, A., Gorgoglione, M.: Comparing Context-Aware Recommender Systems in Terms of Accuracy and Diversity. User Modeling and User-Adapted Interaction (in press, 2013)
16. Campos, P.G., Díez, F., Cantador, I.: Time-Aware Recommender Systems: A Comprehensive Survey and Analysis of Existing Evaluation Protocols. User Modeling and User-Adapted Interaction (in press, 2013)

17. Koren, Y.: Collaborative filtering with temporal dynamics. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 447–456 (2009)
18. Woerndl, W., Groh, G.: Utilizing Physical and Social Context to Improve Recommender Systems. In: Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops, pp. 123–128 (2007)
19. He, J., Chu, W.W.: A social network-based recommender system (SNRS). In: Data Mining for Social Network Data 2010, Annals of Information Systems, vol. 12, pp. 47–74 (2010)
20. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
21. Breiman, L.: Random Forests. Machine Learning 45, 5–32 (2001)
22. Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 230–237 (1999)
23. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann Publishers Inc., San Francisco (2005)
24. Ling, C.X., Huang, J., Zhang, H.: AUC: A Better Measure than Accuracy in Comparing Learning Algorithms. In: Xiang, Y., Chaib-draa, B. (eds.) AI 2003. LNCS (LNAI), vol. 2671, pp. 329–341. Springer, Heidelberg (2003)