Time-Aware Novelty Metrics for Recommender Systems

Pablo Sánchez[1] and Alejandro Bellogín[2][0000−0001−6368−2510]

[1] Universidad Autónoma de Madrid, Madrid, Spain,
pablo.sanchezp@estudiante.uam.es
[2] Universidad Autónoma de Madrid, Madrid, Spain,
alejandro.bellogin@uam.es

Abstract Time-aware recommender systems is an active research area where the temporal dimension is considered to improve the effectiveness of the recommendations. Even though performance evaluation is dominated by accuracy-related metrics – such as precision or NDCG –, other properties of the recommended items like their novelty and diversity have attracted attention in recent years, where several metrics have been defined with this goal in mind. However, it is unclear how suitable these metrics are to measure novelty or diversity in temporal contexts. In this paper, we propose a formulation to capture the time-aware novelty (or *freshness*) of the recommendation lists, according to different time models of the items. Hence, we provide a measure to account for how much a system is promoting fresh items in its recommendations. We show that time-aware recommenders tend to provide more fresh items, although this is not always the case, depending on statistical biases and patterns inherent to the data. Our results, nonetheless, indicate that the proposed formulation can be used to extend the knowledge about what items are being suggested by any recommendation technique aiming to exploit temporal contexts.

1 Introduction

Recommender Systems (RS) have become necessary applications in a large number of companies that offer personalized content to users. The ability to not only get useful and relevant recommendations, but also to offer novel and diverse items, can increase the number of users of a system and generate more benefits for companies. However, even though most recommenders are optimized to make accurate recommendations, nowadays it is recognized that other evaluation dimensions besides accuracy should be considered to properly model the user needs and understand why she wants a recommendation [1]; serendipity, novelty, and diversity, among others, are some of the criteria that are starting to get attention in the RS community beyond accuracy metrics [2].

At the same time, recommendation techniques that consider at some point the context of the user – either social, geographical, or temporal – are becoming more and more popular. Time-aware RS, in particular, allow to consider a type of context that is very easy to capture and provides very useful information to the system, since it allows to discriminate at different levels: moment of the day, day of the week, seasonality, etc. [3]. This type of systems have gained even

more attraction in the past years since the Netflix prize, where modeling the temporal bias of users and items when rating movies was decisive to improve the performance of the algorithms [4].

However, the evaluation of RS has overlooked this important dimension, and most of the work has focused on how different evaluation methodologies (how the data partitioning should be made and which items should be considered when generating the rankings) should be applied to the recommendation data [3], leaving aside the definition of evaluation metrics specifically tailored to the problem of time-aware recommendation.

Therefore, in this paper we address the problem of formulating a time-aware evaluation metric based on the concept of novelty. To achieve this, we first extend the novelty and diversity framework presented in [5] so it can work with temporal data; we then propose different measurements to capture how novel a recommendation list is with respect to some item time model (in other terms, how *fresh* the recommended items are by considering what makes an item *new* or *old* according to a specific item time model). Finally, we validate such measurements using well-known recommendation algorithms – both time-aware and time-agnostic – on three real-world datasets.

2 Background

2.1 Recommender systems

One of the earliest and most popular collaborative filtering approaches is the neighborhood-based recommender, either user-based (UB) or item-based (IB). These approaches are normally represented as an aggregation function of the ratings from the k most similar users or items [6]. On the other hand, matrix factorization (MF) algorithms, or model-based techniques, are often the preferred methods due to their superior accuracy in some domains and datasets [7]. These models try to explain the ratings by characterizing both users and items as k latent factors derived from the original rating matrix.

As an extension of these classical algorithms, context-based RS aim to exploit additional signals such as the time of the day, the user's social connections, the user mood, or whether the user is alone or in a group [3]. Among these signals, the temporal dimension of the interaction – e.g., a rating – is the easiest to capture and exploit. Because of this, several approaches have been proposed in terms of algorithms, as in [8] where the author exploits temporal biases and incorporates them into an MF algorithm, or in [9] where a temporal decay weight is introduced in the IB formulation so recent information is deemed more important.

2.2 Novelty metrics for recommender systems

Originally, RS were evaluated using error metrics such as MAE or RMSE, largely due to the Netflix prize [4], where the objective was to decrease the RMSE error a 10% with respect to the Netflix algorithm. However, this type of evaluation method has now become outdated because it does not matches the user experience [1]. As a consequence, Information Retrieval (IR) metrics such as precision or recall are used to measure the effectiveness of the rankings generated by recommendation algorithms.

Even though these evaluation metrics are closer to the user experience, by optimizing only this type of metrics we ignore other concepts important for the interaction with the system such as the discovery of new or surprising items [10]. Because of this, different ways to define novelty and diversity have emerged in the last years. In particular, in [5] the authors generalized novelty and diversity metrics as follows:

$$m(R_u \mid \theta) = C \sum_{i_n \in R_u} \text{disc}(n) p(rel \mid i_n, u) \text{nov}(i_n \mid \theta) \ , \tag{1}$$

where R_u denotes the items recommended to user u, θ stands for a generic contextual variable – e.g., the user profile or the ranking R_u –, C is a normalizing constant, $\text{disc}(n)$ represents a discount function, the term $p(rel \mid i_n, u)$ introduces relevance in the definition of the metric, and $\text{nov}(i \mid \theta)$ is an item novelty model. For instance, by taking $\text{nov}(i \mid \theta) = 1 - p(seen \mid i)$ (the complement of the probability that the item was seen, i.e., its popularity) this formulation leads to the Expected Popularity Complement (EPC) novelty metric.

3 Time-aware novelty metrics

Our proposed time-aware novelty metrics extend the traditional novelty metrics for RS by integrating the time dimension of user-item interactions with the system. In this section, we explain how we integrate these metrics within the framework presented in [5] (Sect. 3.2) and the different time models derived for items (Sect. 3.1).

3.1 Modeling time profiles for items

In order to extend novelty metrics to consider temporal aspects of the items, we first need to represent and model their temporal dimension. From now on, we shall call these item time models as item temporal representations or item profiles.

Our goal is, for each item i, to produce an item profile $\langle t_1(i), \cdots, t_n(i) \rangle$ according to the time model t. In general, this representation will use the metadata available in the system about the item, such as its creation date, or its release time, modification time, or inclusion in the system catalog (which is not necessarily the same as the creation time, as in music databases). In fact, an item profile could include all these different times in the same representation, since an aggregation function will be later used to summarize it into a single value (see next subsection). An example of this item profiling is used in [11], where the authors exploit the release dates of music songs.

Nonetheless, when dealing with collaborative datasets, another source of information becomes available. It is possible to define the temporal profile of an item as the instants when a user interacted with it in the system. Even though the most typical interaction type in the literature are ratings, this definition is easily extended to any other interaction between users and items, such as comments, reviews, clicks, purchases, or even impressions outside of the system (such as system mentions in social networks). In this way, the item profile would represent any interaction patterns across these different information sources.

Obviously, these time profiles will model the item in a different way. Whereas the metadata-based item representations are based on some objective, static information (such as the release date), an interaction-based representation produces dynamic profiles, which will change depending on when the profiling is performed. At the same time, these latter representations are probably more subjective in the sense that they depend on how users interact with the system, but, because of this, they allow for profiles more tailored to how the community is actually using a specific item in the system.

We can draw a parallelism with how documents are treated when building temporal query profiles. According to [12], documents in IR are annotated with a timestamp corresponding to the date the document was published; this would correspond to the first group of time models presented, those based on metadata. However, we could also annotate the documents according to when they are accessed – or retrieved – as a response to a query; this document profiling strategy would correspond to the second group of time models, those based on interactions.

3.2 Measuring time-aware novelty models for items

As we introduced in Sect. 2.2, the framework presented in [5] allows to formalize and generalize many different diversity and novelty metrics under a common formulation, where item rank and relevance are seamlessly plugged in the model and, hence, considered in the final measurements.

The core idea in this framework is how to define the item novelty model $\text{nov}(i \mid \theta)$, where θ stands for a generic contextual variable. Section 2.2 shows how a novelty defined as the complement of popularity is formulated (EPC), where θ is actually ignored; moreover, when $\text{nov}(i \mid \theta) = \min_{j \in \theta} d(i, j)$ a distance-based measure – where θ represents the items recommended to user u, R_u – is modeled, similar to the intra-list diversity metric (ILD) [13].

In order to model a time-aware novelty metric, we propose to encode the time model of the items in the θ variable as follows:

$$\theta_t = \{\theta_t(i)\} = \{(i, \langle t_1(i), \cdots, t_n(i)\rangle)\} \quad . \tag{2}$$

Here, $t_j(i)$ represents the j-th component of the temporal representation or profile of item i by a specific time model t, as it was described in Sect. 3.1. We propose to compute the item novelty model based on different statistics of each item temporal profile. We leave the analysis of actual time series measures similar to what is done for queries [12]– and a study of their applicability – for future work, since, in most cases, the item profiles are very sparse.

In this context, the most basic model would account the item novelty based on the first appearance of that item in the system (**FIN**, from *First Item Novelty*), that is, $\text{nov}^F(i \mid \theta_t) \propto \min_{j \in \theta_t(i)} \theta_t(i)$; in this sense, an item is novel when looking at the system timeline (by default, starting with the first logged interaction, although this initial timestamp could be configured to have a later value) and onwards. On the other hand, we may also model the item novelty with respect to the point of view of the evaluation split, where an item is more novel if it is closer to the end of the training split, hence, closer to the test split (**LIN**, *Last Item Novelty*). Note that the LIN model does not necessarily simplify to the

Table 1. Statistics of the datasets used in the experiments.

Dataset	Users	Items	Ratings	Density	Scale	Date range
Ep (2-core)	22,556	15,196	75,533	0.022%	[1, 5]	Jan 2001 - Nov 2013
ML	138,493	26,744	20,000,263	0.540%	[0.5, 5]	Jan 1995 - Mar 2015
MT (5-core)	15,411	8,443	518,558	0.398%	[0, 10]	Feb 2013 - Apr 2017

complement of FIN, since its actual value will depend on the time model being used. Additionally, we can also model the item novelty by computing the average (**AIN**) or median (**MIN**) of the item profile.

These item novelty models cannot be integrated like that in Eq. (1), since the aforementioned framework is based on probability models, and hence, these quantities need to be, at least, normalized to produce valid values of the novelty metric. At the moment, the range of the item novelty models is the same as the range of the item time models, which, in general, return timestamps. A simple normalization scheme would divide the output of the item novelty model by the maximum possible value of a specific time model, another possibility would be applying a min-max normalization. Other more complex schemes could be used, providing different relative comparisons between the normalized values, but we found that the min-max normalization produced good-enough results. Therefore, we can formalize the item novelty model as follows:

$$\mathrm{nov}^{f,n}(i \mid \theta_t) = n(f(\theta_t(i)), \theta_t(i)) \ , \tag{3}$$

where f is one of the previous functions LIN, FIN, AIN, or MIN, and n is a normalization function, either the simple normalization function defined as $n(x, s) = x/\max s$, or the min-max normalization formulated as $n(x, s) = (x - \min s)/(\max s - \min s)$.

In summary, we propose a family of time-aware item novelty models that depend on a specific item time model, an aggregation function that summarizes the item temporal profile into a single number, and a normalization function that allows a fair comparison among novelty values for different items. Then, these item novelty models would lead to time-aware novelty metrics when integrated with item rank and relevance models within the framework presented in Eq. (1).

4 Experimental analysis

4.1 Experimental settings

Datasets. The experiments have been performed using three datasets where temporal information is realistic: MovieLens20M (ML), MovieTweetings (MT), and Epinions (Ep). The first two datasets cover the movie domain – one contains the ratings provided within the MovieLens[3] recommendation system and the other collects the IMDb[4] ratings made public by users on Twitter[5] as explained in [14]–, while the other dataset contains ratings to different types of products on Epinions[6]. In MT and Ep we have performed a so-called k-core subset, such that each of the remaining users and items have k ratings each. We used $k = 5$ for MT and $k = 2$ for Ep, where repetitions were also ignored before the k-core (removing

[3] https://movielens.org
[4] http://www.imdb.com
[5] https://twitter.com
[6] http://www.epinions.com

the older interactions). Table 1 summarizes some statistics of these datasets. It should be noted that, since MT is a dataset that is constantly updated, we selected a specific snapshot of 600K ratings (before computing the 5-core) that can be obtained in this url[7]. Moreover, the Ep dataset is made available by the authors from [15]. Finally, ML is available in the GroupLens website.

Evaluation methodology. We have performed a temporal split in each dataset where 80% of the (oldest) ratings have been included in the training set and the rest in the test set. For each user in the test set, every item in the training set is considered by the recommender except those items the user has already rated (i.e., we follow the TrainItems methodology [3]). Furthermore, we use a relevance threshold during the computation of the evaluation metrics so that those items whose ratings are higher or equal than this threshold are considered as relevant. More specifically, for Ep and ML a threshold of 5 is considered, whereas for MT this threshold is 9. The evaluation metrics reported are precision (P) and normalized discounted cumulative gain (NDCG), as implemented by the RankSys library[8], both of them at a cutoff of 5. We also report the ratio of users that receive at least one recommendation (USC, from user space coverage) [2].

Algorithms. We report two non-personalized algorithms: one that produces random recommendations ("Rnd") and another based on item popularity ("Pop"). We also report two neighborhood-based RS: a user-based approach ("UB") and an item-based approach ("IB"). A competitive matrix factorization approach [7] is also included ("HKV") in the comparison. All of these methods can be found in the RankSys framework.

We complement this pool of algorithms with one technique based on implicit information instead of explicit – ratings – ("BPR") [16] implemented in MyMediaLite [17], and two methods that exploit the temporal information in the dataset, to actually analyze whether the proposed metrics are sensitive to this behavior. The first one introduces an exponential time decay weight in a user-based nearest-neighbor algorithm ("TD"), mirroring the method proposed in [9] but for users instead of items; the second one combines Markov chains with similarity measures to capture the sequential behavior of the user ("Fossil") [15].

Furthermore, we also include two non-personalized recommenders that may evidence some biases in the dataset: one that recommends the items according to how they were included in the system (increasing item id, "IdAsc") or the other way around (decreasing item id, "IdDec"). And lastly, to have an idea of how far from the optimal performance (and which algorithms are closer to such skyline) we are, we have also implemented a recommender that directly returns the test set of each user ("SkyPerf"), and another that returns items according to their last interaction (optimizing, hence, the LIN novelty metric, named "SkyFresh").

For the sake of reproducibility, the optimal parameters for these recommenders, together with the source code of the proposed metrics, are available in the following Bitbucket repository: PabloSanchezP/TimeAwareNoveltyMetrics.

4.2 Performance results

Tables 2 to 4 show the performance results of the recommenders presented before on the three datasets described. Here we use the four variations of the time-aware

[7] SiDooms/MovieTweetings (7a1fae8d9f) [8] `http://ranksys.org`

Table 2. Performance comparison in the Epinions dataset. Best value per column is in bold, the second best value is marked with a ‡, and the third one with a †.

Algorithm	P	NDCG	USC	No relevance				Relevance			
				FIN	LIN	AIN	MIN	FIN	LIN	AIN	MIN
Rnd	0.0000	0.0001	**100.0**	0.3812	0.6391	0.4901	0.4753	0.0000	0.0000	0.0000	0.0000
IdAsc	0.0000	0.0000	100.0‡	0.2357	0.5083	0.3599	0.3401	0.0000	0.0000	0.0000	0.0000
IdDec	0.0000	0.0001	100.0†	0.3851	0.5790	0.4766	0.4728	0.0000	0.0000	0.0000	0.0000
Pop	0.0009‡	0.0012†	100.0	0.0788	0.7936	0.2670	0.2152	0.0003	0.0009‡	0.0006‡	0.0005‡
IB	0.0002	0.0005	49.7	0.4567†	0.6705	0.5505	0.5411	0.0001	0.0001	0.0001	0.0001
UB	0.0004	0.0007	49.7	0.3325	0.7625	0.4871	0.4601	0.0001	0.0004	0.0003	0.0003
TD	0.0004	0.0008	49.7	0.6000‡	0.9150‡	**0.7365**	**0.7238**	0.0003†	0.0004	0.0003	0.0003
HKV	0.0006	0.0018‡	50.6	0.2445	0.8808†	0.4366	0.3977	0.0002	0.0006	0.0004	0.0004
BPR	0.0007†	0.0011	50.6	0.1964	0.7917	0.3705	0.3362	0.0004‡	0.0007†	0.0005†	0.0005†
Fossil	0.0002	0.0004	31.1	0.2821	0.7806	0.4527	0.4200	0.0001	0.0001	0.0001	0.0001
SkyPerf	**0.1337**	**0.4441**	66.5	**0.6170**	0.8695	0.7286‡	0.7197‡	**0.2397**	**0.3416**	**0.2845**	**0.2807**
SkyFresh	0.0000	0.0000	100.0	0.4557	**0.9999**	0.6588†	0.5976†	0.0000	0.0000	0.0000	0.0000

Table 3. Performance comparison in the MovieLens dataset.

Algorithm	P	NDCG	USC	No relevance				Relevance			
				FIN	LIN	AIN	MIN	FIN	LIN	AIN	MIN
Rnd	0.0009	0.0010	**100.0**	0.5573	0.9834	0.6993	0.6711	0.0004	0.0009	0.0006	0.0006
IdAsc	0.0099	0.0162	100.0‡	0.0716	0.9991	0.3550	0.2437	0.0007	0.0099	0.0044	0.0038
IdDec	0.0000	0.0000	100.0†	**0.9995**	0.9995	**0.9995**	**0.9995**	0.0000	0.0000	0.0000	0.0000
Pop	0.1027‡	0.1110‡	100.0	0.0781	0.9999†	0.4361	0.3772	0.0080	0.1027‡	0.0455‡	0.0397‡
IB	0.0347	0.0414	17.8	0.3111	0.9998	0.6145	0.5878	0.0107	0.0347	0.0218	0.0212
UB	0.0498†	0.0618†	17.8	0.2431	0.9999	0.5835	0.5594	0.0117	0.0498†	0.0289	0.0277
TD	0.0420	0.0520	17.8	0.6108‡	0.9999‡	0.7838‡	0.7710‡	0.0258‡	0.0420	0.0331†	0.0327†
HKV	0.0498	0.0611	17.8	0.3068	0.9998	0.6122	0.5885	0.0143†	0.0498	0.0303	0.0292
BPR	0.0443	0.0532	17.8	0.3274	0.9998	0.6202	0.5958	0.0124	0.0443	0.0267	0.0256
Fossil	0.0297	0.0348	17.8	0.3085	0.9999	0.6116	0.5816	0.0086	0.0297	0.0181	0.0174
SkyPerf	**0.7094**	**0.8396**	99.7	0.6069†	0.9993	0.7764†	0.7618†	**0.4359**	**0.7116**	**0.5565**	**0.5472**
SkyFresh	0.0027	0.0027	100.0	0.4999	**1.0000**	0.7236	0.7026	0.0016	0.0027	0.0021	0.0020

novelty metric defined in Sect. 3.2 with and without a relevance component; moreover, the discount component (disc(n) in Eq. (1)) is ignored to simplify the results. Furthermore, in these results we use interaction-based item profiles – based on ratings – since they can be applied to all the datasets we are testing; metadata-based item profiles are presented and discussed later.

The first thing we notice in our results are the low values of accuracy-related metrics (P and NDCG). This is a well-known result in the RS literature, and it is mostly due to a high sparsity in the data, especially in the groundtruth [18]. Nonetheless, we should note that the evaluation methodology used in this paper is even more restrictive in this aspect, because the split is temporal, and, thus, there could exist users and items that appear only on training or on test. This has a strong impact on coverage (USC), especially for the CF algorithms.

From these results, however, we can infer that there is a strong popularity bias in two of the datasets (Ep and ML), since Pop is among the best recommendation techniques in terms of accuracy metrics. This is attributed to the temporal dynamics of the dataset, as evidenced in Fig. 1, where the most popular items continue increasing their number of interactions after the training-test splitting point, hence, they are relevant for many users in test.

Regarding the results about our proposed novelty metrics, we shall focus on those without the relevance component, since those values reflect more closely the formulations presented in Sect. 3, whereas those including relevance are distorted

Table 4. Performance comparison in the MovieTweetings dataset.

Algorithm	P	NDCG	USC	No relevance				Relevance			
				FIN	LIN	AIN	MIN	FIN	LIN	AIN	MIN
Rnd	0.0002	0.0003	**100.0**	0.1693	0.8473	0.4435	0.4086	0.0001	0.0002	0.0002	0.0002
IdAsc	0.0004	0.0003	100.0‡	0.1729	0.8873	0.5485	0.5938	0.0000	0.0004	0.0002	0.0002
IdDec	0.0005	0.0004	100.0†	**0.9628**	0.9800	**0.9688**	**0.9669**	0.0005	0.0005	0.0005	0.0005
Pop	0.0028	0.0023	100.0	0.1499	0.9921	0.2534	0.2074	0.0007	0.0028	0.0009	0.0008
IB	0.0082	0.0093	78.5	0.4753	0.9848	0.6002	0.5744	0.0055	0.0081	0.0065	0.0064
UB	0.0104	0.0120	78.5	0.4902	0.9951	0.5937	0.5657	0.0075	0.0104	0.0084	0.0082
TD	0.0264‡	0.0337‡	78.5	0.8487‡	0.9988‡	0.9298‡	0.9282‡	0.0239‡	0.0264‡	0.0253‡	0.0253‡
HKV	0.0150†	0.0190†	78.5	0.4131	0.9939	0.5935	0.5621	0.0093	0.0149†	0.0115†	0.0113†
BPR	0.0100	0.0121	78.5	0.4524	0.9895	0.5984	0.5690	0.0072	0.0100	0.0083	0.0082
Fossil	0.0129	0.0154	75.2	0.5186	0.9951†	0.6294	0.6051	0.0102†	0.0129	0.0111	0.0110
SkyPerf	**0.3468**	**0.5374**	81.6	0.4262	0.9686	0.6514	0.6289	**0.1528**	**0.4485**	**0.2775**	**0.2657**
SkyFresh	0.0037	0.0041	100.0	0.6715†	**1.0000**	0.8072†	0.7924†	0.0033	0.0037	0.0035	0.0035

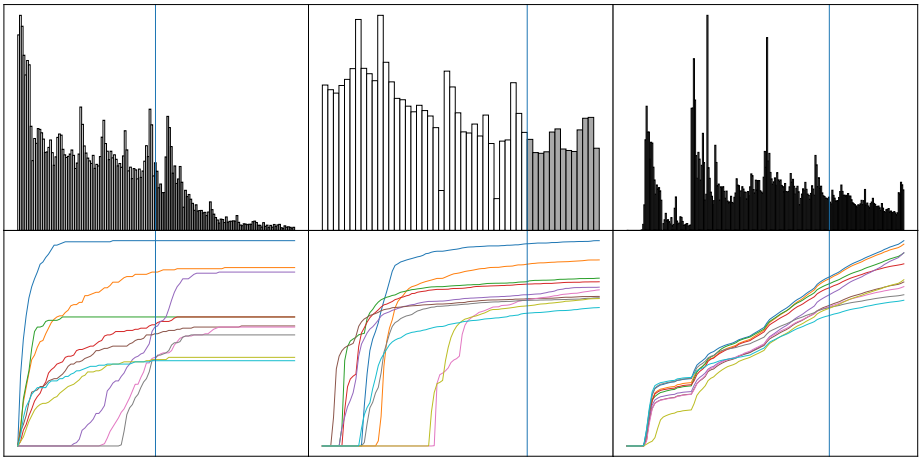

Figure 1. Comparison of rating distributions for Ep (left), MT (center), and ML (right). Top row shows the amount of ratings created throughout time, bottom row shows the cumulative rating distribution of the top-10 most popular items. The vertical line indicates the training-test splitting point.

by the actual accuracy obtained by the algorithms. We first note that the two skyline algorithms (SkyPerf and SkyFresh) obtain the performance we expect: whereas SkyPerf is optimal in terms of accuracy and relevant novelty, SkyFresh is optimal in terms of the LIN metric. Sometimes, this method also obtains good results for other metrics, mostly because the novelty metrics are not independent from each other. In this context, it might be surprising that SkyPerf does not obtain perfect values for every accuracy metric; this is because we are computing cutoff versions of the metrics, which together with the fact that we are using a relevance threshold, may leave some users without any relevant item in its pool; hence, in those cases, the metric would average 0's (because of those "empty" users in test) with other values up to 1. In any case, the reported results are the maximum achievable results, which helps us on finding the gap between observed and maximum accuracy.

We also observe that Rnd usually achieves high values of the novelty metrics (without relevance). In fact, this type of algorithm is, to some extent, showing

the *a priori* probability of retrieving a novel item according to FIN, LIN, and so on. Because these values depend on the dataset and its inherent distributions (see Fig. 1), it makes sense the metrics return very different values for this technique. This result also evidences that such a simple technique should always be reported to properly assess the effect of inner biases in the data.

Related to the data biases, IdAsc and IdDec algorithms show that such simple techniques may produce large improvements in terms of novelty, but not due to an algorithmic rationale but because of how the data is built. In this way, we observe that in ML and MT, IdDec achieves very high values of our novelty metrics. This is because the ids of the items are provided in these datasets (this is not the case for Ep) and they were created sequentially, hence, by knowing that an id has a larger id than another, we know that the former is newer (more *temporally novel* or fresh) than the latter. Even though such an algorithm is not technically interesting, this issue may have a profound impact on how ties are broken when producing a ranking: depending on the dataset, if the tie-breaking strategy sorts the items in an ascending order, older items will be presented before in the ranking.

Now, if we analyze the freshness of personalized recommendation algorithms and, specifically, the time-aware ones (i.e., TD and Fossil), we find the following. First, the Fossil technique does not achieve better results than other techniques unaware of the temporal dimension; for instance, in Ep it is worse than UB but better than BPR, whereas in ML it is the second worst among the personalized algorithms, only after UB. Nonetheless, it should be noted that Fossil, even though it is aware of the temporal dimension, it does not penalize recommending old items as long as they fit in the sequence predicted for the user. On the other hand, TD consistently returns more novel items than most of the algorithms, especially than UB, on which it is based. It is interesting to note that this behavior is in agreement with the results of the four proposed item novelty models.

These tables only show the min-max normalization due to space constraints. We present these results because this normalization has better properties than the simple normalization mentioned in Sect. 3.2. In particular, we have observed it allows for more fine-grained discrimination of the reported values; for instance, LIN in EP for UB and HKV is, as seen in Table 2, 0.7625 and 0.8808, whereas when the simple normalization technique is applied this metric produces values as close as 0.9595 and 0.9797. This effect is more noticeable in some recommenders – e.g., Pop – when its recommended items present their initial interactions located near the beginning of the system timeline, which would produce a very small value with the min-max normalization, however, this value is still very large with the simple normalization; in fact, the obtained values in this situation (again, for Ep) with FIN are 0.0788 vs. 0.8430, respectively.

In summary, based on these results we conclude that the proposed metrics allow to discover whether some recommenders promote more temporal novel items than others; however, depending on the item novelty model some differences arise. For example, LIN always produces very high values, which does not help to discriminate between the recommenders. This might be also attributed to the popularity bias observed in Fig. 1, since this means that it is very easy for almost any algorithm to return novel items interacted recently, due to the observed bursting events near the splitting point, which would create – somewhat

Table 5. Performance comparison in the MovieLens dataset (left) and MovieTweetings (right) when using metadata-based time profiles.

Algorithm	No relevance		Relevance	
	Y-*IN	R-FIN	Y-*IN	R-FIN
Rnd	0.7707	0.5573	0.0008	0.0004
IdAsc	0.8387	0.0716	0.0083	0.0007
IdDec	0.7581	**0.9995**	0.0000	0.0000
Pop	0.8227	0.0781	0.0848‡	0.0080
IB	0.8242	0.3111	0.0282	0.0107
UB	0.8164	0.2431	0.0405†	0.0117
TD	**0.8822**	0.6108‡	0.0372	0.0258‡
HKV	0.8102	0.3068	0.0397	0.0143†
BPR	0.8354	0.3274	0.0364	0.0124
Fossil	0.8445†	0.3085	0.0251	0.0086
SkyPerf	0.8602‡	0.6069†	**0.6126**	**0.4359**
SkyFresh	0.6305	0.4999	0.0018	0.0016

Algorithm	No relevance		Relevance	
	Y-*IN	R-FIN	Y-*IN	R-FIN
Rnd	0.8764	0.1693	0.0002	0.0001
IdAsc	0.2264	0.1729	0.0001	0.0000
IdDec	**0.9907**	**0.9628**	0.0005	0.0005
Pop	0.9693	0.1499	0.0027	0.0007
IB	0.9629	0.4753	0.0079	0.0055
UB	0.9745†	0.4902	0.0102	0.0075
TD	0.9817‡	0.8487‡	0.0260‡	0.0239‡
HKV	0.9494	0.4131	0.0141†	0.0093
BPR	0.9621	0.4524	0.0097	0.0072
Fossil	0.9741	0.5186	0.0127	0.0102†
SkyPerf	0.9184	0.4262	**0.4122**	**0.1528**
SkyFresh	0.9689	0.6715†	0.0036	0.0033

artificially – very popular fresh items. Regarding the FIN model, we believe it might not be very useful in datasets where several items appear at the very beginning, since it would not discriminate those cases, just like the LIN model. On the other hand, the two models that aggregate all the interactions received by an item (AIN and MIN) are more robust to these situations, and, in particular, the one based on the median (MIN) is expected to be more robust to outliers.

Finally, since the previous analysis was only done using the interaction-based item profiles, let us analyze now the metadata-based item profiles. Table 5 shows a comparison between a metadata-based item profile where the release date of the product (in our case, movies from ML and MT, because this information is not available in Ep) is used as the time model. Since the release date (years) of an item is unique, any of the item novelty models would produce the same result, that is why we denote it in the table as Y-*IN. Additionally, we compare these results against the FIN model using ratings as interactions (R-FIN), because this model is conceptually the closest one to Y-*IN: a time representation using the release date would indicate the first possible interaction with such item.

Based on these results, we observe a different situation depending on the dataset. Indeed, whereas in ML the recommender that returns more novel items regarding their release date is TD, in MT is IdDec, an almost identical situation to what we found for the interaction-based profiles. The main rationale behind this is, as noted by the authors of the MT dataset in [14], this dataset is more dynamic and mostly contains new movies; this may explain why looking at the id provides both information about the interaction age but also about the item age in terms of its release date. It should be noted that, in contrast to the interaction-based profile models, we have not evaluated a recommendation technique that exploits the release date of the items, so there was no expected ranking of methods according to Y-*IN. In any case, it is interesting to observe that TD – a time-aware recommender including a time-decay factor – achieves very good results when item profiles are based on metadata, exactly what it showed for interaction-based time models.

5 Related work

The concept of freshness has been widely used in IR, since most search engines aim at returning relevant but fresh (novel) documents. The authors in [19] propose

a scoring method that combines freshness and relevance, and a similar idea is explored but using latent factors and learning-to-rank approaches in [20] based on click logs. This concept has received more attention in the field of Music IR [21], specifically to retrieve those songs a user may have forgotten or new releases from liked artists. In [22], freshness is defined as the strength of strangeness (of a specific song to a user) by applying the Forgotten Curve, which is modeled as a (negative) exponential function.

In recommendation, freshness or temporal novelty has usually referred to the capacity of the system to return new recommendations each time the user interacts with it [23]. In this way, in [24] the authors define temporal diversity and novelty metrics based on the overlap between previous recommendation lists. It is worth mentioning that, as described in [5], these temporal diversity and novelty metrics can be instantiated within the framework presented in Sect. 2.2, and, hence, they also fit the model presented herein. Finally, the only formulation we have found that is similar to one of the proposed time-aware novelty models was presented in [11] in the context of music recommendation. In that work, the authors define freshness as the average of the release date of the recommended songs, which is equivalent to our AIN model using a metadata-based profile. However, in that paper no time-aware RS were analyzed, so no relationships between the metric and the learning algorithm could be derived, as we have presented here.

6 Conclusion and future work

The temporal dimension has been mostly neglected when measuring novelty and diversity in RS, only few works mention it but in a context where the user receives repeating recommendations from the system. In this paper, we have introduced this important dimension in the definition of a family of novelty models that allows us to measure if a recommendation technique is prone to return fresh items or not. Our results show that, while the proposed metrics work as expected, they also open the possibility to be affected by some biases in the data that are not necessarily considered when measuring accuracy-based metrics – such as temporal bursts of activity and relationships between item age and their ids.

Our approach could open up new possibilities towards producing more time-aware novel recommendations, for instance, by reranking optimized rankings for accuracy based on the proposed item freshness models, as it is done for general diversity and novelty measures [10, 24]. Furthermore, the proposed time-aware item novelty models and representations could be applied to streaming RS, where the temporal model of the recommender is not necessarily the same as the one for the metric: while the metric could be recomputed every day, the recommender could be trained every week, for example. In this way, a more fine-grained analysis may be derived so, for instance, the optimal period to train the recommender can be calculated, or a day-by-day sensitivity could be explored for different recommenders. It is worth mentioning that, although this is also possible to achieve with offline data, the conclusions will not be as significant because most of the datasets are very sparse, hence, it is necessary to use real, online data.

Acknowledgments. This research was supported by the Spanish Ministry of Economy, Industry and Competitiveness (TIN2016-80630-P).

References

1. McNee, S.M., Riedl, J., Konstan, J.A.: Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: CHI, ACM (2006) 1097–1101
2. Gunawardana, A., Shani, G.: Evaluating recommender systems. In: Recommender Systems Handbook. Springer (2015) 265–308
3. Campos, P.G., Díez, F., Cantador, I.: Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. UMUAI **24**(1-2) (2014) 67–119
4. Bell, R.M., Koren, Y.: Lessons from the netflix prize challenge. SIGKDD Explorations **9**(2) (2007) 75–79
5. Vargas, S., Castells, P.: Rank and relevance in novelty and diversity metrics for recommender systems. In: RecSys, ACM (2011) 109–116
6. Ning, X., Desrosiers, C., Karypis, G.: A comprehensive survey of neighborhood-based recommendation methods. In: Recommender Systems Handbook. Springer (2015) 37–76
7. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: ICDM, IEEE Computer Society (2008) 263–272
8. Koren, Y.: Collaborative filtering with temporal dynamics. CACM **53**(4) (2010) 89–97
9. Ding, Y., Li, X.: Time weight collaborative filtering. In: CIKM, ACM (2005) 485–492
10. Castells, P., Hurley, N.J., Vargas, S.: Novelty and diversity in recommender systems. In: Recommender Systems Handbook. Springer (2015) 881–918
11. Chou, S., Yang, Y., Lin, Y.: Evaluating music recommendation in a real-world setting: On data splitting and evaluation metrics. In: ICME, IEEE Computer Society (2015) 1–6
12. Jones, R., Diaz, F.: Temporal profiles of queries. ACM TOIS **25**(3) (2007) 14
13. Ziegler, C., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: WWW, ACM (2005) 22–32
14. Dooms, S., Bellogín, A., Pessemier, T.D., Martens, L.: A framework for dataset benchmarking and its application to a new movie rating dataset. ACM TIST **7**(3) (2016) 41:1–41:28
15. He, R., McAuley, J.: Fusing similarity models with markov chains for sparse sequential recommendation. In: ICDM, IEEE (2016) 191–200
16. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: bayesian personalized ranking from implicit feedback. In: UAI, AUAI Press (2009) 452–461
17. Gantner, Z., Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Mymedialite: a free recommender system library. In: RecSys, ACM (2011) 305–308
18. Bellogín, A., Castells, P., Cantador, I.: Statistical biases in information retrieval metrics for recommender systems. Inf. Retr. Journal **20**(6) (2017) 606–634
19. Sato, N., Uehara, M., Sakai, Y.: A case study on freshness based scoring for fresh information retrieval. In: ISCIT. Volume 1. (2004) 210–215 vol.1
20. Wang, H., Dong, A., Li, L., Chang, Y., Gabrilovich, E.: Joint relevance and freshness learning from clickthroughs for news search. In: WWW, ACM (2012) 579–588
21. Knees, P., Schedl, M.: Music Similarity and Retrieval - An Introduction to Audio- and Web-based Strategies. The Information Retrieval Series. Springer (2016)
22. Hu, Y., Ogihara, M.: Nextone player: A music recommendation system based on user behavior. In: ISMIR, University of Miami (2011) 103–108
23. McNee, S.M., Riedl, J., Konstan, J.A.: Making recommendations better: an analytic model for human-recommender interaction. In: CHI, ACM (2006) 1103–1108
24. Lathia, N., Hailes, S., Capra, L., Amatriain, X.: Temporal diversity in recommender systems. In: SIGIR, ACM (2010) 210–217