Good Times Bad Times: A Study on Recency Effects in Collaborative Filtering for Social Tagging

Santiago Larrain
Pontificia Universidad Católica de Chile
Santiago, Chile
slarrain@uc.cl

Christoph Trattner
NTNU & Know-Center
Trondheim Norway & Graz, Austria
ctrattner@know-center.at

Denis Parra
Pontificia Universidad Católica de Chile
Santiago, Chile
dparra@ing.puc.cl

Eduardo Graells-Garrido
Telefonica
Santiago, Chile
eduardo.graells@gmail.com

Kjetil Nørvåg
NTNU
Trondheim, Norway
noervaag@idi.ntnu.no

ABSTRACT

In this paper, we present work-in-progress of a recently started project that aims at studying the effect of time in recommender systems in the context of social tagging, consolidating several approaches presented rather scattered in past related work. The paper presents results of a study where we focused on understanding (i) "when" to use the temporal information into traditional collaborative filtering (*CF*) algorithms, and (ii) "how" to weight the similarity between users and items by exploring the effect of different time-decay functions. As the results of our extensive evaluation conducted over five social tagging systems (Delicious, BibSonomy, CiteULike, MovieLens, and Last.fm) suggest, the step (*when*) in which time is incorporated in the *CF* algorithm has substantial effect on accuracy, and the type of time-decay function (*how*) plays a role on accuracy and coverage mostly under pre-filtering on user-based *CF*, while item-based shows stronger stability over the experimental conditions.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*

Keywords

time-aware recommendations; collaborative filtering; social tagging.

1. INTRODUCTION

Time-Aware recommender systems have been extensively studied in the past and have proven to be more effective than traditional un-contextualized recommender systems [1]. Although there is a huge body of research in this context, only a few studies have investigated to what extent time can help to improve implicit feedback recommender systems relying on data such as social tags [19, 5, 10]. Also there seems to be a gap in the literature on *how* and *when* to incorporate the variable of time in the recommendation process, i.e., which type of decay functions to choose and where to incorporate the variable of time, before or after the filtering step in a Collaborative Filtering (hereinafter *CF*) setting.

Problem Statement: We try to find the best way to incorporate temporal information in a *CF* item recommender for social tagging systems. Our notion of temporal information is related to *recency* considering this argument: if users u and v have bookmarked items using the same tag t they might have some degree of similarity, but if u have used the tag recently and v used it long ago, their similarity might have weaken as a consequence of concept drift [7]. Based on this intuition, we address the following challenges (i) "when" to use the temporal information into traditional *CF* algorithms for time-aware item predictions, since these algorithms consist of two steps where similarity calculations play a role, and (ii) "how" to weight the similarities between users and items exploring the effect of different time-decay functions .

Research Questions: Our study was henceforth driven by the following two research questions:

- **RQ1.** Being aware of different ways of incorporating the variable of time in a *CF* time-aware item recommender system, formally known as pre- or post-filtering, what is the most efficient approach?
- **RQ2.** Being aware of different types of decay functions to model recency effects in recommender systems, which of those functions provides the best approximation in the context of item recommendation in social tagging systems?

Results: Our results indicate that the step (pre- or post-filtering) in which temporal information is incorporated in the recommendation algorithm has a substantial effect on the performance of the algorithms. On the other side, the decay function affects in different ways the version of the *CF* method: item-based is very stable while user-based is affected by the decay function. In addition, combining both user- and item-based recommendation has a small but regular improvement on ranking accuracy.

2. DATASET AND APPROACHES

Since users' preferences drift over time [7], some works on tag-based recommenders have investigated how to incorporate temporal information into *CF* approaches [5, 10, 19]. These works consider that user-similarity is conditioned not only on the common

RecSys '15, September 16 – 20, 2015, Vienna, Austria.
ACM 978-1-4503-3692-5/15/09.
http://dx.doi.org/10.1145/2792838.2799682 .

Dataset	|B|	|U|	|R|	|T|	|TAS|
BIB	82,539	2,437	28,000	30,919	339,337
CUL	10,533	7,182	42,320	46,060	373,271
ML	53,607	3,983	3,983	14,883	92,387
DEL	379,667	8,377	82,557	44,280	1,205,018
LFM	66,353	1,798	8,190	8,378	172,051

Table 1: Descriptive statistics of our datasets. B: Bookmarks, U: users, T: Tags, TAS: Tag Assignments.

Decay	Param.	BIB	CUL	ML	DEL	LFM
–	$T_0 = median(\Delta t)$	417	482	92	184	273
Exp.	$\lambda = \frac{\beta}{T_0}$	.0016	.0014	.0075	.0037	.0025
Power	$\lambda = \frac{\beta}{ln(T_0)}$	.1148	.1121	.1532	.1329	.1235
Linear	$\lambda = \beta * T_0$	834	964	184	368	546
Log.	$\lambda = \frac{\beta}{T_0}$	.0026	0022	.0119	.0059	.0040

Table 2: Decay function parameters for each data set. The parameter β is optimized according to each dataset and decay function.

items they have consumed, but how recently both users have consumed them [14]. For instance, Zheng et al. [19] improved the performance of a tag-based recommender after filtering their data based on recency of tagging interactions by applying a power decay function. Likewise, Huang et al. [5] improved a tag-based recommender relying on a two-step filtering process which modelled the recency effect of interactions as a linear decay function. More recently, Lacic et al. [11] outperformed state-of-the-art tag-based recommendation methods by decaying the similarity of users and items considering the recency of their tagging actions using the Base-Level Learning function [10]. Although these results are certainly promising, there is no consensus in current literature about which is the best approach to incorporate the variable of time.

2.1 Datasets

For reproducibility, we focused on five well-known and freely-available folksonomy datasets in our experiments. In particular, we used datasets of the social bookmark and publication sharing system BibSonomy[1] (BIB), the reference management system CiteULike[2] (CUL), the movie recommendation Website MovieLens[3] (ML), the social bookmarking system Delicious[4] (DEL) and the online music platform LastFM[5] (LFM). As suggested by related work in the field (e.g., [6]), we excluded all automatically imported and generated tags. In the case of CiteULike we randomly selected 10% of the user profiles and in the case of Delicious 2% for reasons of computational effort (see also [3]). The final dataset statistics can be found in Table 1.

2.2 Pre- and Post-Filtering

First, we briefly introduce the traditional versions of CF: *user-based* [15] and *item-based* [16]. User-based CF is made of two steps: (i) given a center user u, find a neighborhood N of the k most similar users (K-nearest neighbors or K-NN) to u by a similarity function $sim(u,v), u \in U \wedge v \in U \wedge u \neq v$, and then (ii) given the items consumed by users in N, which have not yet been consumed by u, recommend items after ranking them by a score function that predicts the value $\hat{v}_{u,i}$ that user u will give to item i:

$$\hat{v}_{u,i} = \bar{v}_u + \alpha \sum_{n \in N} sim(u,n)(v_{n,i} - \bar{v}_n), \quad (1)$$

where $\bar{v}_u$ is the average value that user u has given to items in the dataset, $sim(u,n)$ is a similarity function (usually Cosine or Jaccard), $v_{n,i}$ is the value that user n gave to item i, $\bar{v}_n$ is the average value that user n has given to items in the dataset, and α is a normalization constant. Traditionally, $v_{x,y}$ represents an ordinal rating, e.g. 1-5, but in our model it represents a binary value (user bookmarked the item or not). Then, Eq. 1 is adapted as shown in [19] to:

$$\hat{v}_{u,i} = \frac{\sum_{n \in N} sim(u,n) \times v_{n,i}}{\sum_{n \in N} sim(u,n)} \quad (2)$$

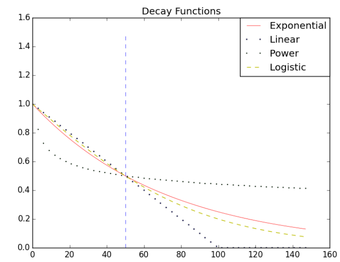

Figure 1: The figure compares the different decay functions at the step of parameter fitting. The x-axis represents temporal data (days since a bookmark was set) and the vertical blue dashed line is a median set on x=50 for representation purposes. BLL is not shown because its parameters are set based on literature.

Since we are interested in studying the recency effect by applying time-decay functions during the K-NN step or at the moment of picking the relevant items, we call our approaches *pre-filtering* and *post-filtering*. In order to implement a *pre-filtering* approach we update each user vector with the decay function, $f(\Delta t_{u,i})$, where $\Delta t_{u,i}$ is the amount of time between a reference time t_{u_0} and the moment where user u interacted with item i. If our similarity function is cosine-based, we change it to:

$$cos'(u,n) = \frac{\sum_{i \in I_{u,n}} f(\Delta t_{u,i}) v_{u,i} f(\Delta t_{n,i}) v_{n,i}}{\sqrt{\sum_{i \in I_{u,n}} (f(\Delta t_{u,i}) v_{u,i})^2 \sum_{i \in I_{u,n}} (f(\Delta t_{n,i}) v_{n,i})^2}}, \quad (3)$$

where $I_{u,n}$ is the set of common items that both users u and n have consumed, and $v_{u,i}$ is the value given by user u to item i. Similar to [2] we apply *post-filtering* as a decay factor $f(\Delta t_{n,i})$ when doing the recommendation on the CF:

$$\hat{v}_{u,i} = \alpha \sum_{n \in N} f(\Delta t_{n,i}) cos'(u,n) v_{n,i}, \quad (4)$$

where α is a normalizing constant, similar to Eq. 1. In the case of item-based collaborative filtering, the method has two steps: (i) create a user model $items(u)$ as a vector of items that user u has consumed with the respective values or ratings that u has given to those items, and then (ii) using a similarity function between items $sim(i,j)$, select the items which are most similar to those in $items(u)$. In this case, *pre-filtering* is implemented by weighting the rating or value of each item in $items(u)$ with the decay function $f(\Delta t_{u,i})$, and *post-filtering* is accomplished by decaying the similarity between two items in a similar fashion to Eq. 3, but considering pairs of items rather users.

2.3 Decay Functions

We have identified five approaches that make use of decay functions to model the recency of user interactions. These approaches

[1] http://www.kde.cs.uni-kassel.de/bibsonomy/dumps
[2] http://www.citeulike.org/faq/data.adp
[3] http://files.grouplens.org/datasets/movielens/
[4] https://www.uni-koblenz.de/FB4/Institutes/IFI/AGStaab/Research/DataSets/PINTSExperimentsDataSets/
[5] http://grouplens.org/datasets/hetrec-2011/

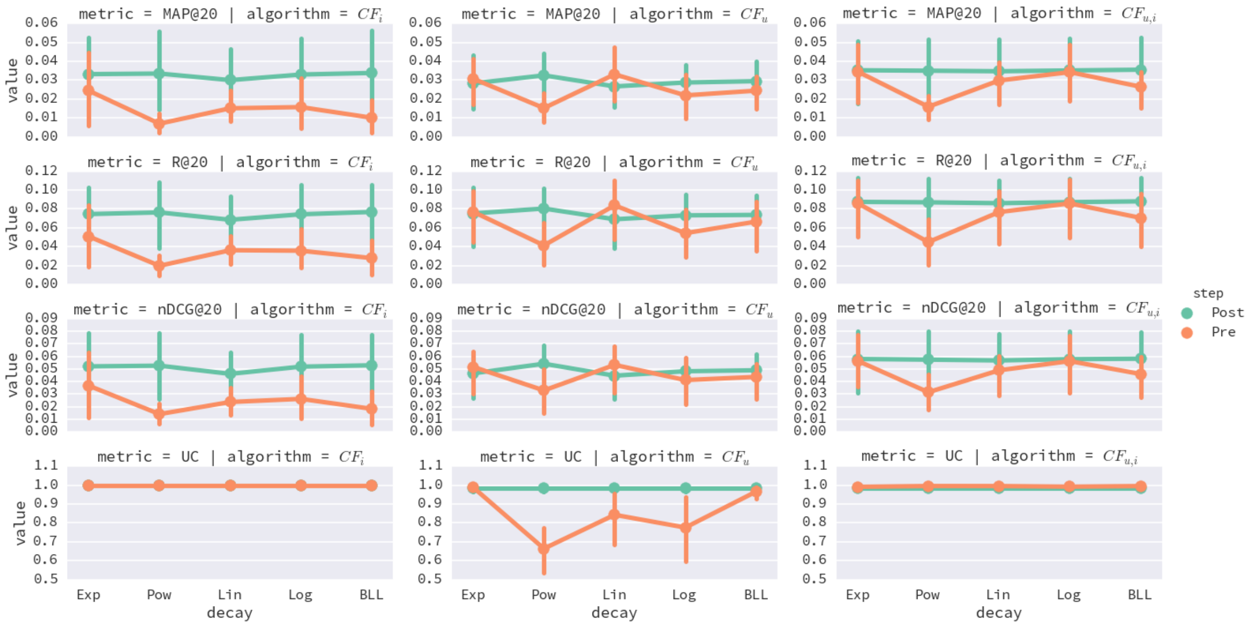

Figure 2: The figure summarises the results averaged over the five datasets in a 4 x 3 plot matrix. Rows are metrics (MAP@20, R@20, nDCG@20 and User Coverage), columns represent the methods (CF_i, CF_u, $CF_{u,i}$) and the x-axis of each plot shows the 5 decay-functions studied (Exponential, Power, Linear, Logistic, and BLL).

decay the similarity between a pair of users if they have interacted with the same item at different times, what we denote in the following functions as $f(\Delta t)$.

Exponential: This type of decay function has been used in several works (e.g., [2, 13]), $f(\Delta t) = e^{-\lambda \times \Delta t}$.

Power: The power decay function was used by Wu et al. [18] in a user-based CF approach for a social tagging system in a digital library, $f(\Delta t) = (\Delta t)^{-\lambda}$.

Linear: A linear decay function was used by Lee and Park [12], where they use the time variable in a user-based CF approach before calculating the user similarity, $f(\Delta t) = max(0, 1 - \frac{\Delta t}{\lambda})$.

Logistic: Suggested by Ding and Li [2] as alternative to exponential decay but disregarded without empirical support since it would be less representative of users' latest preferences. We test it in our experiments as $f(\Delta t) = \frac{2}{1+e^{\lambda \times \Delta t}}$.

BLL: Finally, we employed the Base-Level learning function that was introduced by Kowald et al. [10], $f(\Delta t) = ln(\sum_{i=1}^{n} \Delta t_i^{-\lambda})$.

Parameters: Ding and Li [2] achieved the best performance by applying a half-life decay, i.e., setting a parameter T_0 in such a way that $f(\Delta t)$ reduces by $1/2$ in T_0 days. The effect of this parameter fitting in the different distributions can be seen in Figure 1. The relation between T_0 and and λ, the parameter we set in the aforementioned decay functions, follows the procedure in [2] and Table 2 shows the relation and values for each dataset and decay function in our experiments. To set T_0, we analyzed the distribution of Δt at each dataset and we tested setting T_0 at their mean and median, obtaining better results with $T_0 = median(\Delta t)$. The parameter β in Table 2 was optimized after performing a 5-fold cross-validation on each dataset. The only exception was taken on the BLL approach, where we set $\lambda = .5$ based Kowald et al.'s recommendation [10].

3. EXPERIMENTAL SETUP

Evaluation Methodology: We used a training and test-set split method as proposed by popular and related work in this area [19, 17]. Hence, for each user we sorted her bookmarks in chronological order and used the 20% most recent bookmarks for testing and the rest for training [1]. To quantify the performance of each of our recommender methods, we used a diverse set of well-established metrics in recommender systems. In particular, we report Normalized Discounted Cumulative Gain (nDCG@20), Mean Average Precision (MAP@20), Recall (R@20) and User Coverage (UC) [4, 9].

Recommendation Methods: The approaches we utilized were User- (CF_u) [15], Item- (CF_i) [16] and User-Item-based ($CF_{u,i}$) collaborative filtering (without time as introduced before). The size of the neighborhood for the User-based KNN calculations was K=20, based on the results of [10].

4. RESULTS

RQ1. Pre- and post-filtering. Among all results, this one had the clearest effect, but the size of the effect depends on the method used, as seen in Figure 2. We observe that the largest performance difference between post- and pre-filtering is seen when using the item-based *CF* method, where the only decay function that results in comparable performance for the pre-filtering is the exponential decay. For the other decay functions, there is a large gap in favor of post-filtering. Now, under the user-based *CF* the pre-filtering works better than post-filtering under linear decay, but this improvement in performance is obtained at the cost of user coverage, dropping to barely over 80%. We can also observe the stability of the post-filtering method independent of the decay function. Finally when combining user- and item-based *CF* we see an improvement of pre-filtering with respect to the logistic decay, but the power decay combined with pre-filtering still results in poor performance over MAP@20, R@20 and nDCG@20, though the user coverage problem is alleviated.

RQ2. Decay functions. The recency effect was investigated through different decay functions. If we consider post-filtering, there are no significant differences among decay functions in average, but the power decay function $f(\Delta t) = (\Delta t)^{-b}$ was the one resulting in the top accuracy results as seen in the bottom row of plots in Figure 3, and also in the table of the appendix. On the other hand, the power-decay was the one performing the worst un-

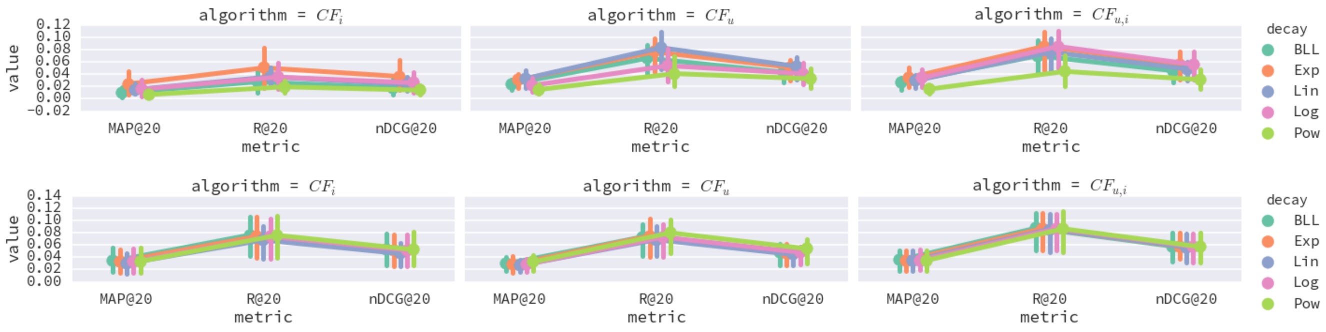

Figure 3: The effect (means over all datasets) of pre- (top row) and post-filtering (bottom row) with different decay functions and algorithms on the MAP@20, R@20 and nDCG@20 performance metric.

der pre-filtering, as seen in the top row of Figure 3. Apart from that, our results don't indicate a significant effect of different decay functions in the performance of tag-based item recommenders.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have shown that it is worthwhile to study the effect of time in recommender systems. We focused in our analysis on the effect on the weighting step and five different decay functions proposed in the literature. As our experimental results conducted over five different social tagging datasets suggest, there is a strong effect on the weighting procedure (pre- and post-filtering) as well on the decay function used but mostly when using pre-filtering, challenging the results obtained by previous published work in the area (e.g., [5, 10, 19]). Though there are differences in the optimal combination of variables to obtain the optimal values at each dataset, our general results suggest using post-filtering with a power decay function. In terms of the algorithm, using user- and item-based combined helps to overcome the weaknesses of each method and performs consistently well, but the improvement over item-based *CF* is actually minimal.

In future work, we will include a more detailed analysis on parameter tuning, studying the relations between graph properties and optimal metric performance, and the use of matrix factorization techniques. First, although we tested two ways of tuning the parameters for the different decay functions, we used the same values for all users (mean vs. median of the Δt distributions) and we can explore further by setting parameter values for each user, based on their tagging habits. Second, although post-filtering with power decay had the most consistent results in general, we realized important differences among datasets so it might be interesting to study graph properties of each dataset and a relation with their performance with combinations of algorithms, filtering step and decay function. Finally, matrix factorization techniques are usually cited as the state-of-art techniques in recommender systems in terms of accuracy, and previous works have studied how to incorporate time, implicit feedback, and contextual variables [8], nevertheless, no previous work have yet explored how to integrate time-decay functions into the matrix factorization recommendation framework of a social tagging system.

Acknowledgments. This work was carried out during the tenure of an ERCIM "Alain Bensoussan" fellowship program by the second author.

6. REFERENCES

[1] P. G. Campos, F. Díez, and I. Cantador. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *UMUAI*, pages 67–119, 2014.

[2] Y. Ding and X. Li. Time weight collaborative filtering. In *Proc. CIKM'05*, pages 485–492, 2005.

[3] J. Gemmell, T. Schimoler, M. Ramezani, L. Christiansen, and B. Mobasher. Improving FolkRank with item-based collaborative filtering. In *Proc. Workshop on Recommender Systems and the Social Web*, 2009.

[4] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM TOIS*, 22(1):5–53, 2004.

[5] C.-L. Huang, P.-H. Yeh, C.-W. Lin, and D.-C. Wu. Utilizing user tag-based interests in recommender systems for social resource sharing websites. *Knowledge-Based Systems*, 56:86–96, 2014.

[6] R. Jäschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag recommendations in folksonomies. In *Proc. PKDD'2007*, pages 506–514. 2007.

[7] Y. Koren. Collaborative filtering with temporal dynamics. In *Proc. KDD'09*, pages 447–456, 2009.

[8] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[9] D. Kowald, E. Lacic, and C. Trattner. Tagrec: Towards a standardized tag recommender benchmarking framework. In *Proc. HT '14*, 2014.

[10] D. Kowald, P. Seitlinger, C. Trattner, and T. Ley. Long time no see: The probability of reusing tags as a function of frequency and recency. In *Proc. WWW'14*. ACM, 2014.

[11] E. Lacic, D. Kowald, P. Seitlinger, C. Trattner, and D. Parra. Recommending items in social tagging systems using tag and time information. *arXiv preprint arXiv:1406.7727*, 2014.

[12] T. Q. Lee, Y. Park, and Y.-T. Park. A time-based approach to effective recommender systems using implicit feedback. *Expert systems with applications*, 34(4):3055–3062, 2008.

[13] D. Li, P. Cao, Y. Guo, and M. Lei. Time weight update model based on the memory principle in collaborative filtering. *Journal of Computers*, 8(11):2763–2767, 2013.

[14] D. Parra and X. Amatriain. Walk the talk: Analyzing the relation between implicit and explicit feedback for preference elicitation. In *Proc. UMAP'11*, pages 255–268, 2011.

[15] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: an open architecture for collaborative filtering of netnews. In *Proc. CSCW'94*, pages 175–186, 1994.

[16] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proc.*, WWW'01, pages 285–295, 2001.

[17] C. Trattner, D. Kowald, and E. Lacic. Tagrec: towards a toolkit for reproducible evaluation and development of tag-based recommender algorithms. *ACM SIGWEB Newsletter*, (Winter):3, 2015.

[18] D. Wu, Z. Yuan, K. Yu, and H. Pan. Temporal social tagging based collaborative filtering recommender for digital library. In *Proc. ICADL'2012*, pages 199–208. 2012.

[19] N. Zheng and Q. Li. A recommender system based on tag and time information for social tagging systems. *Expert Syst. Appl.*, 38:4575–4587, 2011.