Time Weight Content-Based Extensions of Temporal Graphs for Personalized Recommendation

Armel Jacques Nzekon Nzeko'o[1,2,3], Maurice Tchuente[1,2] and Matthieu Latapy[3]
[1]*Sorbonne Universités, UPMC Université Paris 06, IRD, UMI 209 UMMISCO, F-93143, Bondy, France*
[2]*Université de Yaoundé I, CETIC, LIRIMA, Equipe IDASCO, BP 812, Yaoundé, Cameroun*
[2]*Sorbonne Universités, UPMC Université Paris 06, CNRS, UMR 7606, LIP6, F-75005, Paris, France*
{armel.nzekon, maurice.tchuente}@uy1.uninet.cm, matthieu.latapy@lip6.fr

Keywords: Temporal recommendation; Long- and short-term preferences; Session-based temporal graph; Time weight content-based graph; Time-averaged hit ratio; pagerank; Injected preference fusion.

Abstract: Recommender systems are an answer to information overload on the web. They filter and present to customer, a small subset of items that he is most likely to be interested in. Since user's interests may change over time, accurately capturing these dynamics is important, though challenging. The Session-based Temporal Graph (STG) has been proposed by Xiang et al. to provide temporal recommendations by combining long- and short-term preferences. Later, Yu et al. have introduced an extension called Topic-STG, which takes into account topics extracted from tweets' textual information. Recently, we pushed the idea further and proposed Content-based STG. However, in all these frameworks, the importance of links does not depend on their arrival time, which is a strong limitation: at any given time, purchases made last week should have a greater influence than purchases made a year ago. In this paper, we address this problem by proposing Time Weight Content-based STG, in which we assign a time-decreasing weight to edges. Using Time-Averaged Hit Ratio, we show that this approach outperforms all previous ones in real-world situations.

1 INTRODUCTION

The amount of information in web sites like Amazon, Netflix and Last.fm is considerable, and still growing. For users, browsing and searching in such data as becomes very difficult. To address the problem, recommender systems solve this issue by filtering and presenting to users small subsets of items that they are likely to be interested in.

Early recommender systems did not take into account temporal information (Adomavicius & Alexander, 2005). However, user's interests evolve with time, and may be affected by events such as weddings. Modern recommender systems aim at capturing such effects. Sugiyama et al. (Sugiyama, Hatano, & Yoshikawa, 2004) proposed to adapt results according to time-evolving user profiles, based for instance on browsing history. Similarly, Ding et al. (Ding & Li, 2005) proposed to weight user interests according to their age. Some recommender systems focus on short-term preferences (Lathia, Hailes, & Capra, 2009). Some authors capture both long- and short-term preferences and combine them for recommendations (Billsus & Pazzani, 2000; Li, Yang, Wang, & Kitsuregawa, 2007).

Interest in such recommender systems increased considerably since the Koren victory at the 2009 Netflix grand prize (Koren, 2009). However, it was based on a dataset with items rated by users, which are rarely available in practice. Instead, data often contains the history of user in term of their interactions with proposed items. For instance, Last.fm offers datasets in which each line indicated the fact that user *u* listened to song *i* at time *t*.

In this line of research, Xiang et al. (Xiang, et al., 2010) propose Session-based Temporal Graphs (STG), which model long- and short-term preferences separately. However, they ignore features of items, and so they miss for instance the fact that interest in a piece of music is related to its author. In order to improve this, Yu et al. (Yu, Shen, & Yang, 2014) extend the STG into Topic-STG for personalized tweet recommendation. They add *topic nodes* to the STG and link tweets to their topics.

These recommender graphs process edges regardless of their age. This fails to capture the fact that recent transactions are the most likely to reflect user preferences in the near future (Ding & Li, 2005). To take this into account, we propose here to weight edges according to their age, so that older edges have lower influence. We propose the Time Weight Content-based STG, in which edges are labelled with their last occurrence time, and we use an exponential decay function proposed by Ding et al. (Ding & Li, 2005) to weight edges accordingly.

Section 2 shortly presents Session-based Temporal Graph and the two recommendation algorithms on which our work is built. Section 3 introduces our Time Weight Content-based STG model. Section 4 is devoted to experiments and results. We discuss related work in Section 5, and we summarize our findings in Section 6.

2 BACKGROUND

We use the notations and definitions proposed by Xiang et al. (Xiang, et al., 2010), together with some additional concepts related to content and time, summarized in Table 1.

2.1 Session-based Temporal Graph

We consider data under the form of a link stream, i.e. a set of triples (t, u, i) representing the fact that user u has selected item i at time t. For each user u (resp. each item i), we define user node v_u (resp. item node v_i). We denote by **T** the time span of the dataset and we divide **T** into time slices of equal duration. For each of these slices T, we define session node $v_{u,T}$

A session-based temporal graph $G(U, S, I, E, w)$ is a directed bipartite graph with three types of nodes: U is the set of user nodes, S the set of session nodes and I the set of item nodes. The function w: $E \rightarrow \boldsymbol{R}$ is a non-negative weight function for edges. The set of edges, E, is obtained as follows. For each triplet (t, u, i), let us consider T the time slice to which t belongs. Then, E contains edges (v_u , v_i) and (v_i , v_u), which represent long-term preference between user u and item i; and E contains ($v_{u,T}$, v_i) and (v_i , $v_{u,T}$), which represent short-term preferences.

The weight function is defined as:

$$w(v, v') = \begin{cases} 1 & v \in U \cup S, v' \in I \\ \eta_u & v \in I, v' \in U \\ \eta_s & v \in I, v' \in S \end{cases} \qquad (1)$$

Table 1: Notations and definitions.

Symbol	Description
G	bipartite graph STG
CG	bipartite graph Content-based STG
TG	bipartite graph Time weight Content-based STG
E	edge set in any graph
V	set of all nodes in any graph
U, I, S, C	user node set, item node set, session node set, content node set
v_u , v_i , $v_{u,T}$, v_c	user node, item node, session node, content node
w	weight function defined on STG edges
w_C	weight function defined on Content-based STG edges
w_T	time weight function defined on Time weight Content-based STG edges
$\psi(v_k , v_{k+1})$	propagation function of IPF from v_k to v_{k+1}
$out(v)$	out node set of the node v
ρ	parameter to control the preference propagation
β	dose of long-term preference injected to user node
η	parameter to adjust the edge weight from item nodes to user/session nodes
η_c	parameter to control the influence of content features in the preference propagation
τ_0	parameter used to compute the time weight function
α	damping factor for Pagerank personalization

In (1), η_u models the influence of long-term preferences and η_s models the influence of short-term preferences. To simplify the model, we can use $\eta = \eta_u / \eta_s$ for η_u and 1 for η_s

Fig. 1 is an example of STG with 3 user nodes, 5 session nodes, 7 item nodes and 2 time slices. It shows that user u_1 has selected items i_1, i_2, user u_2 has selected items i_3, i_4 and user u_3 has selected item i_5 during the first time slice T_1. During the second time slice T_2, user u_1 has selected i_3 and user u_3 has selected i_6 and i_7.

2.2 Temporal Personalized Random Walk

The Temporal Personalized Random Walk (TPRW) (Xiang, et al., 2010) is a personalization of the Pagerank algorithm defined by Page et al. (Page, Brin, Motwani, & Winograd, 1999) for nodes ranking in graphs. It was defined to tackle temporal

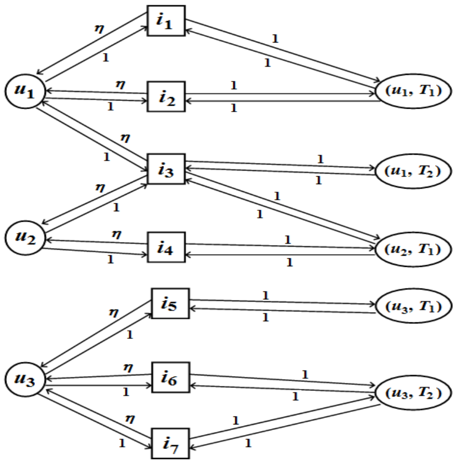

Figure 1: Example of STG.

recommendation using the idea of Haveliwala (Haveliwala, 2002). It corresponds to the following formula:

$$PR = \alpha \cdot M \cdot PR + (1 - \alpha) \cdot d \quad (2)$$

where α is the damping factor, M is a transition matrix and vector d is a user-specific personalized vector indicating which nodes the random walker will jump to after a restart.

When making recommendations for user u, vector d favors user node v_u and the most recent session node $v_{u,T}$ as follows:

$$d(v) = \begin{cases} \beta & v = v_u \\ 1 - \beta & v = v_{u,T} \\ 0 & otherwise \end{cases} \quad (3)$$

In other words, long-term preferences are injected to user node v_u and short-term preferences are injected to session node $v_{u,T}$ through vector d.

When we implement the Pagerank with iterative power law method, we stop when the difference of two consecutives rank vectors is of norm less than or equal to a threshold ε. To circumvent the cases where the convergence is very slow we stop after a maximum number of one hundred iterations.

2.3 Injected Preference Fusion

The IPF algorithm is an extension of the random walk with injection of preferences and customization of preference propagation. To recommend items to a user u, the algorithm proceeds in 3 steps:

- Injection of long-term preferences β on the user node v_u and injection of short-term preferences $(1 - \beta)$ on the most recent session node $v_{u,T}$ of user u.
- Propagation of preferences by random walk of length 3 on the graph according to the formula.

$$\psi(v_k, v_{k+1}) = \left(\frac{w(v_k, v_{k+1})}{\sum_{v' \in out(v_k)} w(v_k, v')}\right)^{\rho} \quad (4)$$

- where $out(v_k)$ denotes the set of out-neighbors of node v_k, ρ is a parameter used to tune the propagation process, $w(v_k, v_{k+1})$ is the weight of arc (v_k, v_{k+1}) and $\psi(v_k, v_{k+1})$ is the proportion of preference of v_k that is propagated to v_{k+1}.
- Recommendation of Top-N items that have received the greatest preference values and that user u has not yet selected.

The IPF random walk length is limited to 3 following experimental result (Xiang, et al., 2010).

3 TIME WEIGHT CONTENT-BASED EXTENSIONS OF STG

In this section, we first illustrate how to construct Content-based STG (CSTG) which is similar to Topic-STG (Yu, Shen, & Yang, 2014). We end by showing how to construct Time Weight Content-based STG (TCSTG).

3.1 Content-based Session-based Temporal Graph

The basic STG model neglects item properties which can contain significant information for the prediction of user's behavior. This motivated Phuong et al. (Phuong, Thang, & Phuong, 2008) to add to the user-item bipartite graph, new nodes corresponding to content. The same idea is applied here to obtain Content-based STG.

To construct the Content-based STG, we need to have item properties in our data, so we don't use a set of triples like in the construction of STG. We rather use a set of quadruples (t, u, i, c) where t, u and i have the same meaning as in STG, and c is a content feature of i.

Content-based STG $CG(U, S, I, C, E, w_C)$ is a directed graph obtained from the STG $G(U, S, I, E, w)$ by adding for any link (t, u, i, c), the six additional arcs (v_u, v_c), (v_c, v_u), $(v_{u,T}, v_c)$, $(v_c, v_{u,T})$, (v_i, v_c) and (v_c, v_i). With respective weights 1, η, 1, 1, η_c, η_c as illustrated in Fig. 2.

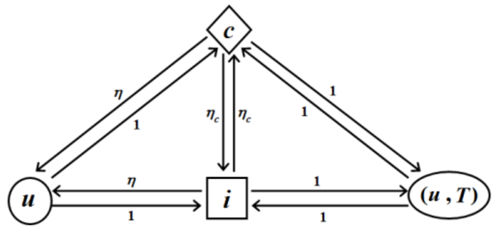

Figure 2: Edge weights in content-based STG.

3.2 Time weight Content-based Session-based Temporal Graph

The Content-based STG neglects the ages of edges when assigning weights. So, it cannot capture the evolution of users' interest which we assume to be sensitive to time as suggested by Ding et al. (Ding & Li, 2005). The recommendation model presented here assigns a greater weight to recent edges and lower weight to older edges. More precisely, the weight of the arc (v, v') is defined by:

$$w_T(v, v') = f(t) \cdot w(v, v') \quad (5)$$

where $w(v, v')$ is the weight in the graph without time weight, t is the most recent time at which edge (v, v') appears and $f(t)$ is a time-dependent decay function as in (Ding & Li, 2005). Here we take

$$f(t) = e^{-\lambda \cdot (t_r - t)} \quad (6)$$

where λ is the decay rate and $(t_r - t)$ is the difference in second between time t_r at which we are making recommendations and t.

The parameter λ can also be defined as $\lambda = 1/\tau_0$ where τ_0 is the delay after which the weight of an edge reduces by 1/2. τ_0 is also called half life parameter.

4 EXPERIMENTS

We have conducted a set of experiments to examine the performance of Time weight Content-based STG. For each model, we consider various values of parameters and we retain the best performance. We also implemented the classic bipartite user-item graph (BIP) to show the effects of taking into account long- and short-term preferences in graph models.

The experiment environment is as follow: the executions of our programs are done using a computer with 64Go of RAM and 16 processors Intel of 2.93GHz and 4MB of cache. For implementation, we have used the Python 2.7 language and the Networkx 1.11 module for graph manipulation. Note that we have changed the Networkx *Pagerank* in order to stop when convergence is not reached after 100 iterations. We used SQLite 3 as DBMS and Matplotlib 1.4.0 to produce graphics.

4.1 Data description

Following the example of Xiang et al, our goal is to make recommendations based on implicit data from various real world domains. To this effect, we used three sets of implicit data: two on citations and social bookmarking namely CiteUlike and Delicious (Cantador, Brusilovsky, & Kuflik, 2011); the third taken from Last.fm (Celma, 2010) is a web site where users can listen to songs.

We model our data as link streams {(t, u, i, c)}, where any quadruple has different interpretation depending on domains. In the case of CiteUlike and Delicious, each quadruplet means that user u has bookmarked page i at time t with tag c. And for the Last.fm data, this means that user u has listened to song i at time t and c is the author of i.

Before modeling our data as link streams, we performed a filtering by ignoring items and users that did not appear a number of times higher than a given threshold σ. Table 2 provides details on our data: date of the first link, date of the last link, total duration of link streams, threshold used, number of users, number of items, number of content features and number of links.

link, date of the last link, total duration of link streams, threshold used, number of users, number of items, number of content features and number of links.

Table 2: Data statistics.

	Start date	**End date**	**Duration**	**σ**
CiteUlike	2010-01-01	2010-07-02	183 days	10
Delicious	2010-05-11	2010-11-09	183 days	7
Last.fm	2005-02-14	2005-08-16	183 days	8

	Users	**Items**	**Content**	**Links**
CiteUlike	1318	424	4216	16885
Delicious	894	298	2789	13825
Last.fm	135	1054	225	41604

4.2 Experiment and evaluation

Before starting experiments, we have to divide the link streams into time windows of a fixed length Δ; it can be one hour, one day, one week or one month. We fix Δ to 15 days. To simplify the experimentation process, we adopt the same Δ as the length of session when constructing STG. Here after, N denotes the number of time slices.

For each time window W_k, for k=1,..,N-1, we proceed as follows:

- Construct the graphs corresponding to data of W_1, W_2, .. ,W_k.
- Compute the Top-N recommendations for users who have selected at least one "new item" during the time window W_{k+1}.
- Evaluate the algorithm by computing the ratio of users for which at least one of these Top-N items recommends has been selected during W_{k+1}. This proportion is also call Hit Ratio (Karypis, 2001).

After determining the Hit Ratio for each window, compute the overall Time Averaged Hit Ratio that is a weighted combination of the N-1 values obtained above for the Hit Ratio. In this combination, the weight of a Hit Ratio is the number of corresponding users.

4.2 Exploration of the range of the parameters

Let us see how the parameters are obtained in Table 3. We proceed as in (Xiang, et al., 2010). The parameters correspond to the vector [τ0, β, η, η_c, ρ, α], whose components are numbered 1, 2, 3, 4, 5, 6 from left to right. This vector is initialized to [0, 0.5, 0.5, 0.5, 0.5, 0.5]. Then, we consider the values of τ0 shown in the second row of Table 3, while maintaining the other parameters at their initial value 0.5. We perform ten experiments and take for τ0 the value corresponding to the best performance. For instance we obtain for τ0 the interval [15, 60] for IPF-TCSTG and [7, 30] for TPRW-TCSTG in CiteUlike dataset as shown in Table 4. Given this optimal value for τ0 we then give to β the eleven successive values shown in the third row of Table 3, while maintaining the other parameters at their initial value. We obtain for β the interval [0.5, 0.6] for IPF-TCSTG and [0.4, 1] for TPRW-TCSTG in CiteUlike dataset as shown in Table 4. This process is repeated for the remaining parameters.

Figure 3 shows all the variations of Time-Averaged Hit Ratio with parameter values in the case of CiteUlike. The complete set of parameters explored is shown in Table 3 and the best values obtained with this procedure are shown in Table 4.

Table 3: Parameters values.

Parameters	**Initial value**	**Set of values**
τ0 (in days)	0	0, 1, 7, 15, 30, 45, 60, 90, 180, 365
β	0.5	0.1 × i for i = 0..10
η	0.5	0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.5, 3, 5, 10, 15, 20, 30, 50, 100
η_c	0.5	0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.5, 3, 5, 10, 15, 20, 30, 50, 100
ρ	0.5	0.1 × i for i = 0..10
α	0.5	0.1 × i for i = 0..10

Table 4: Parameters best values.

CiteUlike	τ**0**	β	η	η_c	ρ	α
IPF-BIP	-	-	-	-	0.1-0.3	-
IPF-STG	-	0.0-0.5	0.1-0.3	-	0.1-0.7	-
IPF-CSTG	-	0.5-1	0.0-0.9	0.2-0.5	0.5-0.6	-
IPF-TCSTG	15-60	0.5-0.6	0.1-0.9	0.4-1.5	0.1-1	-
TPRW-BIP	-	-	-	-	-	0.1-0.9
TPRW-STG	-	0.0-0.7	0.3-0.6	-	-	0.1-0.7
TPRW-CSTG	-	0.5-0.8	0.3-0.6	0.1-0.7	-	0.1-0.6
TPRW-TCSTG	7-30	0.4-1	0.3-1.5	0.5-3	-	0.5-0.8
Delicious	τ**0**	β	η	η_c	ρ	α
IPF-BIP	-	-	-	-	0.1-10	-
IPF-STG	-	0.0-0.4	0-0.1	-	0.1-0.6	-
IPF-CSTG	-	0.5	15-50	0.3-0.9	0.4-1.5	-
IPF-TCSTG	1-7	0.5-0.6	0.5-0.8	50-100	0.5-1.5	-
TPRW-BIP	-	-	-	-	-	0.1-0.9
TPRW-STG	-	0.0-0.4	0.1-0.2	-	-	0.2-0.5
TPRW-CSTG	-	0.0-0.6	15-100	0.2-0.8	-	0.5-0.7
TPRW-TCSTG	7	0-1	0.5-0.8	0-0.1	-	0.1-04
Last.fm	τ**0**	β	η	η_c	ρ	α
IPF-BIP	-	-	-	-	0.1-0.8	-
IPF-STG	-	0.5-1	0.9-1.5	-	1.5-10	-
IPF-CSTG	-	0-0.4	0-0.3	30-100	0.4-0.6	-
IPF-TCSTG	1-15	0-0.4	0.1-0.3	1-100	10-50	-
TPRW-BIP	-	-	-	-	-	0.1-0.5
TPRW-STG	-	0.5-0.7	0.1-1.5	-	-	0.2-0.5
TPRW-CSTG	-	0.2-0.4	0.2-0.5	5-30	-	0.4-0.6
TPRW-TCSTG	30-90	0.5-0.7	0.4-0.6	1-5	-	0.4-0.7

5 RELATED WORK

In this section, we present some work on time aware recommender systems followed by recommender systems that use item properties. Finally, we present some graph-based recommender systems.

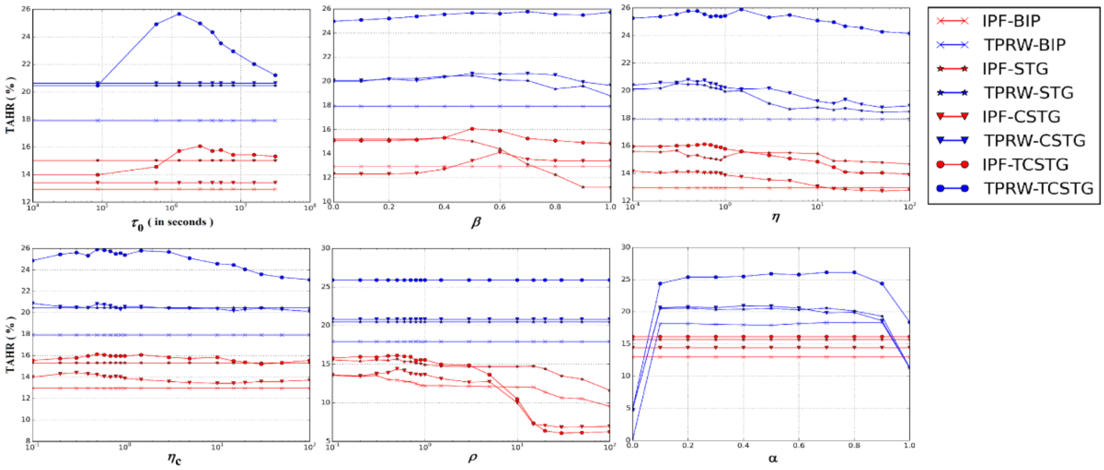

Figure 3: Variation of Time Averaged Hit Ratio with parameter values in the case of CiteUlike.

5.1 Time aware recommender systems

Ding et al. (Ding & Li, 2005) propose the use of an exponential decay function to assign greater weights to latest ratings when computing similarities in collaborative filtering. Subsequently, Liu et al. (Liu, Zhao, Xiang, & Yang, 2010) have proposed an incremental collaborative filtering where two decay functions are used: one to compute similarities and the other for prediction. Recently, Karahodža et al. (Karahodža, Donko, & Šupić, 2015) assumed that the importance of ratings (resp. interest granted to an item category) decrease in a similar manner for similar users. These hypotheses make it possible to improve user-based collaborative filtering.

Some recommender systems which use sliding time window are based on the assumption that importance of information is ephemeral. Thus, Lathia et al. (Lathia, Hailes, & Capra, 2009) set a time window size, then, any information is used during one time slice and ignored at the next time window. During each time window, collaborative filtering prediction errors are computed for different values of the parameter k of the of the k-nearest neighbors approach. The value which minimizes the prediction error is used for the next time window. Such recommender systems only capture short-term preferences.

Some works are not based only on short-term preferences but also considers that importance of certain information persists over time. Those works take long-term preferences into account and propose mechanisms to combine both preference types (Li, Yang, Wang, & Kitsuregawa, 2007; Kang & Choi, 2011). The STG (Xiang, et al., 2010) that we extend in this work is a part of those works but it has the particularity that it is a graph and was designed with goal of improving Top-N recommendations on implicit data. However, this model ignores item properties.

5.2 Content-based recommender systems

The content-based recommender systems seek to recommend similar items to the one the user already like. As Lops et al. (Lops, De Gemmis, & Semeraro, 2011) argue, the basic idea is to match features associated to users' preferences and items so as to recommend new items that address their needs. This approach is used in various domains ranging from recommending books on Amazon website based on their description (Mooney & Roy, 2000), to recommending authors and papers on a research field based on tags assigned by users (Diederich & Iofciu, 2006) and recommending web pages (Pazzani, Muramatsu, & Billsus, 1996). Pazzani et al. (Pazzani & Billsus, 2007) illustrate the common treats of most recommender systems based on this approach.

Although content-based recommender systems can propose items that have not already been purchased in the past, it is also useful to use user similarities by combining this approach with collaborative filtering techniques. The result is a hybrid system. Indeed, Balabanovic et al. (Balabanović & Shoham, 1997) show that the combination of collaborative filtering and content-based filtering may result in a recommender system that eliminates the weaknesses of both approaches. This is confirmed by Basu et al. (Basu, Hirsh, & Cohen, 1998) for recommendation of videos by taking into accounts not only ratings and user relationships for collaborative filtering, but also adds information related to videos. In this paper, we have used a graph model to realize this combination.

5.3 Graph based recommender systems

The simplest graph-based recommender systems only use user-item links that are available at a given time to build a bipartite graph in which there are user nodes and item nodes. A bidirectional edge is created between a user node and an item node if the user has purchased the concerned item. Finally, an item is recommended to a user if the user has not yet purchased that item and if there is a path from the user to that item. The most used recommender algorithms on the graphs are based on the random walk (Gori & Pucci, 2006; Baluja, et al., 2008), like Pagerank and IPF which are used in this paper.

The use of graph paths to recommend new items to each user reflects the logic of collaborative filtering logic as recommended items are those similar users have already purchased. However, such recommender graphs do not take into consideration information linked to item properties. To remedy this limitation, Phuong et al. (Phuong, Thang, & Phuong, 2008) have constructed a recommender graph in which they have added a third node type: the type "content". This type of node represents any property of the item to recommend for example author and genre of songs. The obtained recommender system is actually a combined collaborative filtering and content-based filtering.

Although the graph of Phuong et al. (Phuong, Thang, & Phuong, 2008) takes item properties into account, this model ignores the temporal aspect of data and therefore cannot accurately capture short- and long-term preferences of users. Thus, Yu et al. (Yu, Shen, & Yang, 2014) propose the Topic-STG which combines those two preference types using the STG advantages, then made an extension by inserting topics to which tweets belong. This allows them to have better performance than STG. However, the Topic-STG (Yu, Shen, & Yang, 2014) handles edges regardless of age. This is not in accordance with the hypothesis of concept drift which suggests that recent data should have more consideration than older. For this purpose, we propose a new extension of STG according to works of Li et al. (Li & Tang, 2008) where they decrease the weight of older edges using a decay function. Edges weights are update in constant frequencies.

5 CONCLUSION

This paper proposes time weight content-based extensions of the temporal graph model introduced by Liang Xiang et al. We represent content by nodes, similarly to Jianjun Yu et al., but we penalize older interactions. Using the temporal personalized pagerank algorithm, we show that this improves obtained recommendations. This gives evidence of the fact that the age of interactions is a relevant feature for recommender systems, and our approach is able to take benefit from it. This open promising perspectives regarding the best ways to model interaction ages, and to which extent this may improve recommendation algorithms.

ACKNOWLEDGEMENTS

This work is funded in part by the African Center of Excellence in Information and Communication Technologies (CETIC), with the support of the World Bank, by the European Commission H2020 FETPROACT 2016-2017 program under grant 732942 (ODYCCEUS), by the ANR (French National Agency of Research) under grants ANR-15-CE38-0001 (AlgoDiv) and ANR-13-CORD-0017-01 (CODDDE), by the French program "PIA - Usages, services et contenus innovants" under grant O18062-44430 (REQUEST), and by the Ile-de-France program FUI21 under grant 16010629 (iTRAC).

REFERENCES

Adomavicius, G., & Alexander, T. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions , 17* (6), 734-749.

Balabanović, M., & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM , 40* (3), 66-72.

Baluja, S., Seth, R., Sivakumar, D., Jing, Y., Yagnik, J., Kumar, S., et al. (2008, April). Video suggestion and discovery for youtube: taking random walks through the view graph. *In Proceedings of the 17th international conference on World Wide Web. ACM* , 895-904.

Basu, C., Hirsh, H., & Cohen, W. (1998, July). Recommendation as classification: Using social and content-based information in recommendation. *In Proceedings of the fifteenth national/tenth conference* *on Artificial intelligence/Innovative applications of artificial intelligence* , 714-720.

Billsus, D., & Pazzani, M. J. (2000). User modeling for adaptive news access. *User modeling and user-adapted interaction , 10* (2-3), 147-180.

Cantador, I., Brusilovsky, P., & Kuflik, T. (2011). 2nd Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011). *In Proceedings of the 5th ACM conference on Recommender systems* .

Celma, Ò. (2010). *Music Recommendation and Discovery in the Long Tail.* Springer.

Diederich, J., & Iofciu, T. (2006, October). Finding Communities of Practice from User Profiles Based On Folksonomies. *In Innovative Approaches for Learning and Knowledge Sharing, EC-TEL Workshop Proc.* , 288-297.

Ding, Y., & Li, X. (2005, October). Time Weight Collaborative Filtering. *In Proceedings of the 14th ACM international conference on Information and knowledge management, ACM* , 485-492.

Gori, M., & Pucci, A. (2006). Research Paper Recommender Systems: A Random-Walk Based Approach. *In International Conference on Web Intelligence, IEEE/WIC/ACM* , 778-781.

Haveliwala, T. H. (2002, May). Topic-sensitive pagerank. *In Proceedings of the 11th international conference on World Wide Web* , 517-526.

Kang, J., & Choi, J. (2011, April). An Ontology-Based Recommendation System Using Long-Term and Short-Term Preferences. *In International Conference on Information Science and Applications (ICISA),IEEE* , 1-8.

Karahodža, B., Donko, D., & Šupić, H. (2015, May). Temporal Dynamics of Changes in Group Users Preferences in Recommender Systems. *In 38th International Convention on Information and Communication Technology, Electronics and Microelectronics, IEEE* , 1262-1266.

Karypis, G. (2001, October). Evaluation of Item-Based Top-N Recommendation Algorithms. *In Proceedings of the tenth international conference on Information and knowledge management* , 247-254.

Koren, Y. (2009). The bellkor solution to the netflix grand prize. *Netflix prize documentation , 81*.

Lathia, N., Hailes, S., & Capra, L. (2009, July). Temporal Collaborative Filtering With Adaptive Neighbourhoods. *In Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, ACM* , 796-797.

Li, L., Yang, Z., Wang, B., & Kitsuregawa, M. (2007). Dynamic adaptation strategies for long-term and short-term user profile to personalize search. *In Advances in Data and Web Management* , 228-240.

Li, Y., & Tang, J. (2008). Expertise Search in a Time-varying Social Network. *In The Ninth International Conference on Web-Age Information Management, IEEE* , 293-300.

Liu, N. N., Zhao, M., Xiang, E., & Yang, Q. (2010). Online Evolutionary Collaborative Filtering. *In Proceedings of the fourth ACM conference on Recommender systems. ACM* , 95-102.

Lops, P., De Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. *In Recommender systems handbook. Springer US* , 73-105.

Mooney, R. J., & Roy, L. (2000, June). Content-based book recommending using learning for text categorization. *In Proceedings of the fifth ACM conference on Digital libraries* , 195-204.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999, November). The PageRank citation ranking: bringing order to the web. *Technical Report, Stanford InfoLab* .

Pazzani, M. J., & Billsus, D. (2007). Content-Based Recommendation Systems. *In The adaptive web. Springer Berlin Heidelberg* , 325-341.

Pazzani, M., Muramatsu, J., & Billsus, D. (1996). Syskill & Webert: Identifying interesting web sites. *In Proceedings of the thirteenth American Association for Artificial Intelligence AAAI/IAAI , 1,* 54-61.

Phuong, N. D., Thang, L. Q., & Phuong, T. M. (2008). A Graph-Based Method for Combining Collaborative and Content-Based Filtering . *In Trends in Artificial Intelligence. Springer Berlin Heidelberg* , 859-869.

Sugiyama, K., Hatano, K., & Yoshikawa, M. (2004, May). Adaptive web search based on user profile constructed without any effort from users. *In Proceedings of the 13th international conference on World Wide Web, ACM* , 675-684.

Xiang, L., Yuan, Q., Zhao, S., Chen, L., Zhang, X., Yang, Q., et al. (2010). Temporal recommendation on graphs via long-and short-term preference fusion. *In : Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM* , 723-732.

Yu, J., Shen, Y., & Yang, Z. (2014). Topic-STG : Extending the Session-based Temporal Graph Approach for Personalized Tweet Recommendation. *In Proceedings of the companion publication of the 23rd international conference on World wide web companion. International World Wide Web Conferences Steering Committee* , 413-414.