



A general graph-based framework for top- N recommendation using content, temporal and trust information

Armel Jacques NZEKON NZEKO'O^{*1,2,3}, Maurice TCHUENTE^{1,2}, Matthieu LATAPY³

¹Sorbonne Université, IRD, UMMISCO, F-93143, Bondy, France

²Université de Yaoundé I, CETIC, FS, Département d'Informatique, BP 812, Yaoundé, Cameroun

³Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, LIP6, F-75005, Paris, France

*Corresponding author: armel.nzekon@lip6.fr

DOI: [10.18713/JIMIS-300519-5-2](https://doi.org/10.18713/JIMIS-300519-5-2)

Submitted: February 27, 2019 - Published: May 12, 2019

Volume: 5 - Year: 2019

Issue: **Graph and network analysis**

Editor: Vincent Labatut

Abstract

Recommending appropriate items to users is crucial in many e-commerce platforms. One common approach consists in selecting the N most relevant items for each user. To achieve this, recommender systems rely on various kinds of information, like item and user features, past interest of users for items and trust between users. Current systems generally use only one or two such pieces of information, which limits their performance. In this paper, we design and implement GraFC2T2, a general graph-based framework to easily combine various kinds of information for top- N recommendation. It encodes content-based features, temporal and trust information into a graph model, and uses personalized PageRank on this graph to perform recommendation. Experiments are conducted on Epinions and Ciao datasets, and comparisons are done with systems based on matrix factorization and deep learning using F1-score, Hit ratio and MAP evaluation metrics. The results show that combining different kinds of information generally improves recommendation. This shows the relevance of the proposed framework.

Keywords

Top- N Recommendation; Graph; Collaborative Filtering; Content; Temporal information; Trust; PageRank; Link streams

I INTRODUCTION

Many e-commerce platforms have large and fast growing sets of items to present to users. For instance, Amazon had a total of 53.38 millions books as on January 10th, 2018¹. Such huge

¹<https://www.scrapehero.com/many-products-amazon-sell-january-2018/>

quantities of products make it challenging for users to search and find interesting items for them. Then, they often rely on the help provided by recommender systems.

Various approaches co-exist, the most classical ones being rating prediction and top- N recommendation (Steck, 2013). Rating prediction estimates the rating value that a user is likely to give to items. Top- N recommendation ranks items for a given user and selects the N most interesting ones, for a given N . Many research works are dedicated to rating prediction. This requires explicit rating data whereas, in many platforms dedicated for instance to e-commerce, ratings are not available, and recommender systems have to deal with implicit data such as users' purchase, browsing and streaming history. In such situations, top- N recommendation can still be carried out (Cremonesi *et al.*, 2010).

In addition to the previous remark, top- N recommender systems are everywhere from on-line shopping websites to video portals (Christakopoulou and Karypis, 2016). For all these reasons, we focus here on top- N recommendation problem from positive implicit feedback, a problem already considered in many papers such as Rendle *et al.* (2009); Ning and Karypis (2011); Shi *et al.* (2012) and Guo *et al.* (2017).

One of the main families of techniques, called Collaborative Filtering (CF), takes benefit from correlations between user interests. Initially, CF recommender systems focused only on user-item interactions (Konstan *et al.*, 1997; Herlocker *et al.*, 1999; Sarwar *et al.*, 2001) and did not integrate side information among the following list: item features like the genre of a movie or the author of a song, context of interactions like location, timestamps or weather, and trust between users. Since such side information strongly influences user choices (for instance, users may listen to a new song because they like the singer), performances of such systems may be limited. In addition, side information helps solving problems like cold start and data sparsity (Burke, 2002; Adomavicius and Tuzhilin, 2005; Massa and Avesani, 2007; Campos *et al.*, 2014).

For these reasons, much effort was devoted to the inclusion of side information into CF techniques. For instance, hybrid systems incorporate item features in order to combine CF and content-based filtering (CBF) (Burke, 2002; Chen *et al.*, 2016; Shu *et al.*, 2018). Likewise, a winning team of the Netflix competition (Koren *et al.*, 2009; Koren, 2009) included temporal information into a CF system in order to track the dynamics of user interests and increase recommendation accuracy. Including trust information in order to take into account the fact that people tend to adopt items already chosen by trusted friends is also possible (Papagelis *et al.*, 2005; Massa and Avesani, 2007; Guo *et al.*, 2017).

Some previous works consider only one type of side information, and therefore fail to capture the combined influence of several types of side information on user interests. Others works suggest that progress in this direction may significantly improve recommendation, and combine two kinds of side information into CF (Ning and Karypis, 2012; Yu *et al.*, 2014; Strub *et al.*, 2016; Nzekon Nzeko'o *et al.*, 2017). However, to the best of our knowledge, none of these approaches include content-based features, users' preferences temporal dynamics and trust relationships between users simultaneously.

Our goal in this paper is to propose a general graph-based recommender framework that makes it easy to combine variety of side information. However, recommender systems are used in very diverse situations, which makes the design of a fully general system out of reach. We therefore made several assumptions which, although very general, do not apply to some contexts. First, we focus on top- N recommendation task because it is prevalent in many on-line

shopping recommender systems like video portals. In addition, we considered the situations where the recommender system aims at offering each user a product that he/she has not yet selected in the past. In some situations, clients may repeatedly buy the same product, but this is a quite different problem. We also we assumed that recent activities are more important than older ones, a situation known as concept drift. This is often but not always true in practice; interest in a given kind of product may for instance be periodic, like for birthday gifts or seasonal needs. Extending our work in this direction is promising, when data is available. Finally, we consider positive links only (that typically represent a purchase), as this is the most prevalent case in practice; considering more subtle feedback from users, and in particular negative feedback, is a very promising direction for future work.

Contribution

In this paper, we propose GraFC2T2, a general graph-based framework for top- N recommendation combining content-based features, temporal information, and trust into a personalized PageRank system. The design of this framework is very modular in order to make it easy to include other side information and/or replace personalized PageRank by another graph-based method. Thanks to GraFC2T2, it becomes easy to explore the benefit of using various kinds of side information, and then to find appropriate parameters for combining them for particular applications. We conduct experiments on Epinions and Ciao datasets to illustrate the use of GraFC2T2, and we show that it outperforms state-of-the-art thanks to the increased use of side information.

Figure 1 summarizes the global architecture of GraFC2T2, made of two big parts: the recommender graph construction, and the use of this graph to perform recommendation. The recommender graph encodes available information by combining a basic graph, which we detail in Section II, with methods to capture content-based features and edge weight capturing time information, which we detail in Section III. Then, we use the obtained recommender graph to perform recommendation, with a trust-aware personalized PageRank detailed in Section IV.

Notice that our framework makes it possible to explore wide sets of modeling choices, as well as to incorporate additional possibilities if needed. We illustrate this on two real-world datasets from Epinions and Ciao in Sections V and VI. Section VII discusses related work.

This work builds upon our previous paper (Nzekon Nzeko'o *et al.*, 2017), which extends the Session-based Temporal Graph proposed by Xiang *et al.* (2010) by adding time-weight and content-based information. On the other hand, the data representation that we use is the link stream formalism, presented by Latapy *et al.* (2018). This model allowed us to propose the Link Stream Graph (Nzekon Nzeko'o *et al.*, 2019).

We provide an online implementation of our framework² in order to help other researchers and practitioners to conduct experiments on their own datasets, and to test the relevance of new ideas and features.

II DATA MODELING

We consider a set U of users, a set I of items, and a time interval T , and we assume that we observed the past interest of users in U for items in I during T . We model this data by a bipartite link stream $L = (T, U, I, E)$ where $E \subseteq T \times U \times I$ is a set of links: each link (t, u, i) in E represents a purchase (u bought product i at time t), an interest in a cultural item (like movie

²<https://github.com/nzekonarmel/GraFC2T2>

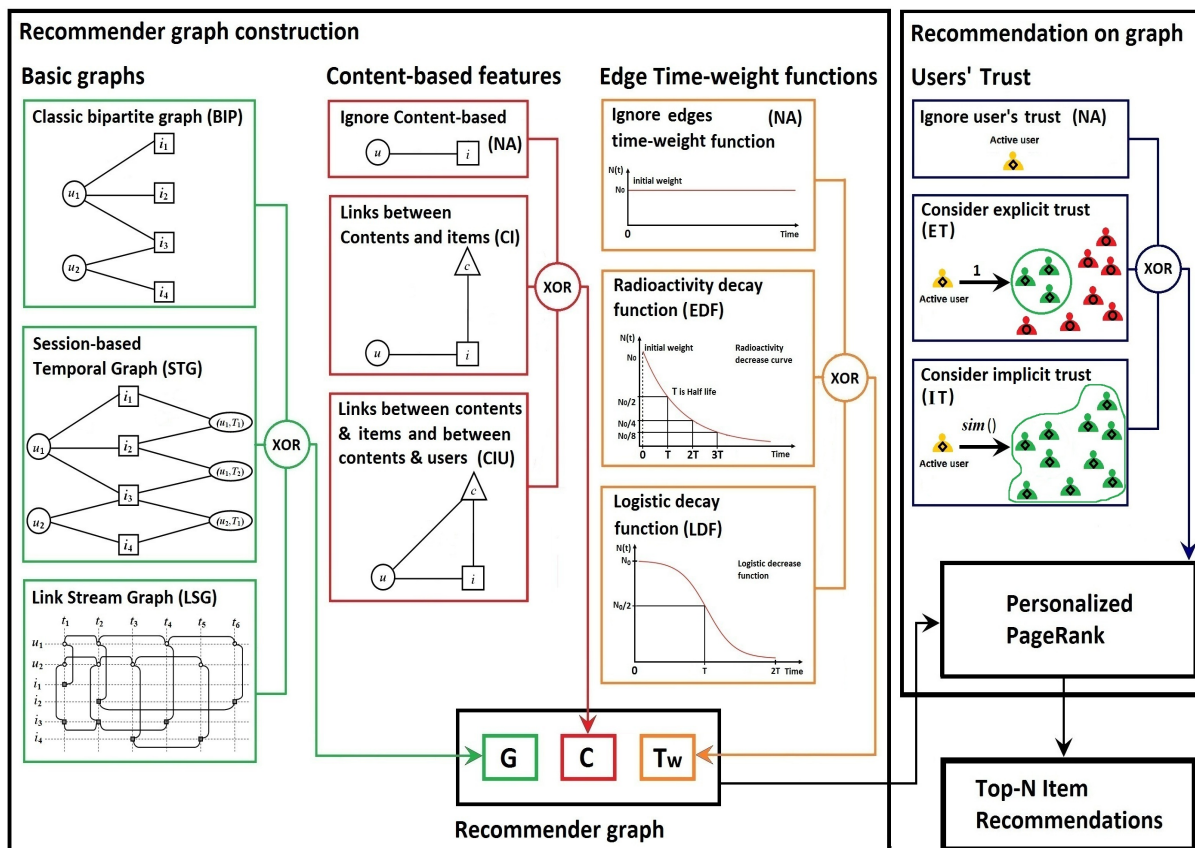


Figure 1: **The global architecture of GraFC2T2**, our general purpose graph-based recommender framework. Recommender graphs are built from three components: a basic graph that models user-item relations, content-based features that enrich basic graph, and link time-weight function that penalizes old edges, see Sections II and III. Then, we perform top- N recommendation over this graph using user trust and personalized PageRank, see Section IV.

watching or song listening), or another user-item relational event, depending on the application context. See Viard *et al.* (2016); Latapy *et al.* (2018) for a full description of the link stream formalism. In the following, we will illustrate definitions with the guiding example of Figure 2.

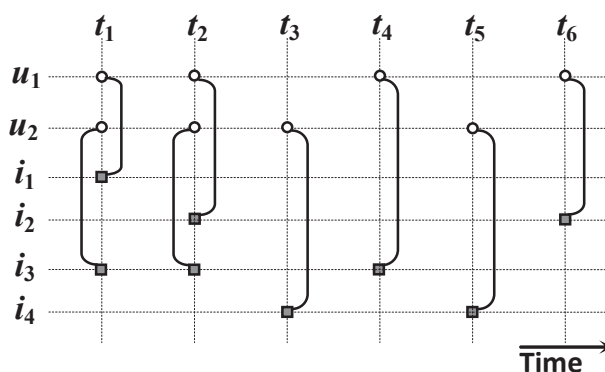


Figure 2: **Guiding example**: we consider the link stream $L = (T, U, I, E)$ in which the set of users is $U = \{u_1, u_2\}$, the set of items is $I = \{i_1, i_2, i_3, i_4\}$, the observation period is $T = [t_1, t_6]$, and $E = \{(t_1, u_1, i_1), (t_1, u_2, i_3), (t_2, u_1, i_2), (t_2, u_2, i_3), (t_3, u_2, i_4), (t_4, u_1, i_3), (t_5, u_2, i_4), (t_6, u_1, i_2)\}$. This means for instance that user u_1 was interested in item i_2 at time t_2 .

2.1 Classical bipartite graph

We first consider the most classical recommender graph introduced in the literature (Huang *et al.*, 2004; Baluja *et al.*, 2008), that we denote by BIP. It is a directed bipartite graph (U, I, E') where U and I are the set of users and items defined above, and $E' \subseteq U \times I$ is the set of links defined by $E' = \{(u, i) : \exists t \in T, (t, u, i) \in E\}$. In other words, u is linked to i in BIP if user u was interested in item i during the observation period. Figure 3a displays the BIP graph for the guiding example.

2.2 Session-based temporal graph

In a first attempt to capture time information, we then consider Session-based Temporal Graphs proposed by Xiang *et al.* (2010), that we denote by STG.

This graph encodes time information using a set S of session nodes defined as follows. First, for a given Δ , the observation interval T is divided into $\frac{|T|}{\Delta}$ time slices $T_k = [(k-1) \cdot \Delta, k \cdot \Delta]$ of equal duration Δ . Then, S contains the couples (u, T_k) such that there exists a link (t, u, i) in E with $t \in T_k$. In other words, each user leads to a session node (u, T_k) in S for each time interval T_k during which this user was active.

This finally leads to the definition of STG as a tripartite graph (U, I, S, E'') with U , I , and S defined above, and $E'' = E' \cup \{(u, T_k, i) : \exists t \in T_k, (t, u, i) \in E\}$. In other words, we add to BIP the nodes in S , and a link between each session node (u, T_k) and the items selected by user u during time slice T_k . Figure 3b shows the STG representation for the guiding example.

Notice that in the original model Xiang *et al.* (2010), any link from u to i has a weight 1 and any link from i to u has a weight η , where η is a parameter. For simplicity, we do not consider this parameter here (or, equivalently, $\eta = 1$), but it may easily be added if needed.

2.3 Link stream graph

In order to capture time information while avoiding the drawbacks of choosing a time window size Δ like for STG, we introduce the following link stream graph, that we denote by LSG (Nzekon Nzeko'o *et al.*, 2019).

This graph is first defined by a set of nodes representing users and items over time: $\{(t, u) : \exists i, (t, u, i) \in E\} \cup \{(t, i) : \exists u, (t, u, i) \in E\}$. In other words, each user u is represented by the nodes (t, u) such that a link involves u in L a time t , and each item is represented similarly.

We then define the set of links $\{(t, u), (t, i) : (t, u, i) \in E\} \cup \{(t, u), (t', u) : \exists i, (t, u, i) \in E, t' = \min\{x : x > t \text{ and } \exists i', (x, u, i') \in E\} \cup \{(t, i), (t', i) : \exists u, (t, u, i) \in E, t' = \min\{x : x > t \text{ and } \exists u', (x, u', i) \in E\}$. In other words, each user node (t, u) is linked to both the item nodes (t, i) such that $(t, u, i) \in E$ and to the next user node representing u . Item nodes are linked similarly. See Figure 3c for an illustration on our guiding example.

III ADDING CONTENT-BASED FEATURES AND TIME-WEIGHT FUNCTIONS

Once a basic recommender graph is built as explained in previous section, the GraFC2T2 framework adds elements to capture content-based and temporal features. Again, we propose several choices, and we present them below.

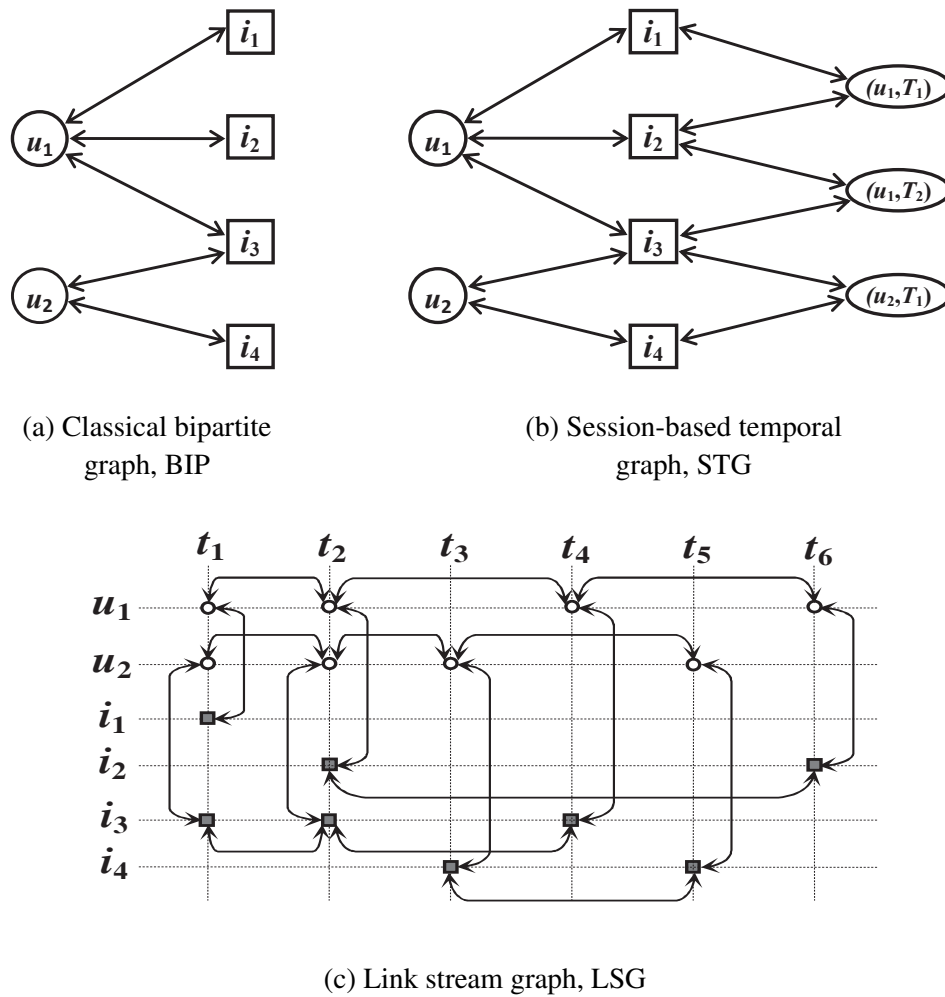


Figure 3: Classical bipartite graph, Session-based temporal graph and Link stream graph obtained from our guiding example. The weight of each edge is 1.

3.1 Content-based features

Let C be the set of all possible content-based features and let $g(i) \subseteq C$ be the subset of content-based features associated with item i , for any i . One element of $g(i)$ can be the category, the brand or the color of item i . Following the method proposed in (Nguyen *et al.*, 2008; Yu *et al.*, 2014; Nzekon Nzeko'o *et al.*, 2017), we model these features by content nodes that we link to item nodes in basic recommender graphs.

In the cases of BIP and STG, we add a content node c for each content-based feature c in C , and we link each item node i to the content node c for each c in $g(i)$. For LSG, we add a content node (t, c) for each (t, i) in the basic graph such that c is in $g(i)$, and we link (t, c) to (t, i) . We call this inclusion of content-based features CI because it adds links only between content and item nodes. See Figure 4.

We also propose a strategy linking content nodes to both item and user nodes, that we call CIU. The idea is to link user nodes to the content nodes of the items they are interested in. Therefore, in addition to CI additions, CIU adds to BIP a link (u, c) between each user node u and content node c whenever there is an item node linked to both u and c ; to STG a link between each session node (u, T_k) and content node c whenever there is an item node linked to both (u, T_k)

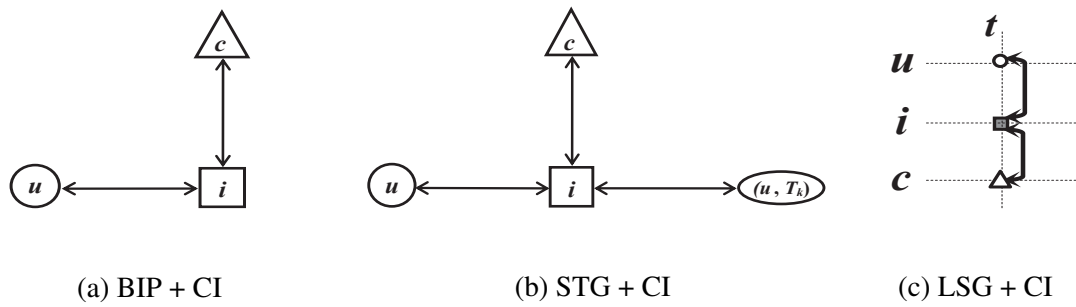


Figure 4: Inclusion of nodes and links representing content-based features with the CI strategy, for each basic recommender graph.

and c ; and to LSG a link between each user node (t, u) and content node (t, c) whenever there is an item node linked to both. See Figure 5.

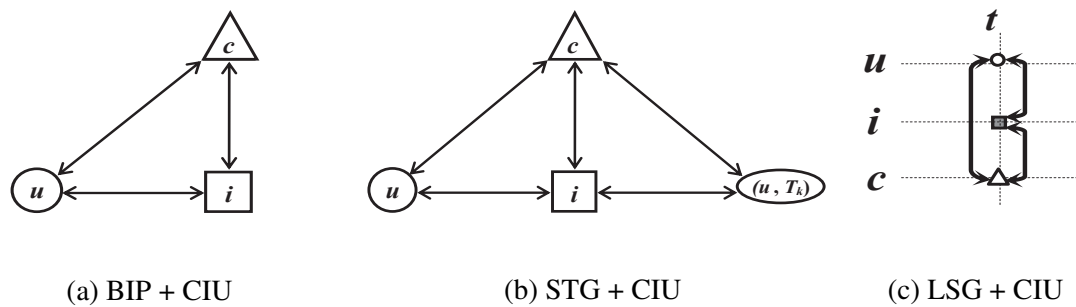


Figure 5: Inclusion of nodes and links representing content-based features with the CIU strategy, for each basic recommender graph.

Compared to CI, the CIU method increases the influence of content-based features linked to items that the target user has already selected in the past. In other words, the CIU method do a better promotion of items that have the same features as the choices of the target user.

3.2 Time-weight functions

Until now, we modeled time information directly within the structure of STG and LSG graphs, but their edge weights give a static view of previous user interests. Since such interests evolve over time, as pointed out for instance by Ding and Li (2005), this is not sufficient. We therefore follow the methodology proposed in that paper, consisting in adding time-dependent weights to the links of recommender graphs.

The idea is to give a high weight to recent links, and to decrease this weight with their age: the weight at time t of any link (a, b) whose most recent appearance time is $t_e \leq t$, is of the form $w_t(a, b) = f(t - t_e) \cdot w(a, b)$, where $f()$ is a decay function. Many different decay functions may make sense, and we designed GraFC2T2 to make it easy to integrate those functions. We consider here the two following classical choices.

- Our first example is the exponential decay function (EDF) illustrated in Figure 6a: $f(x) = e^{-x \cdot \ln(2)/\tau_0}$, where τ_0 is the radioactivity half life; after a delay of τ_0 , the link weight is divided by 2.

- We also consider the logistic decay function (LDF) illustrated in Figure 6b: $f(x) = 1 - 1/(e^{-K(x-\tau_0)} + 1)$ where K is the steepness of the curve and τ_0 is the sigmoid midpoint; if $x = \tau_0$ then $f(x) = 0.5$.

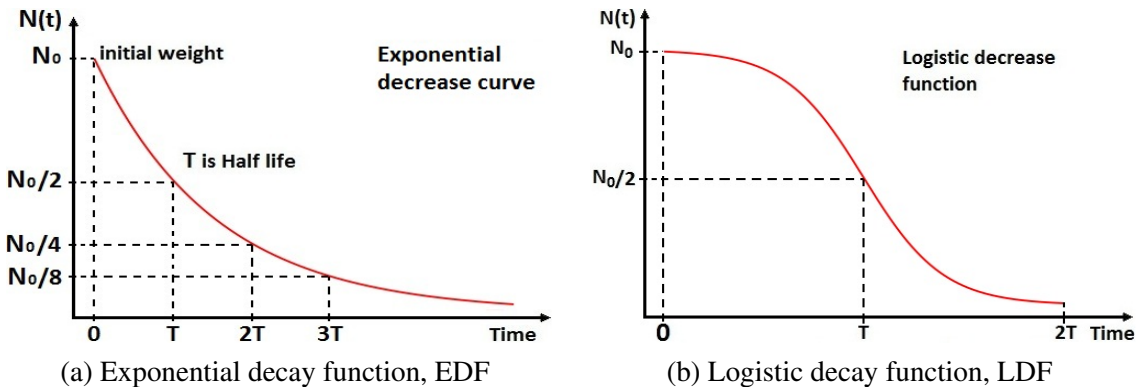


Figure 6: Edge time-weight functions.

IV RECOMMENDATION WITH PERSONALIZED PAGERANK AND TRUST

Once a recommender graph is built with a combination of choices proposed in previous sections, we are ready to perform top- N recommendation from this graph. We present below the personalized PageRank approach and an extension to include the concept of trust between users.

4.1 Personalized PageRank

Personalized PageRank algorithm is defined by [Page et al. \(1999\)](#) for node ranking in graphs so that nodes can be ranked efficiently in order of importance. The first application was on web pages, especially in the Google search engine. Then this algorithm has been widely used in recommender systems because of the good prediction quality obtained ([Gori et al., 2007](#); [Kim and El Saddik, 2011](#); [Şora, 2015](#)).

Following this last observation, [Xiang et al. \(2010\)](#) proposed the Temporal Personalized Random Walk (TPRW) to compute recommendations on STG. It was defined to tackle temporal recommendation using the personalization idea of [Haveliwala \(2002\)](#), corresponding to the following formula:

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

Where PR is PageRank vector that contains the importance of each node at the end of the propagation process that we want to compute; M is the transition matrix of the considered graph; α is the damping factor; and d is the personalization vector indicating which nodes the random walker will jump to after a restart. In other words, d allows to initialize the weight of source nodes. This process favors the recommendation of products that are close to source nodes: items close to source nodes with large weights in vector d , are favored (see below).

For a given user u at time t , we define the personalized temporal vector d as follows, depending on the type of basic graph:

- for BIP, the walker always restarts from u : $d(u) = 1$ and $d(v) = 0$ if $v \neq u$;
- for STG, the walker either restarts from u or from its most recent session node (u, T_k) : $d(u) = \beta$, $d(u, T_k) = 1 - \beta$, $d(v) = 0$ if $v \neq u$ and $v \neq (u, T_k)$;

- for LSG, the walker always restarts from the most recent temporal node representing u , (t', u) : $d(t', u) = 1$ and $d(t'', v) = 0$ if $(t'', v) \neq (t', u)$.

Then, we run PageRank over the recommender graph to compute the interest of each user u for item i at time t , and output the N items with highest interest (in LSG, the interest for item i is the sum of interests for (t, i) , for all t).

4.2 Trust integration

Trust relationships are interesting for improving recommendation, especially for cold users and cold items (users or items for which very limited information is available). Some systems incorporate trust information explicitly specified by users (Jamali and Ester, 2009; Guo *et al.*, 2017; Pan *et al.*, 2017), but since such explicit information is rarely available, several approaches infer implicit trust (Pitsilis and Marshall, 2004; Papagelis *et al.*, 2005; Hwang and Chen, 2007; Lathia *et al.*, 2008). In this section, we describe how to include these both types of trust in our framework.

We assume trust relationships are modeled for each user u by a set TR_u of users trusted by u , and that $trust(u, v)$ gives the trust level of u for all v in TR_u , with $\sum_{v \in TR_u} trust(u, v) = 1$. We denote the method where explicit trust relationships are given by ET (Explicit Trust). We also use an implicit trust metric based on similarity measures as proposed by Papagelis *et al.* (2005) and denote this method by IT (Implicit Trust). In this method, $TR_u = U$ is the set of all users, and $trust(u, v) = |I_u \cap I_v| / |I_u \cup I_v|$ is the Jaccard similarity between users u and v . Note that other similarity measures may be used, such as cosine index.

We then update the personalized temporal vector d definition as follows (with the same notations as in the initial definition above):

- for BIP, $d(u) = 1 - \gamma$, $d(v) = (\gamma \cdot trust(u, v)) / |TR_u|$ if $v \in TR_u$ and $d(v) = 0$ otherwise;
- for STG, we share the jumping probability β between u and its trusted users: $d(u) = \beta \cdot (1 - \gamma)$, $d(v) = (\beta \cdot \gamma \cdot trust(u, v)) / |TR_u|$ for all $v \in TR_u$; and we share the probability $1 - \beta$ between u most recent session node and the ones of trusted users: $d(u, T_k) = (1 - \beta) \cdot (1 - \gamma)$, $d(v, T_v) = (1 - \beta) \cdot \gamma \cdot trust(u, v) / |TR_u|$ where $v \in TR_u$ and (v, T_v) is the most recent session node of v . We set all other entries of d to 0.
- for LSG, $d(t_k, u) = 1 - \gamma$, $d(t_v, v) = \gamma \cdot trust(u, v) / |TR_u|$ if $v \in TR_u$ and (t_v, v) is the most recent node representing v , and all other entries of d are 0.

V EXPERIMENTAL SETUP

Previous sections defined our general graph-based framework GraFC2T2, that gives wide levels of freedom for selecting and combining its various components into a top- N recommender system. These component capture several kinds of side information, in particular content-based, temporal, and trust features. In this section, we describe an experimental setup that we use in the next section to evaluate our framework. This setup consists in two real-world datasets, an evaluation method relying on three metrics, and a parameter selection method to optimize results.

5.1 Datasets

We use publicly available datasets extracted from product reviews Epinions and Ciao³ (Tang *et al.*, 2012), where users can write reviews and give their opinions on a wide category of products like Home, Health, Computers and Media. We model each dataset as a set of review tuples (u, i, c, r, t) meaning that user u has assigned the rating $r \in \{0, 1, 2, 3, 4, 5\}$ to item i at time t , with c being a content-based feature of item i . The explicit trust networks of these datasets are considered such that for each user u , the set TR_u is given for the ET method. Table 1 provides key information on these datasets: start and end dates, as well as numbers of distinct users, items, content-based features, ratings, explicit trust relationships, ratings density and trust relationships density.

	Start date	End date	$\ U\ $	$\ I\ $	$\ C\ $	Ratings	Trust	δ_r	δ_t
Epinions	2010-01-01	2010-12-31	1 843	15 899	24	17 722	4 867	0.06%	0.14%
Ciao	2007-01-01	2010-12-31	879	6 005	6	8 109	23 121	0.15%	3.00%

Table 1: Basic data statistics

Since our framework does not use ratings but only positive links between users and items, we discard all tuples such that the rating it contains is lower than 2.5 or the average rating of involved user.

5.2 Evaluation

Evaluating recommender systems is a difficult task. In this paper, we use three classical metrics for top- N recommendations: F1-score (F1), Hit Ratio (HR) and Mean Average Precision (MAP) (Baeza-Yates and Ribeiro-Neto, 2011). Higher values of these metrics indicate better recommendation performance.

F1-score is a trade-off between ranking precision and recall such that optimizing F1-score is more robust than optimizing precision or recall. Precision is the fraction of good recommendations over all recommended items and recall is the fraction of good recommendations over all relevant items to recommend. For one user u ,

$$Precision = \frac{hit_N(u)}{N}, \quad (2)$$

$$Recall = \frac{hit_N(u)}{I_{new}(u)} \quad (3)$$

and

$$F1 = 2 \cdot \frac{Precision \times Recall}{Precision + Recall} = 2 \cdot \frac{hit_N(u)}{I_{new}(u) + N}, \quad (4)$$

where N is the length of recommendation list, $hit_N(u)$ denotes the number of good recommendations to u in the top- N items and $I_{new}(u)$ is the set of new items to recommend to u . For all users the equation of F1-score is:

$$F@N = \frac{\sum_{u \in U} 2 \times hit_N(u)}{\sum_{u \in U} (I_{new}(u) + N)}. \quad (5)$$

³<https://www.cse.msu.edu/~tangjili/trust.html>

Hit Ratio is the fraction of users to whom the recommender system has made at least one good recommendation over all users:

$$H@N = \frac{\sum_{u \in U} (\text{hit}_N(u) > 0)}{|U|}. \quad (6)$$

Mean Average Precision considers the order of items in the top- N recommendation in order to give better evaluation scores to results that recommend better items first:

$$M@N = \frac{\sum_{u \in U} AP_N(u)}{|U|}, \quad (7)$$

where

$$AP_N(u) = \frac{1}{\text{hit}_N(u)} \sum_{k=1}^N \frac{\text{hit}_k(u)}{k} \times h(k) \quad (8)$$

is the average precision of top- N recommendations done to user u and $h(k) = 1$ if the k -th recommended item is a good recommendation and 0 otherwise.

These metrics evaluate a given top- N recommendation. Since we actually can't perform recommendations on live users, we perform evaluation on past data described above. Following the classical method established by previous works (Li and Tang, 2008; Lathia *et al.*, 2009; Campos *et al.*, 2014; Nzekon Nzeko'o *et al.*, 2017), we partition data according to $k + 1$ time windows of equal duration, and we use them as follow. For each of the k first slices:

- we build recommender graphs that correspond to data of this slice and all previous slices (training set),
- we compute top- N recommendations for users who have selected at least one new item in the next time slice (test set),
- we compute for each evaluation metric M the numerator M_{num_k} and the denominator M_{deno_k} of its definition, given above.

Once we have the values of M_{num_k} and M_{deno_k} of each of the k first windows, we combine them into the Time Averaged (TA) value of the metric under concern:

$$TA(M) = \frac{\sum_k M_{num_k}}{\sum_k M_{deno_k}}. \quad (9)$$

This leads to a time-averaged value of F1-score, Hit ratio and MAP, that we all use for evaluation. Indeed, evaluation metrics can be in disagreement (Gunawardana and Shani, 2009), and so using several metrics is essential to obtain accurate insight on result quality.

In our experiments, we set k to 7 in order to have large enough data slices and meaningful averages. We consider exploring the role of this parameter, as well as the use of more advanced evaluation metrics, as future work.

5.3 Parameter estimation

For each basic graph type, GraFC2T2 defines and implements 27 possible combinations of side information modelings, see Figure 1. Our priority is to explore the behaviors and differences of all these variants, and so we did our best to keep the number of other parameters reasonable. Still, the different version of recommender systems encoded in GraFC2T2 call for several parameter selection.

Exhaustive search for the best values is out of reach, and many subtle techniques exist to explore the parameter space in search for good values. Since this search is not the focus of this paper, we use a simple approach called Randomized Search Cross-Validation (more advanced methods may easily be included in our framework, though) (Bergstra and Bengio, 2012). This method randomly selects parameter values in a predefined set of possible values, usually designed to span well the whole set of values. Here, we use 50 such random settings, sampled in the set defined by Table 2.

	Parameter meaning	Predefined values
Δ	STG session duration	7, 30, 90, 180, 365, 540, 730 days
β	STG long-term preference	0.1, 0.3, 0.5, 0.7, 0.9
τ_0	half life of EDF and LDF	7, 30, 90, 180, 365, 540, 730 days
K	decay slope of LDF	0.1, 0.5, 1, 5, 10, 50, 100
γ	influence of trusted users	0.05, 0.1, 0.15, 0.3, 0.5, 0.7, 0.9
α	damping factor for PageRank	0.05, 0.1, 0.15, 0.3, 0.5, 0.7, 0.9

Table 2: Predefined values of parameters

VI EXPERIMENTAL RESULTS

This section presents extensive experimentations on our GraFC2T2 framework, in order to study its performances in practice, to explore the contribution of each side information in these cases, and to compare obtained results to state-of-the-art recommender systems.

6.1 Performances of GraFC2T2

Table 3 presents the results we obtained for Top-10 item recommendation for Epinions and Ciao datasets. We chose $N = 10$ as for instance Deshpande and Karypis (2004), Xiang *et al.* (2010) and Bernardes *et al.* (2015), and other values we tested gave similar results as one may see in the appendix (Section B). In these tables, each column corresponds to a metric and a basic recommender graph, and each row corresponds to a combination of side information added to this recommender graph. Each cell contains the value of the evaluation metric for the recommender graph made of basic graph in column and side information in row. White color of cell corresponds to the best result and dark color indicates lower performance.

We summarize the insight obtained from these results in Table 4. For each basic recommender graph (vertically) and each evaluation metric (horizontally), we selected the three recommender graphs that achieve the best performances and we display on the corresponding row the performances obtained on the basic graph (without side information), the best obtained performances (with side information), the improvement percentage, and the name of the corresponding version of recommender graph with side information.

All best improvements thanks to side information in GraFC2T2 are at least 46% for Epinions and at least 41% for Ciao. Table 4 also shows that the best combination of side information

EPINIIONS	F1@10			HR@10			MAP@10		
	BIP	STG	LSG	BIP	STG	LSG	BIP	STG	LSG
-	2.18	2.0	1.14	5.17	4.77	4.24	2.23	2.17	1.71
ET	1.81	1.86	1.14	4.64	4.64	4.11	2.04	2.07	1.66
IT	2.17	2.42	1.61	5.44	5.44	5.31	2.29	2.34	2.24
EDF	3.74	3.32	2.25	6.1	5.97	5.7	2.31	2.35	2.32
LDF	3.16	2.63	2.26	5.97	5.31	5.97	2.24	2.09	2.54
CI	2.53	3.0	0.86	5.97	6.23	3.32	2.48	2.73	1.74
CIU	3.29	3.51	0.66	6.37	6.63	2.79	2.66	2.88	1.66
EDF-ET	2.77	3.32	1.77	5.44	5.84	5.04	2.13	1.97	2.09
EDF-IT	3.88	4.43	2.74	7.03	7.16	6.37	2.64	2.57	2.42
LDF-ET	2.12	2.63	1.58	4.91	5.31	4.77	2.03	2.1	2.13
LDF-IT	3.1	2.77	3.29	6.1	6.5	7.16	2.98	2.98	3.01
CI-ET	2.44	3.03	0.92	5.84	6.23	3.45	2.28	2.67	1.74
CI-IT	3.12	3.14	1.77	6.37	6.37	5.44	2.49	2.66	2.15
CIU-ET	3.03	3.51	0.66	6.1	6.63	2.79	2.43	2.9	1.66
CIU-IT	3.89	3.72	1.99	7.03	7.03	5.44	2.79	2.88	2.25
CI-EDF	4.88	4.84	0.91	7.69	6.9	3.32	3.06	2.73	1.74
CI-LDF	4.91	3.56	1.1	7.03	6.5	3.98	2.61	2.67	1.75
CIU-EDF	4.51	6.48	0.7	7.69	7.69	2.92	3.23	3.03	1.66
CIU-LDF	5.29	4.49	0.9	7.16	6.76	3.58	2.89	2.85	1.72
CI-EDF-ET	3.84	4.73	0.98	6.63	6.76	3.45	2.44	2.46	1.74
CI-EDF-IT	4.85	4.47	1.69	7.43	6.76	5.31	2.8	2.7	2.17
CI-LDF-ET	3.02	3.39	0.92	5.97	6.37	3.45	2.28	2.66	1.74
CI-LDF-IT	3.81	4.02	3.41	6.76	6.63	7.29	3.0	3.0	2.66
CIU-EDF-ET	4.75	6.13	0.7	7.16	7.43	2.92	2.64	2.95	1.66
CIU-EDF-IT	6.34	7.66	1.99	7.82	7.96	5.44	3.32	3.18	2.27
CIU-LDF-ET	4.06	3.78	0.7	6.63	6.63	2.92	2.48	2.78	1.66
CIU-LDF-IT	5.69	4.49	3.68	7.82	7.03	7.03	3.18	3.07	3.17

CIAO	F1@10			HR@10			MAP@10		
	BIP	STG	LSG	BIP	STG	LSG	BIP	STG	LSG
-	1.18	1.48	1.45	5.26	5.63	6.53	1.9	1.99	2.24
ET	1.08	1.44	1.5	5.08	5.44	6.72	1.74	1.9	2.17
IT	2.18	1.92	2.25	7.62	7.62	7.26	2.39	2.39	2.28
EDF	1.63	1.7	2.0	6.35	5.99	7.44	2.03	2.26	2.34
LDF	2.02	1.74	3.27	7.26	7.44	9.26	2.63	2.84	3.51
CI	1.25	2.14	1.25	6.53	6.35	5.26	2.04	2.14	1.7
CIU	2.38	4.56	1.24	7.08	8.53	4.9	2.39	3.01	1.48
EDF-ET	1.17	1.47	1.41	5.26	5.81	6.53	1.68	2.1	2.18
EDF-IT	2.13	3.08	2.37	9.98	9.44	7.26	3.04	2.84	2.32
LDF-ET	1.18	1.49	1.5	5.44	5.63	6.72	1.81	2.01	2.18
LDF-IT	2.66	2.76	2.76	8.53	8.53	8.71	2.91	2.96	2.66
CI-ET	1.39	1.66	1.51	5.99	6.17	5.63	1.97	2.06	1.72
CI-IT	2.41	2.25	2.84	7.8	7.8	8.53	2.47	2.42	2.43
CIU-ET	2.02	4.0	1.31	6.9	8.53	5.08	2.27	3.12	1.5
CIU-IT	2.76	4.21	2.96	8.53	8.53	8.35	2.82	2.96	2.44
CI-EDF	2.58	3.15	1.66	7.62	8.17	6.17	2.33	2.76	1.77
CI-LDF	3.38	2.91	2.37	8.35	8.89	7.62	2.87	2.86	2.85
CIU-EDF	3.46	4.46	1.44	8.71	9.26	5.44	3.24	3.29	1.65
CIU-LDF	7.74	5.79	2.11	9.98	9.8	6.9	3.46	3.31	2.68
CI-EDF-ET	1.86	2.11	1.63	7.26	6.72	5.99	2.27	2.37	1.81
CI-EDF-IT	3.42	3.54	2.9	9.8	9.8	7.99	3.09	3.14	2.46
CI-LDF-ET	1.66	2.67	1.74	6.35	7.08	5.81	1.98	2.12	1.77
CI-LDF-IT	4.56	4.89	2.64	10.7	11.3	8.53	3.19	3.16	3.16
CIU-EDF-ET	2.69	4.79	1.53	8.53	9.26	5.63	2.76	3.09	1.65
CIU-EDF-IT	4.62	5.1	2.88	10.3	10.5	8.35	3.32	3.37	2.38
CIU-LDF-ET	2.73	6.42	1.45	8.35	9.98	5.44	2.29	3.34	1.64
CIU-LDF-IT	5.07	6.11	2.51	11.1	11.1	9.07	3.34	3.35	3.18

Table 3: Epinions and Ciao - Performance with optimal settings. Each cell contains the value of an evaluation metric for the recommender graph made of basic graph in column and side information in row. White color of cell corresponds to the best result and dark color indicates lower performance

Epinions Dataset													
	No	BIP				STG				LSG			
		Basic	Best	Imp.	BIP-Best	Basic	Best	Imp.	STG-Best	Basic	Best	Imp.	LSG-Best
F@10	1	2.18	6.34	190%	CIU-EDF-IT	2.0	7.66	282%	CIU-EDF-IT	1.14	3.68	221%	CIU-LDF-IT
	2	2.18	5.69	160%	CIU-LDF-IT	2.0	6.48	223%	CIU-EDF	1.14	3.41	197%	CI-LDF-IT
	3	2.18	5.29	142%	CIU-LDF	2.0	6.13	206%	CIU-EDF-ET	1.14	3.29	187%	LDF-IT
H@10	1	5.17	7.82	51%	CIU-EDF-IT	4.77	7.96	66%	CIU-EDF-IT	4.24	7.29	71%	CI-LDF-IT
	2	5.17	7.82	51%	CIU-LDF-IT	4.77	7.69	61%	CIU-EDF	4.24	7.16	68%	LDF-IT
	3	5.17	7.69	48%	CI-EDF	4.77	7.43	55%	CIU-EDF-ET	4.24	7.03	65%	CIU-LDF-IT
M@10	1	2.23	3.32	48%	CIU-EDF-IT	2.17	3.18	46%	CIU-EDF-IT	1.71	3.17	85%	CIU-LDF-IT
	2	2.23	3.23	45%	CIU-EDF	2.17	3.07	41%	CIU-LDF-IT	1.71	3.01	76%	LDF-IT
	3	2.23	3.18	42%	CIU-LDF-IT	2.17	3.03	39%	CIU-EDF	1.71	2.66	55%	CI-LDF-IT

Ciao Dataset													
	No	BIP				STG				LSG			
		Basic	Best	Imp.	BIP-Best	Basic	Best	Imp.	STG-Best	Basic	Best	Imp.	LSG-Best
F@10	1	1.18	7.74	556%	CIU-LDF	1.48	6.42	332%	CIU-LDF-ET	1.45	3.27	125%	LDF
	2	1.18	5.07	330%	CIU-LDF-IT	1.48	6.11	311%	CIU-LDF-IT	1.45	2.96	104%	CIU-IT
	3	1.18	4.62	291%	CIU-EDF-IT	1.48	5.79	290%	CIU-LDF	1.45	2.9	99%	CI-EDF-IT
H@10	1	5.26	11.1	110%	CIU-LDF-IT	5.63	11.3	100%	CI-LDF-IT	6.53	9.26	41%	LDF
	2	5.26	10.7	103%	CI-LDF-IT	5.63	11.1	96%	CIU-LDF-IT	6.53	9.07	38%	CIU-LDF-IT
	3	5.26	10.3	96%	CIU-EDF-IT	5.63	10.5	87%	CIU-EDF-IT	6.53	8.71	33%	LDF-IT
M@10	1	1.9	3.46	82%	CIU-LDF	1.99	3.37	69%	CIU-EDF-IT	2.24	3.51	57%	LDF
	2	1.9	3.34	76%	CIU-LDF-IT	1.99	3.35	68%	CIU-LDF-IT	2.24	3.18	42%	CIU-LDF-IT
	3	1.9	3.32	74%	CIU-EDF-IT	1.99	3.34	67%	CIU-LDF-ET	2.24	3.16	41%	CI-LDF-IT

Table 4: Best recommender graphs - Comparison of the three best recommender graph combinations with the associated basic graph. We display the obtained improvement percentage.

for Epinions is CIU-EDF-IT for BIP and STG basic graphs and CIU-LDF-IT for LSG basic graph. For Ciao, good results are obtained with CIU-LDF-IT for all basic graphs. These results clearly confirm the relevance of graphs extended simultaneously with content, time and trust information.

6.2 Impact of side information

We now give details on the impact of side information and their combination in GraFC2T2. This is context dependent, as observed behaviors vary with datasets; one may however easily test the GraFC2T2 framework with his/her own datasets and discover the best choices for the case under concern. The discussion provided here is mostly an illustration of this.

When we consider the basic graphs with no side information, in the case of Epinions, BIP gives the best results for all evaluation metrics. Instead, LSG gives the best Hit ratio and MAP, while STG gives the best F1-score in the case of Ciao.

If we include only one kind of side information, we observe that explicit trust (ET) does not improve the results, but implicit trust (IT) does for all basic graphs. The insertion of time-weight always produces improvements. Finally, content-based features increase performances for BIP and STG but not for LSG. For Epinions, the best graph with one kind of side information is BIP-EDF in F1-score and STG-CIU in Hit ratio and MAP. In Ciao, the best one is LSG-LDF in Hit ratio and MAP, and STG-CIU is the best in F1-score. This shows that the impact of a unique kind of side information highly depends on the basic graph and on the data.

Recommendations using two kinds of side information perform significantly better than with only one kind of side information. For instance, in the Epinions case, performances increase

from 3.74% to 6.48% in F1-score, from 6.63% to 7.69% in Hit ratio and from 2.88% to 3.23% in MAP. Combining time-weight with implicit trust performs better than time-weight and trust taken separately. Similarly, combining content-based features with implicit trust is better than content-based features or trust taken separately, but generally less interesting than combining time-weight and implicit trust. Combining content-based features and time-weight usually produces better improvements for BIP and STG but no improvement for LSG. In Epinions, BIP-CI-EDF and BIP-CIU-EDF perform best. In Ciao, BIP-CIU-LDF is always better. This confirms the relevance of graphs that integrate content-based features and time, like time-weight content-based STG proposed by [Nzekon Nzeko'o et al. \(2017\)](#).

Using three kinds of side information does not greatly improve the best performances achieved with two kinds of side information. For instance, in Epinions, the performances increase from 6.48 to 7.66% in F1-score, from 7.69 to 7.96% in Hit ratio and 3.23 to 3.32% in MAP. Nevertheless, Table 4 shows that recommender graphs with three kinds of side information are by far the most frequent among the best ones. For this reason, we recommend the use of content-based, time and trust information simultaneously in order to increase the chances to achieve good results.

6.3 Best values of parameters

In this section, we focus only on recommender graphs with CIU-EDF-IT and CIU-LDF-IT combination that are most common in the best performance in Table 4. We have made the following observations:

- In Epinions dataset, for the combination CIU-EDF-IT, $\Delta = 7$, $\beta = 0.5$, $\tau_0 = 90$ for BIP and STG and 180 for LSG, $\gamma \in \{0.15, 0.3\}$ for BIP and STG and 0.9 for LSG, and $\alpha = 0.9$. For the combination CIU-LDF-IT, $\Delta = 365$, $\beta = 0.7$, $\tau_0 \in \{30, 90\}$ for BIP and STG and 7 for LSG, $K = 0.5$ for BIP, 100 for STG and 5 for LSG, $\gamma \in \{0.1, 0.15\}$ for BIP and STG and 0.9 for LSG, and $\alpha \in \{0.7, 0.9\}$;
- In Ciao dataset, for the combination CIU-EDF-IT, $\Delta = 180$, $\beta = 0.3$, $\tau_0 = 180$, $\gamma = 0.9$ and $\alpha = 0.9$. For the combination CIU-LDF-IT, $\Delta = 540$, $\beta = 0.1$, $\tau_0 = 365$ for BIP and STG and 180 for LSG, $K = 10$ for BIP and STG and 100 for LSG, $\gamma \in \{0.7, 0.9\}$, and $\alpha = 0.9$;

The values of these parameters indicate that in Epinions, the weights of the data used (edge weights) decrease faster than in Ciao; τ_0 is small in Epinions $\{7, 30, 90\}$ and is larger in Ciao $\{180, 365\}$. Regarding trust, γ is still high in Ciao $\{0.7, 0.9\}$ and is smaller in Epinions $\{0.1, 0.15, 0.3\}$ which shows that the influence of implicit trust is more important in Ciao. However, this influence must always be great for the graph LSG $\{0.9\}$ in all datasets.

6.4 Comparison with state-of-the-art systems without side information

We now compare the performances of GraFC2T2 with those of some state-of-the-art top- N recommender systems that don't take into account side information. The considered models are: Most-Popular-Item (MPI) that computes the ranking score of an item by its popularity; the ranking oriented collaborative filtering, user-based (UBCF) and item-based (IBCF) collaborative filtering ([Karypis, 2001](#); [McLaughlin and Herlocker, 2004](#)); some recommender systems for positive implicit feedback scenarios, Bayesian Personalized Ranking (BPR) ([Rendle et al., 2009](#)), Sparse linear methods for top- N recommender systems (SLIM) ([Ning and Karypis, 2011](#)), collaborative less-is-more filtering (CLiMF) ([Shi et al., 2012](#)) and Matrix factorization with Alternating Least Squares (ALS) ([Hu et al., 2008](#)).

We use Randomized Search Cross-Validation to have good performances of the considered recommender systems. For UBCF and IBCF models, 10 settings are generated such that the neighborhood size $k \in \{10, 20, 30, 40, 50, 80, 100, 150, 200, 500\}$. For BPR, SLIM, CLIMF and ALS models, 50 settings are generated such that the number of latent factors $l \in \{10, 20, 30, 50, 100, 200, 500\}$, learning rate and all regularization bias are taken in $\{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05\}$.

Table 5 presents the best results obtained for these recommender systems that don't take into account side information and those obtained with our framework. This shows that GraFC2T2 outperforms these systems and illustrates the relevance of a general framework in which various kinds of side information can be added to improve recommendations.

		MPI	UBCF	IBCF	BPR	SLIM	CLIMF	ALS	GraFC2T2
Epinions	F@10	1.79	0.30	0.70	0.15	0.82	1.97	2.27	7.66
	H@10	4.91	1.46	2.79	0.80	2.92	5.17	4.91	7.96
	M@10	2.07	0.61	1.29	0.45	1.16	2.15	2.26	3.32
Ciao	F@10	2.26	0.31	0.94	0.22	1.49	3.38	2.10	7.74
	H@10	7.62	1.63	4.17	1.27	5.08	8.71	6.90	11.3
	M@10	2.62	0.59	1.65	0.56	2.09	3.06	2.46	3.51

Table 5: Experiment results on Epinions and Ciao datasets for Top-10. Performances are given in percentage and best ones are highlighted in bold.

A future step in our research is to compare the results obtained by GraFC2T2 to those produced by state-of-the-art systems that include side information. We have already observed that the results produced by GraFC2T2 are comparable to those presented by [Xiao et al. \(2017\)](#) where both item and social visibilities are modeled. Moreover, we have also made a comparison with Trust aware Denoising Auto Encoder (TDAE) technique based on deep learning ([Pan et al., 2017](#)). The results for Epinions (M@10 = 1.32%) and Ciao (M@10 = 3.07%) confirm the relevance of GraFC2T2.

Notice that the most basic, non-personalized approach MPI is able to achieve better results compared to BPR, SLIM, UBCF and IBCF. This indicates that users tend to consume popular items. This is not the first work in which MPI is better than BPR or other matrix factorization models, [Zhao et al. \(2014\)](#) and [Guo et al. \(2017\)](#) have made the same observation.

VII RELATED WORK

As we already said, many contributions improve collaborative filtering (CF) recommender systems with the inclusion of side information, and we used several ideas proposed in these previous works. In the rest of this section, we shortly review key related references.

7.1 Trust-based recommender systems

CF usually suffers from data sparsity and cold start problems, which may be solved in part with user trust. For instance, [Papagelis et al. \(2005\)](#) used trust inference by transitive associations between users in a social network. [Ma et al. \(2017\)](#) use explicit trust and distrust to improve clustering-based CF recommendation, while [Guo et al. \(2014\)](#) merge ratings of trusted neighbors to infer probable preferences of other users, and identify similar users for item recommendations.

In some cases, trust can be explicitly provided by users, as do Massa and Avesani (2007), but in other ones, this information is not given and it can be inferred from user behaviors. For example, (Papagelis *et al.*, 2005) use Pearson correlation to compute implicit trust using ratings dataset and in cases where there is only implicit data, measure like Jaccard and Cosine can be used. In other works, trust enhancement is done by trust propagation on trust network where the weight of an link (u, u') is the trust of u to u' Deng *et al.* (2014).

Note that work on influencers can also be considered here, as there is a trust relationship between influencers and their followers (Liu *et al.*, 2015; Grafström *et al.*, 2018). Our framework is able to integrate the impact of influencers in the same way as trust between users. The main difference is who influences who and how much. Once you have the answers to these questions, the customization of PageRank is done according to these answers. The impact of influencers or influencer-based recommendation is not studied in this work, but it is a good issue for future work.

The concept of influence is a good example of other side information that may be included in our system (Liu *et al.*, 2015; Grafström *et al.*, 2018). Similarly to trust (although these two concepts are different) influence may be used to customize PageRank, once it is correctly quantified. For instance, influence may be seen as a trust relationship between influencers and their followers.

7.2 Time aware recommender systems

Most recommender systems that take temporal aspects into account are based on concept drift: older information is less important than recent information for predicting future user purchases. For this reason, Ding and Li (2005) proposed the use of the time-weight decay functions we used in this paper, in order to assign greater weight to the most recent ratings in similarity computations. In addition, Gaillard and Renders (2015) propose an incremental matrix completion method, that automatically allows the factors related to both users and items to adapt "on-line" to concept drift hypothesis. Going further, Liu *et al.* (2010) propose an online incremental CF in which a decay function is used for similarity computations and another one is used for rating prediction. Time-weight functions are also used in other studies as by Koren (2009); Karahodža *et al.* (2015) and Nzekon Nzeko'o *et al.* (2017).

Other approaches to concept drift assume that the importance of information used for recommendations is ephemeral, e.g. Lathia *et al.* (2009) divide time into slices and use data only within a single slice. Such recommender systems therefore focus on user short-term preferences. It however seems that some preferences are stable and persist over time, and so that old information should also be included. For this reason, some works (Xiang *et al.*, 2010; Li *et al.*, 2007) capture both short-term preferences and long-term preferences and combine them in the recommendation process. For example, Xiang *et al.* (2010) propose STG to incorporate temporal aspects by separately modeling long-term preferences and short-term preferences within a graph model.

7.3 Content-based recommender systems

These systems aim at recommending items similar to the ones the user liked in the past. A way to achieve this, developed by Lops *et al.* (2011), is to match features associated to user preferences with those of items. Then, recommendation is performed in three steps: extracting relevant features from items, build user preference profiles based on item features, and finally select new items that fit user preferences. This approach is used in several domains such as recommendation of books (Mooney and Roy, 2000) and recommendation of web pages (Pazzani *et al.*, 1996).

Using content-based features may improve CF techniques by allowing more details on user favorite item features and increase the possibility to reach items that have not been selected in the past by other users. Some works indeed show that these hybrid recommender systems solve weaknesses of both approaches (Balabanović and Shoham, 1997; Basu *et al.*, 1998; Burke, 2002).

Recent work on content-based approaches are dedicated to the Social Book Search (SBS). The SBS Lab investigates book search in scenarios where users search with more than just a query, and look for more than objective metadata. It has two tracks. The first one is a Suggestion Track aiming at developing test collections for evaluating ranking effectiveness of book retrieval and recommender systems. The second one is an Interactive Track aimed at developing user interfaces that support users through each stage during complex search tasks and to investigate how users exploit professional metadata and user-generated content (Koolen *et al.*, 2015).

7.4 Graph-based recommender systems

The simplest graph-based recommender system rely on the classical bipartite graph (BIP) in which only user-item links are used. Most used algorithms are based on random walk (Baluja *et al.*, 2008), like Injected Preference Fusion (Xiang *et al.*, 2010) and PageRank which is used in this paper; they compute a probability to reach items from the user under concern, and recommend the ones with highest probability.

Graph-based systems may be seen as CF systems, and so one may use the same idea as in hybrid recommender systems to improve them (Burke, 2002). Nguyen *et al.* (2008) achieve this by adding a third node type: content nodes. The resulting graph ignores temporal aspects, though. To improve this, Yu *et al.* (2014) propose the Topic-STG which incorporate content-based features and the temporal dynamic of STG. However these graphs handle each link regardless of its age, which contradicts the concept drift assumption. This is why we propose the Time-weight and content-based STG (Nzekon Nzeko'o *et al.*, 2017), where old links have a lower weight than recent ones. Up to our knowledge, none of these graph-based works takes advantage of content-based, time and trust information simultaneously.

We note that, despite the fact that recommender graphs are not much studied compared to model-based techniques such as matrix factorization or neural networks, they remain relevant. For example Pixie recommender system proposed by Eksombatchai *et al.* (2018) is the recent scalable graph-based real-time system developed and deployed at Pinterest. Given a set of user-specific pins as a query, Pixie selects in real-time from billions of possible pins that are most related to the query. To generate recommendations, Eksombatchai *et al.* develop Pixie Random Walk algorithm that uses the Pinterest object graph of 3 billion nodes and 17 billion edges. This has been made possible thanks to the technological evolution of Random Access Memories.

CONCLUSION

Our main goal with this paper was to show that including several side information improves the quality of recommender graphs built for top- N recommendation task. For this purpose, we designed and implemented GraFC2T2, a recommender graph framework which makes it easy to explore various approaches for modeling and combining many features of interests for recommendation. In particular, GraFC2T2 extends classical bipartite graphs, session-based temporal graphs and link stream graphs by integrating content-based features, time-weight functions, and user trust into a personalized PageRank system.

The experiments we conducted on Epinions and Ciao datasets with F1-score, Hit ratio and MAP evaluation metrics show that best performances are always reached by graphs that integrate at least two side information and that graphs with time-weight always outperform the others. The resulting improvements are of at least 41%. Moreover, comparison with state-of-the-art matrix factorization and classical user-based and item-based collaborative filtering methods confirms the relevance of GraFC2T2 framework for top- N recommendation. Good improvements obtained in recommender graphs by integration of side information do not guarantee such improvement for other types of recommender systems such as matrix factorization and neural network. We therefore consider inclusion of content-based, time and trust information simultaneously in such system as a key perspective.

References

- Adomavicius G., Tuzhilin A. (2005). [Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions](#). *IEEE Transactions on Knowledge & Data Engineering* 17(6), 734–749. doi:10.1109/TKDE.2005.99.
- Baeza-Yates R. A., Ribeiro-Neto B. (2011). *Modern information retrieval*. Addison-Wesley.
- Balabanović M., Shoham Y. (1997). [Fab: content-based, collaborative recommendation](#). *Communications of the ACM* 40(3), 66–72. doi:10.1145/245108.245124.
- Baluja S., Seth R., Sivakumar D., Jing Y., Yagnik J., Kumar S., Ravichandran D., Aly M. (2008). [Video suggestion and discovery for youtube: taking random walks through the view graph](#). In *17th international conference on World Wide Web*, pp. 895–904. ACM. doi:10.1145/1367497.1367618.
- Basu C., Hirsh H., Cohen W. (1998). [Recommendation as classification: Using social and content-based information in recommendation](#). In *Aaai/iaai*, pp. 714–720.
- Bergstra J., Bengio Y. (2012). [Random search for hyper-parameter optimization](#). *Journal of Machine Learning Research* 13, 281–305.
- Bernardes D., Diaby M., Fournier R., FogelmanSoulié F., Viennet E. (2015). [A social formalism and survey for recommender systems](#). *ACM SIGKDD Explorations Newsletter* 16(2), 20–37. doi:10.1145/2783702.2783705.
- Burke R. (2002). [Hybrid recommender systems: Survey and experiments](#). *User modeling and user-adapted interaction* 12(4), 331–370. doi:10.1023/A:1021240730564.
- Campos P. G., Díez F., Cantador I. (2014). [Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols](#). *User Modeling and User-Adapted Interaction* 24(1-2), 67–119. doi:10.1007/s11257-012-9136-x.
- Chen Y., Zhao X., Gan J., Ren J., Hu Y. (2016). [Content-based top-N recommendation using heterogeneous relations](#). In *Australasian Database Conference*, Volume 9877 of *Lecture Notes in Computer Science*, pp. 308–320. Springer. doi:10.1007/978-3-319-46922-5_24.
- Christakopoulou E., Karypis G. (2016). [Local item-item models for top-n recommendation](#). In *10th ACM Conference on Recommender Systems*, pp. 67–74. ACM. doi:10.1145/2959100.2959185.
- Cremonesi P., Koren Y., Turrin R. (2010). [Performance of recommender algorithms on top-n recommendation tasks](#). In *4th ACM conference on Recommender systems*, pp. 39–46. ACM. doi:10.1145/1864708.1864721.

- Deng S., Huang L., Xu G. (2014). [Social network-based service recommendation with trust enhancement](#). *Expert Systems with Applications* 41(18), 8075–8084. doi:10.1016/j.eswa.2014.07.012.
- Deshpande M., Karypis G. (2004). [Item-based top-n recommendation algorithms](#). *ACM Transactions on Information Systems (TOIS)* 22(1), 143–177. doi:10.1145/963770.963776.
- Ding Y., Li X. (2005). [Time weight collaborative filtering](#). In *14th ACM international conference on Information and knowledge management*, pp. 485–492. ACM. doi:10.1145/1099554.1099689.
- Eksombatchai C., Jindal P., Liu J. Z., Liu Y., Sharma R., Sugnet C., Ulrich M., Leskovec J. (2018). [Pixie: A system for recommending 3+ billion items to 200+ million users in real-time](#). In *World Wide Web Conference on World Wide Web*, pp. 1775–1784. International World Wide Web Conferences Steering Committee. doi:10.1145/3178876.3186183.
- Gaillard J., Renders J.-M. (2015). [Time-sensitive collaborative filtering through adaptive matrix completion](#). In *European Conference on Information Retrieval*, Volume 9022 of *Lecture Notes in Computer Science*, pp. 327–332. Springer. doi:10.1007/978-3-319-16354-3_35.
- Gori M., Pucci A., Roma V., Siena I. (2007). [ItemRank: A Random-Walk Based Scoring Algorithm for Recommender Engines](#). In *20th international joint conference on Artificial intelligence*, Volume 7, pp. 2766–2771.
- Grafström J., Jakobsson L., Wiede P. (2018). [The Impact of Influencer Marketing on Consumers' Attitudes](#). Technical report, Jnkping University.
- Gunawardana A., Shani G. (2009). [A survey of accuracy evaluation metrics of recommendation tasks](#). *Journal of Machine Learning Research* 10, 2935–2962.
- Guo G., Zhang J., Thalmann D. (2014). [Merging trust in collaborative filtering to alleviate data sparsity and cold start](#). *Knowledge-Based Systems* 57, 57–68. doi:10.1016/j.knosys.2013.12.007.
- Guo G., Zhang J., Zhu F., Wang X. (2017). [Factored similarity models with social trust for top-N item recommendation](#). *Knowledge-Based Systems* 122, 17–25. doi:10.1016/j.knosys.2017.01.027.
- Haveliwala T. H. (2002). [Topic-sensitive PageRank](#). In *11th international conference on World Wide Web*, pp. 517–526. ACM. doi:10.1145/511446.511513.
- Herlocker J. L., Konstan J. A., Borchers A., Riedl J. (1999). [An algorithmic framework for performing collaborative filtering](#). In *22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 230–237. ACM. doi:10.1145/312624.312682.
- Hu Y., Koren Y., Volinsky C. (2008). [Collaborative filtering for implicit feedback datasets](#). In *8th IEEE International Conference on Data Mining*, pp. 263–272. Ieee. doi:10.1109/ICDM.2008.22.
- Huang Z., Chen H., Zeng D. (2004). [Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering](#). *ACM Transactions on Information Systems* 22(1), 116–142. doi:10.1145/963770.963775.
- Hwang C.-S., Chen Y.-P. (2007). [Using trust in collaborative filtering recommendation](#). In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, Volume 4570 of *Lecture Notes in Computer Science*, pp. 1052–1060. Springer. doi:10.1007/978-3-540-73325-6_105.
- Jamali M., Ester M. (2009). [Using a trust network to improve top-N recommendation](#). In *3rd ACM conference on Recommender systems*, pp. 181–188. ACM. doi:10.1145/1639714.1639745.
- Karahodža B., Donko D., Šupić H. (2015). [Temporal dynamics of changes in group user's preferences in recommender systems](#). In *38th International Convention on Information and Communication Technology, Electronics and Microelectronics*, pp. 1262–1266. IEEE. doi:10.1109/MIPRO.2015.7160469.
- Karypis G. (2001). [Evaluation of item-based top-n recommendation algorithms](#). In *10th international conference on Information and knowledge management*, pp. 247–254. ACM.
- Kim H.-N., El Saddik A. (2011). [Personalized PageRank vectors for tag recommendations: inside FolkRank](#). In *5th ACM conference on Recommender systems*, pp. 45–52. ACM. doi:10.1145/2043932.2043945.
- Konstan J. A., Miller B. N., Maltz D., Herlocker J. L., Gordon L. R., Riedl J. (1997). [GroupLens: applying collaborative filtering to Usenet news](#). *Communications of the ACM* 40(3), 77–87. doi:10.1145/245108.245126.
- Koolen M., Bogers T., Gäde M., Hall M., Huurdeman H., Kamps J., Skov M., Toms E., Walsh D. (2015). [Overview of the CLEF 2015 social book search lab](#). In *International conference of the cross-language evaluation forum for European languages*, Volume 9283 of *Lecture Notes in Computer Science*, pp. 545–564. doi:10.1007/978-

- Koren Y. (2009). Collaborative filtering with temporal dynamics. In *15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 447–456. doi:10.1145/1557019.1557072.
- Koren Y., Bell R., Volinsky C. (2009). Matrix factorization techniques for recommender systems. *Computer* 42(8), 30–37. doi:10.1109/MC.2009.263.
- Latapy M., Viard T., Magnien C. (2018). Stream graphs and link streams for the modeling of interactions over time. *Social Network Analysis and Mining* 8, 61. doi:10.1007/s13278-018-0537-7.
- Lathia N., Hailes S., Capra L. (2008). Trust-based collaborative filtering. In *IFIP International Conference on Trust Management*, Volume 263 of *IFIP The International Federation for Information Processing*, pp. 119–134. Springer. doi:10.1007/978-0-387-09428-1_8.
- Lathia N., Hailes S., Capra L. (2009). Temporal collaborative filtering with adaptive neighbourhoods. In *32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 796–797. ACM. doi:10.1145/1571941.1572133.
- Li L., Yang Z., Wang B., Kitsuregawa M. (2007). Dynamic adaptation strategies for long-term and short-term user profile to personalize search. In *Asia-Pacific Web Conference / International Conference on Web-Age Information Management*, Volume 4505 of *Lecture Notes in Computer Science*, pp. 228–240. Springer. doi:10.1007/978-3-540-72524-4_26.
- Li Y., Tang J. (2008). Expertise search in a time-varying social network. In *9th International Conference on Web-Age Information Management*, pp. 293–300. IEEE. doi:10.1109/WAIM.2008.100.
- Liu N. N., Zhao M., Xiang E., Yang Q. (2010). Online evolutionary collaborative filtering. In *4th ACM conference on Recommender systems*, pp. 95–102. ACM. doi:10.1145/1864708.1864729.
- Liu S., Jiang C., Lin Z., Ding Y., Duan R., Xu Z. (2015). Identifying effective influencers based on trust for electronic word-of-mouth marketing: A domain-aware approach. *Information sciences* 306, 34–52. doi:10.1016/j.ins.2015.01.034.
- Lops P., De Gemmis M., Semeraro G. (2011). Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, Chapter 3, pp. 73–105. Springer. doi:10.1007/978-0-387-85820-3_3.
- Ma X., Lu H., Gan Z., Zeng J. (2017). An explicit trust and distrust clustering based collaborative filtering recommendation approach. *Electronic Commerce Research and Applications* 25, 29–39. doi:10.1016/j.elerap.2017.06.005.
- Massa P., Avesani P. (2007). Trust-aware recommender systems. In *ACM Conference on Recommender systems*, pp. 17–24. ACM. doi:10.1145/1297231.1297235.
- McLaughlin M. R., Herlocker J. L. (2004). A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In *27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 329–336. ACM. doi:10.1145/1008992.1009050.
- Mooney R. J., Roy L. (2000). Content-based book recommending using learning for text categorization. In *5th ACM conference on Digital libraries*, pp. 195–204. ACM. doi:10.1145/336597.336662.
- Nguyen D. P., Le Q. T., Tu M. P. (2008). A graph-based method for combining collaborative and content-based filtering. In *Pacific Rim International Conference on Artificial Intelligence*, pp. 859–869. Springer. doi:10.1007/978-3-540-89197-0_80.
- Ning X., Karypis G. (2011). Slim: Sparse linear methods for top-n recommender systems. In *11th IEEE International Conference on Data Mining*, pp. 497–506. IEEE. doi:10.1109/ICDM.2011.134.
- Ning X., Karypis G. (2012). Sparse linear methods with side information for top-n recommendations. In *6th ACM conference on Recommender systems*, pp. 155–162. ACM. doi:10.1145/2365952.2365983.
- Nzekon Nzeko'o A. J., Tchunte M., Latapy M. (2017). Time Weight Content-Based Extensions of Temporal Graphs for Personalized Recommendation. In *13th International Conference on Web Information Systems and Technologies*.
- Nzekon Nzeko'o A. J., Tchunte M., Latapy M. (2019). Link Stream Graph for Temporal Recommendations. In *Colloquium of Mathematics and Computer Science*.
- Page L., Brin S., Motwani R., Winograd T. (1999). The PageRank citation ranking: Bringing order to the web.

Technical report, Stanford InfoLab.

- Pan Y., He F., Yu H. (2017). [Trust-aware Collaborative Denoising Auto-Encoder for Top-N Recommendation](#). *arXiv cs.IR*, 1703.01760.
- Papagelis M., Plexousakis D., Kutsuras T. (2005). [Alleviating the sparsity problem of collaborative filtering using trust inferences](#). In *Trust management*, Volume 3477 of *Lecture Notes in Computer Science*, pp. 224–239. Springer. doi:10.1007/11429760_16.
- Pazzani M. J., Muramatsu J., Billsus D. (1996). [Syskill & Webert: Identifying interesting web sites](#). In *13th AAAI national conference on Artificial intelligence*, pp. 54–61.
- Pitsilis G., Marshall L. F. (2004). [A model of trust derivation from evidence for use in recommendation systems](#). Technical report, University of Newcastle upon Tyne, School of Computing Science.
- Rendle S., Freudenthaler C., Gantner Z., Schmidt-Thieme L. (2009). [BPR: Bayesian personalized ranking from implicit feedback](#). In *25th conference on uncertainty in artificial intelligence*, pp. 452–461. AUAI Press.
- Sarwar B., Karypis G., Konstan J., Riedl J. (2001). [Item-based collaborative filtering recommendation algorithms](#). In *10th international conference on World Wide Web*, pp. 285–295. ACM. doi:10.1145/371920.372071.
- Shi Y., Karatzoglou A., Baltrunas L., Larson M., Oliver N., Hanjalic A. (2012). [CLiMF: learning to maximize reciprocal rank with collaborative less-is-more filtering](#). In *6th ACM conference on Recommender systems*, pp. 139–146. ACM. doi:10.1145/2365952.2365981.
- Shu J., Shen X., Liu H., Yi B., Zhang Z. (2018). [A content-based recommendation algorithm for learning resources](#). *Multimedia Systems* 24(2), 163–173. doi:10.1007/s00530-017-0539-8.
- Şora I. (2015). [A PageRank based recommender system for identifying key classes in software systems](#). In *IEEE 10th Jubilee International Symposium on Applied Computational Intelligence and Informatics*, pp. 495–500. IEEE. doi:10.1109/SACI.2015.7208254.
- Steck H. (2013). [Evaluation of recommendations: rating-prediction and ranking](#). In *7th ACM conference on Recommender systems*, pp. 213–220. ACM. doi:10.1145/2507157.2507160.
- Strub F., Gaudel R., Mary J. (2016). [Hybrid recommender system based on autoencoders](#). In *1st Workshop on Deep Learning for Recommender Systems*, pp. 11–16. ACM. doi:10.1145/2988450.2988456.
- Tang J., Gao H., Liu H. (2012). [mTrust: Discerning multi-faceted trust in a connected world](#). In *5th ACM international conference on Web search and data mining*, pp. 93–102. ACM. doi:10.1145/2124295.2124309.
- Viard T., Latapy M., Magnien C. (2016). [Computing maximal cliques in link streams](#). *Theoretical Computer Science* 609, 245–252. doi:10.1016/j.tcs.2015.09.030.
- Xiang L., Yuan Q., Zhao S., Chen L., Zhang X., Yang Q., Sun J. (2010). [Temporal recommendation on graphs via long-and short-term preference fusion](#). In *16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 723–732. ACM. doi:10.1145/1835804.1835896.
- Xiao L., Min Z., Yongfeng Z., Yiqun L., Shaoping M. (2017). [Learning and transferring social and item visibilities for personalized recommendation](#). In *ACM Conference on Information and Knowledge Management*, pp. 337–346. ACM. doi:10.1145/3132847.3132911.
- Yu J., Shen Y., Yang Z. (2014). [Topic-STG: Extending the session-based temporal graph approach for personalized tweet recommendation](#). In *23rd International Conference on World Wide Web*, pp. 413–414. ACM. doi:10.1145/2567948.2577328.
- Zhao T., McAuley J., King I. (2014). [Leveraging social connections to improve personalized ranking for collaborative filtering](#). In *23rd ACM International Conference on Conference on Information and Knowledge Management*, pp. 261–270. ACM. doi:10.1145/2661829.2661998.

A ACKNOWLEDGEMENTS

We thank Raphaël Fournier, Tiphaine Viard and JIMIS reviewers for their helpful comments on previous versions. This work is funded in part by the African Center of Excellence in Information and Communication Technologies (CETIC), the Sorbonne University-IRD PDI program, and by the ANR (French National Agency of Research) under grant ANR-15-CE38-0001 (AlgoDiv).

B APPENDIX

In this section, we present the results obtained for top-20, -50 and -100. The section is divided in two parts: the first one presents performances obtained for all combinations of side information and basic graphs of the framework; the second highlights the 3 best combinations, according to basic graph and evaluation metric.

These two parts confirm observations made on top-10 results in the Section VI. For example, recommender graphs that integrate simultaneously content-based, users' preferences temporal dynamic and trust relationship between users, are usually the best. Thus, we recommend the simultaneous integration of these three side information in order to increase the chances to achieve good performances.

EPINIONS	F1@20			HR@20			MAP@20		
	BIP	STG	LSG	BIP	STG	LSG	BIP	STG	LSG
-	1.56	1.59	1.11	8.22	8.22	7.16	2.38	2.4	1.84
ET	1.31	1.49	1.05	7.43	7.96	6.9	2.18	2.24	1.8
IT	1.73	1.77	1.39	8.75	9.02	8.89	2.51	2.49	2.41
EDF	2.7	2.15	2.55	10.6	9.55	10.6	2.62	2.56	2.61
LDF	1.88	1.68	1.53	8.89	8.49	8.49	2.38	2.23	2.67
CI	1.95	2.15	0.76	9.28	9.68	5.44	2.64	2.98	1.89
CIU	2.16	2.36	0.63	9.95	10.1	4.64	2.9	3.13	1.73
EDF-ET	1.8	1.68	2.46	8.75	8.62	10.3	2.36	2.15	2.41
EDF-IT	2.47	2.5	2.43	9.95	10.1	10.3	2.84	2.67	2.64
LDF-ET	1.39	1.57	1.21	7.69	8.22	7.56	2.17	2.24	2.32
LDF-IT	1.92	1.95	2.18	9.55	9.81	10.5	3.12	3.12	3.06
CI-ET	1.67	2.12	0.79	8.49	9.55	5.57	2.4	2.83	1.9
CI-IT	2.05	2.19	1.49	9.95	10.1	8.75	2.63	2.8	2.35
CIU-ET	2.07	2.12	0.64	9.68	9.95	4.64	2.67	3.13	1.73
CIU-IT	2.19	2.35	1.53	10.1	10.1	9.15	2.9	3.13	2.43
CI-EDF	3.25	2.88	0.81	11.7	11.1	5.7	3.33	2.97	1.9
CI-LDF	2.57	2.38	0.85	10.6	9.68	5.97	2.76	2.83	1.9
CIU-EDF	3.45	2.93	0.67	11.8	11.3	5.04	3.45	3.26	1.79
CIU-LDF	3.02	2.39	0.74	11.0	10.6	5.31	3.01	3.09	1.82
CI-EDF-ET	2.6	2.58	0.81	10.5	10.7	5.7	2.7	2.71	1.9
CI-EDF-IT	3.25	3.26	1.52	11.3	11.4	8.89	3.03	2.99	2.4
CI-LDF-ET	1.98	2.17	0.86	9.15	9.68	5.97	2.42	2.83	1.9
CI-LDF-IT	2.63	2.3	2.09	10.5	10.2	9.81	3.15	3.14	2.77
CIU-EDF-ET	2.92	2.85	0.67	11.0	10.9	5.04	2.87	3.11	1.79
CIU-EDF-IT	3.44	3.33	1.48	11.7	11.5	9.02	3.55	3.41	2.46
CIU-LDF-ET	2.43	2.34	0.67	10.1	10.5	5.04	2.73	3.01	1.74
CIU-LDF-IT	2.81	2.51	1.93	10.7	10.5	9.42	3.29	3.24	3.26

Table 6: Epinions Dataset - Performances with optimal settings for Top-20.

EPINIONS	F1@50			HR@50			MAP@50		
	BIP	STG	LSG	BIP	STG	LSG	BIP	STG	LSG
-	1.06	1.08	0.91	15.3	15.4	13.9	2.5	2.55	2.03
ET	0.95	1.05	0.87	14.2	15.1	13.5	2.3	2.4	1.99
IT	1.16	1.16	0.97	16.0	16.4	14.7	2.52	2.63	2.55
EDF	1.25	1.28	1.04	17.1	16.4	15.4	2.64	2.71	2.7
LDF	1.14	1.15	0.99	16.0	16.0	14.7	2.5	2.4	2.72
CI	1.55	1.48	0.56	19.0	18.4	9.68	2.87	3.15	1.93
CIU	1.59	1.53	0.48	19.0	18.6	8.62	3.01	3.26	1.83
EDF-ET	1.09	1.11	0.89	15.5	15.5	14.2	2.43	2.33	2.52
EDF-IT	1.36	1.34	0.97	17.5	16.8	15.0	2.83	2.74	2.73
LDF-ET	0.97	1.08	0.87	14.6	15.4	13.5	2.28	2.4	2.38
LDF-IT	1.18	1.16	0.98	16.6	16.4	14.9	3.09	3.09	3.07
CI-ET	1.43	1.47	0.56	18.2	18.3	9.68	2.63	3.05	1.93
CI-IT	1.49	1.47	0.89	18.4	18.4	14.2	2.77	3.03	2.41
CIU-ET	1.49	1.49	0.48	18.3	18.6	8.49	2.76	3.27	1.83
CIU-IT	1.51	1.55	0.91	18.6	18.6	14.6	3.07	3.29	2.5
CI-EDF	1.74	1.66	0.58	19.9	19.9	9.95	3.29	3.06	1.94
CI-LDF	1.67	1.51	0.6	19.6	18.7	10.1	2.89	3.05	1.95
CIU-EDF	1.63	1.77	0.55	19.4	19.9	9.81	3.5	3.36	1.84
CIU-LDF	1.59	1.55	0.54	19.0	19.0	9.95	3.2	3.33	1.92
CI-EDF-ET	1.72	1.63	0.59	19.8	19.8	9.95	2.71	2.93	1.94
CI-EDF-IT	1.52	1.66	0.88	18.8	19.9	14.2	3.1	2.99	2.46
CI-LDF-ET	1.56	1.52	0.56	19.0	18.8	9.68	2.65	3.06	1.93
CI-LDF-IT	1.49	1.49	0.93	18.6	18.6	14.5	3.14	3.14	2.82
CIU-EDF-ET	1.52	1.75	0.55	18.7	19.9	9.81	3.02	3.29	1.83
CIU-EDF-IT	1.62	1.73	0.91	19.6	19.8	14.6	3.65	3.48	2.54
CIU-LDF-ET	1.47	1.55	0.51	18.2	19.0	9.42	2.83	3.18	1.84
CIU-LDF-IT	1.52	1.53	0.97	18.7	18.8	15.1	3.36	3.45	3.27

Table 7: Epinions Dataset - Performances with optimal settings for Top-50.

EPINIIONS	F1@100			HR@100			MAP@100		
	BIP	STG	LSG	BIP	STG	LSG	BIP	STG	LSG
-	0.7	0.72	0.67	22.5	22.4	21.9	2.54	2.59	2.13
ET	0.69	0.68	0.66	22.3	22.0	21.5	2.35	2.43	2.09
IT	0.74	0.77	0.66	23.3	23.6	21.9	2.6	2.68	2.61
EDF	0.77	0.77	0.67	22.9	23.3	21.9	2.66	2.66	2.72
LDF	0.71	0.72	0.67	22.5	22.3	21.9	2.54	2.42	2.65
CI	0.89	0.91	0.39	26.3	26.3	14.3	2.84	3.13	1.83
CIU	0.87	0.92	0.39	26.0	27.5	14.3	3.03	3.35	1.82
EDF-ET	0.71	0.7	0.65	22.1	22.0	21.2	2.44	2.33	2.54
EDF-IT	0.76	0.79	0.67	23.5	23.9	22.3	2.9	2.77	2.77
LDF-ET	0.71	0.69	0.66	22.4	21.9	21.5	2.33	2.42	2.36
LDF-IT	0.75	0.79	0.66	23.6	23.9	21.9	3.1	3.1	3.02
CI-ET	0.87	0.88	0.39	25.9	25.7	14.3	2.6	2.96	1.83
CI-IT	0.88	0.88	0.58	26.1	26.5	20.0	2.78	2.93	2.48
CIU-ET	0.86	0.92	0.39	25.9	27.5	14.3	2.85	3.36	1.82
CIU-IT	0.89	0.92	0.59	26.5	27.3	20.2	3.15	3.35	2.52
CI-EDF	0.92	0.98	0.42	26.8	27.6	15.1	3.29	3.15	1.85
CI-LDF	0.89	0.88	0.42	26.3	26.0	15.1	2.88	2.95	1.89
CIU-EDF	0.91	1.03	0.4	26.3	28.5	14.5	3.58	3.34	1.83
CIU-LDF	0.89	0.94	0.41	26.3	27.6	15.0	3.12	3.33	1.9
CI-EDF-ET	0.91	0.95	0.42	26.5	27.2	15.0	2.71	2.89	1.84
CI-EDF-IT	0.92	0.95	0.58	26.7	27.2	20.0	3.2	3.06	2.49
CI-LDF-ET	0.88	0.89	0.41	25.7	25.7	14.6	2.61	2.95	1.83
CI-LDF-IT	0.88	0.88	0.57	26.1	26.5	20.0	3.16	3.16	2.82
CIU-EDF-ET	0.91	1.02	0.39	26.1	28.5	14.3	2.94	3.21	1.82
CIU-EDF-IT	0.91	1.02	0.59	27.2	28.5	20.2	3.73	3.44	2.55
CIU-LDF-ET	0.88	0.94	0.39	26.0	27.6	14.3	2.79	3.17	1.82
CIU-LDF-IT	0.89	0.92	0.59	26.5	27.3	20.3	3.37	3.43	3.26

Table 8: Epinions Dataset - Performances with optimal settings for Top-100.

CIAO	F1@20			HR@20			MAP@20		
	BIP	STG	LSG	BIP	STG	LSG	BIP	STG	LSG
-	1.61	1.93	1.71	9.26	9.98	10.2	2.13	2.25	2.44
ET	1.36	1.69	1.61	8.71	9.26	9.98	1.96	2.13	2.35
IT	3.08	2.72	2.2	13.1	12.9	11.8	2.76	2.75	2.51
EDF	2.24	2.35	2.51	11.1	11.4	12.3	2.28	2.56	2.65
LDF	2.05	1.99	2.7	10.2	10.2	13.2	2.74	3.0	3.78
CI	3.13	3.58	1.03	12.2	12.3	7.44	2.37	2.45	1.75
CIU	2.93	4.39	0.87	13.2	14.5	6.72	2.8	3.43	1.55
EDF-ET	2.01	2.01	1.69	10.7	10.3	11.4	2.0	2.36	2.41
EDF-IT	3.15	2.69	2.34	14.2	13.8	12.0	3.27	3.11	2.56
LDF-ET	1.44	1.69	1.61	8.71	9.26	9.98	2.02	2.22	2.36
LDF-IT	2.82	2.82	2.6	12.3	12.5	12.3	3.06	3.25	2.81
CI-ET	2.59	3.25	1.09	11.6	12.2	7.8	2.3	2.39	1.88
CI-IT	3.35	3.47	2.76	13.4	14.0	12.7	2.87	2.83	2.68
CIU-ET	3.78	4.39	1.03	14.0	14.5	7.26	2.73	3.51	1.63
CIU-IT	3.64	4.71	3.27	14.5	14.7	13.4	3.13	3.39	2.66
CI-EDF	4.81	3.95	1.48	15.1	13.8	9.26	2.77	3.01	1.84
CI-LDF	4.27	3.92	1.54	14.9	14.2	10.3	3.29	3.21	2.92
CIU-EDF	4.37	4.19	1.32	15.8	14.7	8.17	3.66	3.6	1.84
CIU-LDF	3.43	4.55	1.39	15.2	15.4	9.62	3.62	3.68	2.82
CI-EDF-ET	2.19	3.78	1.47	13.8	13.6	9.07	2.76	2.78	1.93
CI-EDF-IT	4.18	3.86	2.95	15.4	14.9	13.1	3.41	3.42	2.76
CI-LDF-ET	2.7	3.33	1.24	11.6	13.1	8.71	2.31	2.44	2.01
CI-LDF-IT	4.87	4.92	3.38	15.1	14.5	13.8	3.44	3.48	3.45
CIU-EDF-ET	3.7	4.38	1.32	15.8	14.5	8.17	3.24	3.42	1.84
CIU-EDF-IT	3.84	4.56	3.39	16.0	15.8	13.6	3.63	3.64	2.69
CIU-LDF-ET	4.2	4.5	1.21	14.5	15.4	8.35	2.79	3.69	1.69
CIU-LDF-IT	5.06	4.36	3.28	16.0	15.8	14.5	3.59	3.69	3.49

Table 9: Ciao Dataset - Performances with optimal settings for Top-20.

CIAO	F1@50			HR@50			MAP@50		
	BIP	STG	LSG	BIP	STG	LSG	BIP	STG	LSG
-	1.46	1.55	1.51	19.4	19.6	19.8	2.34	2.5	2.74
ET	1.58	1.52	1.47	19.6	19.2	20.0	2.2	2.43	2.67
IT	1.69	1.72	1.62	20.9	21.1	20.5	2.92	2.84	2.73
EDF	1.74	1.74	1.81	21.4	20.7	20.9	2.56	2.78	2.86
LDF	1.47	1.62	1.77	19.4	20.3	20.9	2.91	3.0	3.97
CI	2.27	2.18	0.97	22.1	22.5	15.2	2.55	2.71	1.92
CIU	2.37	2.65	0.88	24.0	24.9	14.3	3.03	3.67	1.72
EDF-ET	1.63	1.59	1.59	20.3	20.9	20.7	2.24	2.65	2.73
EDF-IT	1.98	1.79	1.7	22.1	21.8	21.2	3.39	3.23	2.81
LDF-ET	1.59	1.58	1.47	19.6	19.6	20.0	2.25	2.54	2.68
LDF-IT	1.9	1.69	1.65	21.8	20.5	20.5	3.18	3.29	2.99
CI-ET	2.34	2.07	0.98	23.0	23.0	15.4	2.51	2.68	1.92
CI-IT	2.53	2.29	1.57	23.6	22.7	20.5	2.98	2.91	2.76
CIU-ET	2.27	2.66	0.88	24.0	24.7	14.2	2.98	3.75	1.74
CIU-IT	2.52	2.86	1.58	24.7	25.8	21.2	3.26	3.62	2.88
CI-EDF	2.45	2.36	1.26	23.0	24.0	18.9	2.98	3.18	2.02
CI-LDF	2.56	2.23	1.17	23.4	22.9	17.6	3.45	3.29	3.12
CIU-EDF	2.57	2.71	1.03	24.0	25.4	16.5	3.87	3.87	2.07
CIU-LDF	2.62	2.78	1.13	24.0	24.7	16.9	3.74	3.82	3.01
CI-EDF-ET	2.63	2.34	1.29	24.1	23.8	18.9	3.01	3.04	2.05
CI-EDF-IT	2.6	2.44	1.59	24.5	24.1	20.7	3.53	3.56	2.94
CI-LDF-ET	2.31	2.21	0.99	23.4	23.8	15.4	2.63	2.75	2.05
CI-LDF-IT	2.51	2.42	1.64	23.8	23.8	21.1	3.61	3.56	3.61
CIU-EDF-ET	2.69	2.69	1.07	24.3	25.8	16.9	3.42	3.7	2.07
CIU-EDF-IT	2.82	2.69	1.61	25.4	25.4	21.4	3.69	3.73	2.91
CIU-LDF-ET	2.55	2.72	1.08	23.8	25.2	16.7	3.0	3.86	1.93
CIU-LDF-IT	2.99	2.88	1.61	24.9	24.9	20.9	3.69	3.76	3.51

Table 10: Ciao Dataset - Performances with optimal settings for Top-50.

CIAO	F1@100			HR@100			MAP@100		
	BIP	STG	LSG	BIP	STG	LSG	BIP	STG	LSG
-	1.11	1.15	1.06	27.9	28.3	27.8	2.39	2.49	2.64
ET	1.08	1.15	1.03	27.9	28.1	27.4	2.25	2.36	2.56
IT	1.16	1.15	1.07	28.7	28.7	27.9	2.88	2.85	2.73
EDF	1.25	1.27	1.18	30.5	31.0	29.9	2.46	2.64	2.86
LDF	1.23	1.14	1.17	29.6	28.5	29.2	2.96	3.05	3.94
CI	1.54	1.51	0.96	33.8	33.4	26.7	2.62	2.74	2.02
CIU	1.58	1.57	0.89	34.3	34.5	25.8	3.08	3.51	1.72
EDF-ET	1.21	1.27	1.11	29.2	30.3	28.3	2.18	2.57	2.62
EDF-IT	1.22	1.26	1.09	28.9	30.1	27.9	3.37	3.22	2.78
LDF-ET	1.11	1.15	1.03	28.7	28.7	27.6	2.32	2.47	2.57
LDF-IT	1.14	1.18	1.07	27.9	29.2	27.9	3.22	3.31	3.02
CI-ET	1.57	1.52	0.98	33.9	33.4	26.5	2.58	2.65	2.04
CI-IT	1.52	1.53	1.09	33.2	33.4	29.9	2.97	2.91	2.83
CIU-ET	1.57	1.56	0.9	34.8	35.2	25.8	3.03	3.6	1.8
CIU-IT	1.48	1.5	1.09	33.4	34.1	29.2	3.23	3.47	2.85
CI-EDF	1.66	1.56	1.01	34.5	33.6	28.5	2.92	3.13	2.14
CI-LDF	1.6	1.49	1.0	34.5	33.6	27.2	3.37	3.32	3.18
CIU-EDF	1.73	1.74	0.9	35.0	35.9	26.5	3.83	3.76	2.17
CIU-LDF	1.56	1.5	0.89	34.5	34.3	26.0	3.7	3.81	2.98
CI-EDF-ET	1.71	1.54	1.01	34.5	33.6	29.2	2.91	2.98	2.12
CI-EDF-IT	1.57	1.54	1.16	33.0	33.4	29.8	3.41	3.55	2.85
CI-LDF-ET	1.55	1.55	0.98	33.9	33.4	26.5	2.63	2.71	2.16
CI-LDF-IT	1.49	1.5	1.15	32.7	33.8	30.5	3.58	3.59	3.5
CIU-EDF-ET	1.84	1.76	0.91	35.8	35.9	26.9	3.33	3.66	2.17
CIU-EDF-IT	1.74	1.73	1.1	34.7	35.8	29.6	3.64	3.68	2.89
CIU-LDF-ET	1.59	1.57	0.9	34.7	34.7	25.8	3.04	3.72	1.98
CIU-LDF-IT	1.52	1.51	1.13	33.2	34.5	30.5	3.75	3.79	3.46

Table 11: Ciao Dataset - Performances with optimal settings for Top-100.

Epinions Dataset													
	No	BIP				STG				LSG			
		Basic	Best	Imp.	BIP-Best	Basic	Best	Imp.	STG-Best	Basic	Best	Imp.	LSG-Best
F@20	1	1.56	3.45	121%	CIU-EDF	1.59	3.33	109%	CIU-EDF-IT	1.11	2.55	129%	EDF
	2	1.56	3.44	120%	CIU-EDF-IT	1.59	3.26	105%	CI-EDF-IT	1.11	2.46	121%	EDF-ET
	3	1.56	3.25	109%	CI-EDF-IT	1.59	2.93	84%	CIU-EDF	1.11	2.43	118%	EDF-IT
H@20	1	8.22	11.8	43%	CIU-EDF	8.22	11.5	40%	CIU-EDF-IT	7.16	10.6	48%	EDF
	2	8.22	11.7	41%	CI-EDF	8.22	11.4	38%	CI-EDF-IT	7.16	10.5	46%	LDF-IT
	3	8.22	11.7	41%	CIU-EDF-IT	8.22	11.3	37%	CIU-EDF	7.16	10.3	44%	EDF-ET
M@20	1	2.38	3.55	49%	CIU-EDF-IT	2.4	3.41	41%	CIU-EDF-IT	1.84	3.26	77%	CIU-LDF-IT
	2	2.38	3.45	45%	CIU-EDF	2.4	3.26	35%	CIU-EDF	1.84	3.06	66%	LDF-IT
	3	2.38	3.33	40%	CI-EDF	2.4	3.24	34%	CIU-LDF-IT	1.84	2.77	50%	CI-LDF-IT

Epinions Dataset													
	No	BIP				STG				LSG			
		Basic	Best	Imp.	BIP-Best	Basic	Best	Imp.	STG-Best	Basic	Best	Imp.	LSG-Best
F@50	1	1.06	1.74	64%	CI-EDF	1.08	1.77	62%	CIU-EDF	0.91	1.04	14%	EDF
	2	1.06	1.72	61%	CI-EDF-ET	1.08	1.75	61%	CIU-EDF-ET	0.91	0.99	9%	LDF
	3	1.06	1.67	57%	CI-LDF	1.08	1.73	59%	CIU-EDF-IT	0.91	0.98	7%	LDF-IT
H@50	1	15.3	19.9	30%	CI-EDF	15.4	19.9	29%	CI-EDF	13.9	15.4	10%	EDF
	2	15.3	19.8	29%	CI-EDF-ET	15.4	19.9	29%	CIU-EDF	13.9	15.1	8%	CIU-LDF-IT
	3	15.3	19.6	28%	CI-LDF	15.4	19.9	29%	CI-EDF-IT	13.9	15.0	7%	EDF-IT
M@50	1	2.5	3.65	45%	CIU-EDF-IT	2.55	3.48	36%	CIU-EDF-IT	2.03	3.27	61%	CIU-LDF-IT
	2	2.5	3.5	40%	CIU-EDF	2.55	3.45	35%	CIU-LDF-IT	2.03	3.07	51%	LDF-IT
	3	2.5	3.36	34%	CIU-LDF-IT	2.55	3.36	31%	CIU-EDF	2.03	2.82	38%	CI-LDF-IT

Epinions Dataset													
	No	BIP				STG				LSG			
		Basic	Best	Imp.	BIP-Best	Basic	Best	Imp.	STG-Best	Basic	Best	Imp.	LSG-Best
F@100	1	0.7	0.92	30%	CI-EDF-IT	0.72	1.03	44%	CIU-EDF	0.67	0.67	0%	-
	2	0.7	0.92	30%	CI-EDF	0.72	1.02	42%	CIU-EDF-ET	0.67	0.67	0%	LDF
	3	0.7	0.91	29%	CIU-EDF-IT	0.72	1.02	42%	CIU-EDF-IT	0.67	0.67	0%	EDF-IT
H@100	1	22.5	27.2	20%	CIU-EDF-IT	22.4	28.5	27%	CIU-EDF	21.9	22.3	1%	EDF-IT
	2	22.5	26.8	18%	CI-EDF	22.4	28.5	27%	CIU-EDF-ET	21.9	21.9	0%	-
	3	22.5	26.7	18%	CI-EDF-IT	22.4	28.5	27%	CIU-EDF-IT	21.9	21.9	0%	IT
M@100	1	2.54	3.73	46%	CIU-EDF-IT	2.59	3.44	32%	CIU-EDF-IT	2.13	3.26	53%	CIU-LDF-IT
	2	2.54	3.58	41%	CIU-EDF	2.59	3.43	32%	CIU-LDF-IT	2.13	3.02	41%	LDF-IT
	3	2.54	3.37	33%	CIU-LDF-IT	2.59	3.36	29%	CIU-ET	2.13	2.82	32%	CI-LDF-IT

Table 12: Epinions Dataset - Best recommender graphs for Top-20, -50 and -100. Comparison of the three best recommender graph combinations with the associated basic graph.

Ciao Dataset													
	No	BIP				STG				LSG			
		Basic	Best	Imp.	BIP-Best	Basic	Best	Imp.	STG-Best	Basic	Best	Imp.	LSG-Best
F@20	1	1.61	5.06	215%	CIU-LDF-IT	1.93	4.92	155%	CI-LDF-IT	1.71	3.39	98%	CIU-EDF-IT
	2	1.61	4.87	202%	CI-LDF-IT	1.93	4.71	144%	CIU-IT	1.71	3.38	97%	CI-LDF-IT
	3	1.61	4.81	199%	CI-EDF	1.93	4.56	136%	CIU-EDF-IT	1.71	3.28	91%	CIU-LDF-IT
H@20	1	9.26	16.0	72%	CIU-EDF-IT	9.98	15.8	58%	CIU-EDF-IT	10.2	14.5	42%	CIU-LDF-IT
	2	9.26	16.0	72%	CIU-LDF-IT	9.98	15.8	58%	CIU-LDF-IT	10.2	13.8	35%	CI-LDF-IT
	3	9.26	15.8	70%	CIU-EDF	9.98	15.4	54%	CIU-LDF	10.2	13.6	33%	CIU-EDF-IT
M@20	1	2.13	3.66	72%	CIU-EDF	2.25	3.69	64%	CIU-LDF-ET	2.44	3.78	54%	LDF
	2	2.13	3.63	70%	CIU-EDF-IT	2.25	3.69	64%	CIU-LDF-IT	2.44	3.49	42%	CIU-LDF-IT
	3	2.13	3.62	70%	CIU-LDF	2.25	3.68	63%	CIU-LDF	2.44	3.45	41%	CI-LDF-IT

Ciao Dataset													
	No	BIP				STG				LSG			
		Basic	Best	Imp.	BIP-Best	Basic	Best	Imp.	STG-Best	Basic	Best	Imp.	LSG-Best
F@50	1	1.46	2.99	104%	CIU-LDF-IT	1.55	2.88	85%	CIU-LDF-IT	1.51	1.81	20%	EDF
	2	1.46	2.82	93%	CIU-EDF-IT	1.55	2.86	84%	CIU-IT	1.51	1.77	17%	LDF
	3	1.46	2.69	83%	CIU-EDF-ET	1.55	2.78	79%	CIU-LDF	1.51	1.7	12%	EDF-IT
H@50	1	19.4	25.4	30%	CIU-EDF-IT	19.6	25.8	31%	CIU-IT	19.8	21.4	8%	CIU-EDF-IT
	2	19.4	24.9	28%	CIU-LDF-IT	19.6	25.8	31%	CIU-EDF-ET	19.8	21.2	7%	EDF-IT
	3	19.4	24.7	27%	CIU-IT	19.6	25.4	29%	CIU-EDF	19.8	21.2	7%	CIU-IT
M@50	1	2.34	3.87	65%	CIU-EDF	2.5	3.87	54%	CIU-EDF	2.74	3.97	44%	LDF
	2	2.34	3.74	60%	CIU-LDF	2.5	3.86	54%	CIU-LDF-ET	2.74	3.61	31%	CI-LDF-IT
	3	2.34	3.69	57%	CIU-LDF-IT	2.5	3.82	52%	CIU-LDF	2.74	3.51	28%	CIU-LDF-IT

Ciao Dataset													
	No	BIP				STG				LSG			
		Basic	Best	Imp.	BIP-Best	Basic	Best	Imp.	STG-Best	Basic	Best	Imp.	LSG-Best
F@100	1	1.11	1.84	66%	CIU-EDF-ET	1.15	1.76	52%	CIU-EDF-ET	1.06	1.18	11%	EDF
	2	1.11	1.74	57%	CIU-EDF-IT	1.15	1.74	51%	CIU-EDF	1.06	1.17	10%	LDF
	3	1.11	1.73	56%	CIU-EDF	1.15	1.73	50%	CIU-EDF-IT	1.06	1.16	9%	CI-EDF-IT
H@100	1	27.9	35.8	27%	CIU-EDF-ET	28.3	35.9	26%	CIU-EDF	27.8	30.5	9%	CI-LDF-IT
	2	27.9	35.0	25%	CIU-EDF	28.3	35.9	26%	CIU-EDF-ET	27.8	30.5	9%	CIU-LDF-IT
	3	27.9	34.8	24%	CIU-ET	28.3	35.8	26%	CIU-EDF-IT	27.8	29.9	7%	EDF
M@100	1	2.39	3.83	60%	CIU-EDF	2.49	3.81	52%	CIU-LDF	2.64	3.94	49%	LDF
	2	2.39	3.75	57%	CIU-LDF-IT	2.49	3.79	52%	CIU-LDF-IT	2.64	3.5	32%	CI-LDF-IT
	3	2.39	3.7	55%	CIU-LDF	2.49	3.76	51%	CIU-EDF	2.64	3.46	31%	CIU-LDF-IT

Table 13: Ciao Dataset - Best recommender graphs for Top-20, -50 and -100. Comparison of the three best recommender graph combinations with the associated basic graph.