The essential role of time in network-based recommendation

ALEXANDRE VIDMER and MATÚŠ MEDO (a)

Department of Physics, University of Fribourg - Chemin du Muse 3, CH-1700 Fribourg, Switzerland

received 10 June 2016; accepted in final form 5 December 2016

published online 21 December 2016

PACS 07.05.Kf – Data analysis: algorithms and implementation; data management
PACS 89.20.-a – Interdisciplinary applications of physics
PACS 89.20.Ff – Computer science and technology

Abstract – Random walks on bipartite networks have been used extensively to design personalized recommendation methods. While aging has been identified as a key component in the growth of information networks, most research has focused on the networks’ structural properties and neglected the often available time information. Time has been largely ignored both by the investigated recommendation methods as well as by the methodology used to evaluate them. We show that this time-unaware approach overestimates the methods’ recommendation performance. Motivated by microscopic rules of network growth, we propose a time-aware modification of an existing recommendation method and show that by combining the temporal and structural aspects, it outperforms the existing methods. The performance improvements are particularly striking in systems with fast aging.

Introduction. – Increasing data availability and computational capacity [1], interconnections between previously separate data domains [2,3], and the immediate commercial importance of recommendation [4–6] all contribute to the unceasing interest in the study of recommender systems [7]. The goal of recommendation is to use data on past user preferences to obtain personalized “recommendation” of new items (shopping items, YouTube videos, or any other content) that an individual user might appreciate. From the physics perspective, it has been interesting to realize that well-known physics processes, such as random walks and heat diffusion, on network representations [8] of the underlying data give rise to efficient recommendation methods [9–12].

Despite physics being a science that aims at understanding the evolution of systems, the research of network-based recommendation by physicists has entirely neglected the dimension of time which turns out to be of high importance for traditional recommendation approaches [5,13–15]. While the ignorance of time is understandable from the historical perspective —early datasets often lacked the time information—the situation is very different now. The role of time in the evolution of information networks (that serve as input data for recommendation) has been demonstrated [16–18], modeled [19–21], and turned into numerous useful applications [22–24].

The ignorance of time in the research of network-based recommendation manifests itself prominently in the evaluation of recommendation methods. This evaluation is usually done by hiding part of the input data—this part is commonly referred to as the probe set—and using the rest of the data—which is referred to as the training set—as input for a recommendation method. The obtained results are finally evaluated on the basis of how well they reproduce the hidden data [25,26]. Crucially, the probe is typically chosen at random [12,27]; the training set therefore includes both past and future data entries.

We show for the first time that when instead the latest data are hidden and thus the task is to reproduce strictly future data entries, the performance of network-based recommendation methods becomes dramatically worse than the performance observed on a random probe. Sophisticated network-based recommendation methods then turn out to be outperformed by trivial approaches such as the recent item popularity increase which has been shown to be a good predictor of the future popularity increase [28].

We propose to combine a simple network-based recommendation method with dynamical features of the network growth. The resulting hybrid method is shown to consistently and significantly outperform the existing network-based methods on various e-commerce datasets. Note that various well-performing recommendation methods that take temporal dynamics into account already

(a)E-mail: matus.medo@unifr.ch
exist (e.g., [29]). Our work is the first one that does the same for network-based recommendation.

The problem of a random probe. – We begin by describing in detail how a probe chosen at random favors network-based recommendation methods. Input data for network-based recommendation consists of a set of links between users and items; links connect users with the items that they have collected, or otherwise associated themselves with (e.g., a user is connected with the products he has reviewed on Amazon.com). These data are naturally represented as a bipartite user-item network with $U$ user nodes and $I$ item nodes. Elements of the network’s evolving adjacency matrix, $a_{\alpha i}(t)$, are one when user $i$ is connected with item $\alpha$ at time $t$ and zero otherwise. Degree $k_\alpha(t)$ of user node $\alpha$ corresponds to the number of items collected by user $i$ until time $t$, and degree $k_i(t)$ of item node $\alpha$ corresponds to the number of users who have collected item $\alpha$ until time $t$. Both the user and the item degree distribution are typically broad [30,31].

Upon random probe division, links from the input data are chosen at random and moved to the probe. The probability that a randomly chosen link is attached to item $\alpha$ is proportional to its degree $k_\alpha(t)$. At the same time, most network-based recommendation methods are known to tend to recommend popular items [11,32,33] which means that choosing the probe at random is inherently favorable to these methods because thus-created probes share the bias of recommendation methods themselves.

While the tendency of the random probe towards popular items may seem worrisome, one might think that due to the omnipresent preferential attachment mechanism [34] (sect. 8), the degree of an item is a good proxy for the item’s future degree increase. If that were the case, the random probe would be effectively equivalent to the probe consisting of the most recent links. However, aging has been shown to be essential in the growth of information networks [19,20,35,36]. The relationship between the degree of a node and its future degree increase is therefore more complex (see fig. 1). Particularly, two nodes of the same degree may have very different expected degree increase if their age differs substantially. In summary, only evaluation on the time probe can be representative of how a recommendation method represents the users’ future actions and interests.

With the help of the network growth model with relevance and aging [19], the previous arguments can be put in a more precise mathematical form. According to the model, nodes acquire new links at the rate proportional to

$$k_\alpha(t)R_\alpha(t),$$

where $R_\alpha(t)$ is the relevance of node $\alpha$ at time $t$; $R_\alpha(t)$ generally decays with time. It thus follows that $k_\alpha(t)$ alone is a poor predictor of the node degree increase in the future because small relevance can prevent a popular node from receiving many new links. To obtain better predictions of the degree increase, one may look at the system at a sufficiently short time scale $\tau$ over which both $k_\alpha(t)$ and $R_\alpha(t)$ change little and therefore, according to eq. (1), the node’s propensity to attract new links remains approximately unchanged. If the node has attracted $k_\alpha(t) - k_\alpha(t - \tau)$ links in the past, we thus expect it to attract the same number of links in the future. The recent degree increase of a node is therefore anticipated to predict well the node’s future degree increase.

![Fig. 1: (Color online) Comparison of the random and time probe in the Netflix data.](image)

**Recommendation methods and their evaluation.** – We now describe an elementary network-based recommendation method, a benchmark time-aware method, and a novel hybrid method. Two further network-based methods [11,37] are described in SI.

1) **Probabilistic spreading** (ProbS). All items collected by a chosen user until time $t$ are assigned a unit amount of resource which is then propagated over the bipartite network by a random walk process from the item side to the user side and back [10]. In each step, the amount
of resource at a node is uniformly distributed among the node’s neighbors. The resource values obtained on the item side in the second propagation step are consequently interpreted as item recommendation scores for the given user. Denoting the initial resource vector for user $i$ as $f^{(i)}$, its elements are $f^{(i)}_{j_0} = a_{i_0}(t)$, the final resource values $g^{(i)}_k$ can be written as $g^{(i)}_k = \sum_{\beta=1}^I W_{k_\beta}(t) f^{(i)}_{j_\beta}$, where

$$W_{k_\beta}(t) = \frac{1}{k_\beta(t)} \sum_{j=1}^U a_{j_\alpha}(t) a_{j_\beta}(t)$$

(2)

This directly represents the above-described spreading process: $a_{j_\alpha} a_{j_\beta}$ corresponds to paths going from item $j$ to item $\alpha$ through user $j$ and the division with $k_\beta$ and $k_\beta$ corresponds to the uniform division of resources among neighbors of nodes $\alpha$ and $j$, respectively.

2) The recent degree increase (DI) has been shown to perform well in predicting the future popularity of items [28] (see also fig. 1(d)). After choosing the time window length $\tau$, one computes the item degree increase within this window as $\Delta k_\alpha(t, \tau) = k_\alpha(t) - k_\alpha(t - \tau)$. The drawback of this approach is that many items can share the same value of $\Delta k_\alpha(t, \tau)$ and the order in which these items should be represented to a user is then ambiguous (this is particularly relevant when $\tau$ is small and many of the degree increase values are small or zero). Since the item popularity is an important factor in the evolution of a network, we complement the recent degree increase with item popularity and compute the item score as

$$\Delta k^{(i)}_\alpha(t, \tau) = \Delta k_\alpha(t, \tau) + \varepsilon k_\alpha(t)$$

(3)

The value of $\varepsilon$ is set small enough so that the order of items of different $\Delta k_\alpha(t, \tau)$ is not changed (this is guaranteed when $\varepsilon < 1/\max k_\alpha$; we use $\varepsilon = 10^{-9}$ in our simulations).

3) Time-aware probabilistic spreading (TProbS). Items at the top of ProbS’s recommendation lists tend to have high degree [32,33]. To achieve high degree, nodes typically need considerable time and, as a result, ProbS tends to recommend old items. To improve the ranking of recent items, we propose to combine the ProbS score $g^{(i)}_k$ with the network’s recent temporal dynamics as

$$t^{(i)}_k = g^{(i)}_k \frac{\Delta k^{(i)}_\alpha(t, \tau)}{k_\alpha(t)}$$

(4)

By multiplying with $\Delta k^{(i)}_\alpha(t, \tau)$, we give advantage to the items that have recently attracted substantial new audience; the momentum of popularity demonstrated in fig. 1(d) suggests that these items will be also attractive in the near future. By additionally dividing by $k_\alpha(t)$, we compensate for the fact that since both $g^{(i)}_k$ and $\Delta k^{(i)}_\alpha(t, \tau)$ are generally proportional to the item degree, their combination $g^{(i)}_k \Delta k^{(i)}_\alpha(t, \tau)$ strongly favors popular items.

4) Similarity-preferential diffusion (SimS) has been proposed in [37] and found to be one of the best-performing network-based recommendation methods in [12]. One first introduces user similarity based on the items that the two users have both collected,

$$s_{ij} = \sum_{\beta=1}^I \frac{a_{i_\beta} a_{j_\beta}}{k_\beta}$$

(5)

and the final resource values take the form

$$h^{(i)}_\alpha = \sum_{j=1}^U \frac{a_{j_\alpha} s^{(i)}_{ij}}{k_\beta^2 k_\alpha^{1-\lambda}}$$

(6)

Here $\theta > 0$ is a parameter that decides how much more weight we give the users with high similarity and $\lambda \in [0,1]$ is a parameter that combines ProbS with its counterpart HeatS (heat spreading; see [11] for the ProbS-HeatS hybrid). SimS reduces to ProbS when $\theta = 1$ and $\lambda = 1$ and to HeatS when $\theta = 1$ and $\lambda = 0$.

5) Time-aware version of the ProbS-HeatS hybrid method (THybrid) combines SimS with DI in the same way as TProbS. Denoting the final resource values of the ProbS-HeatS hybrid as $h^{(i)}_\alpha$, the THybrid score reads

$$u^{(i)}_\alpha = h^{(i)}_\alpha \frac{\Delta k^{(i)}_\alpha(0, \tau)}{k_\alpha(0)}$$

(7)

THybrid has two parameters to optimize over: $\tau$ and $\lambda$.

As mentioned above, recommendation methods are evaluated by measuring their ability to reproduce the hidden probe data. The traditional random probe consists of a chosen fraction (we use here 10%) of all links that are chosen at random. The remaining 90% of data then form the training data that are used as input for recommendation methods. The natural way to modify this construction to a time-aware probe would be to choose the most recent 10% of all links. However, this approach has two disadvantages. First, it would only allow to construct one probe (as opposed to the random probe where different random divisions of the input data can be evaluated) and hence provides limited information about the inherent variability of the recommendation performance. Second, due to the new items constantly appearing in the system, the most recent 10% of the links that are moved to the time-aware probe are likely to contain a number of new items that are not present in the remaining training data. Since they lack any connections, these items cannot be recommended to the users by a network-based recommendation method which effectively and non-transparently reduces the size of the probe set to the actually recommendable items. While various approaches have been developed to overcome this “cold start” problem [38], they generally go beyond the scope of this article.

To avoid the mentioned disadvantages, we choose a different time-aware approach. Assuming that the time stamps of the complete data run from 0 to $T_m$, we choose the probe time stamp $T_P$ at random and move all entries with time stamps from $[T_P, T_P + \Delta_P)$ to the probe. All entries with time stamps smaller than $T_P$ then constitute the
training data (see fig. 2 for an illustration). The parameter \( \Delta_P \) characterizes the temporal span of the probe; it should be long enough to allow a substantial number of entries to enter the probe, yet short enough to prevent a large number of probe entries which correspond to items that are not present in the training data. We choose \( \Delta_P = 1 \) day for Netflix and Yelp, and \( \Delta_P = 1 \) hour for Digg; the fraction of probe links corresponding to the items absent in the training set is negligible (less than 2%) in all three cases. The exact choice of \( \Delta_P \) and the \( T_P \) ranges does not qualitatively alter the results. To calibrate the recommendation methods, we constrain the choice of \( T_P \) to \([0.8 T_m, 0.9 T_m]\). To obtain the final performance measurements, we draw \( T_P \) at random from \([0.9 T_m, T_m - \Delta_P]\). In both calibration and final evaluation, results are averaged over 100 values of \( T_P \). Note that while other time-aware approaches to constructing the probe exist (such as the ones where, unlike in our approach, the ratio between the number of training and probe entries is fixed), they generally lead to the same conclusions.

**Evaluation metrics.** Upon computing recommendation scores of items for all users, we are ultimately interested in knowing, which items we should “recommend” to each individual user. Since all methods implicitly assume that the higher the score is, the more likely the item is to be appreciated by the user, a recommendation list for a user is obtained by ranking all items according to their recommendation score for the given user in a descending order. In doing so, the items that have been already collected by the user are ignored because the user already knows them.

To quantify recommendation accuracy, we use recall which is a standard metric for accuracy in information filtering [25,26]. For a given user, recall is defined as the fraction of the user’s probe items that occur in the top \( L \) places of the recommendation list. Probe items that are absent in the training data (mostly items that only appear at time \( T_P \) or later) are ignored. The fraction of such probe entries is negligible —less than 2% for Digg and further less for the other two datasets. Overall recall \( R(L) \) is then computed by averaging recall values over all users. Recall ranges from zero (worst) to one (best). The ranking score of item \( \alpha \) in the recommendation list of user \( i \) is obtained by dividing the item’s ranking position \( r_{i\alpha} \) by the number of items that have not yet been collected by user \( i \). Ranking score of a method is obtained by averaging \( r_{i\alpha}/(L - k_i) \) over all probe entries [11]. The best result is close to zero (when all probe items rank at the top of their respective recommendation lists, indicating good recommendation results), while the worst result is 0.5 (when the probe items are randomly distributed in the recommendation lists). Finally, to address the diversity aspect of recommendation [11], we compute the average degree (popularity) of the top \( L \) items at time \( T_P \), \( k_R(L) \). We use \( L = 50 \) for \( R(L) \) and \( k_R(L) \).

**Description of the analyzed datasets.** We evaluate the chosen recommendation methods on three datasets originating from distinct e-commerce systems. Basic statistical properties of these datasets are summarized in table 1.

Netflix.com is a streaming service that was mainly a DVD rental service at the time when the dataset has been released. The company has released the dataset, which is nowadays a golden standard in the information filtering community, for its Netflix Prize [39]. Out of the 450000 users contained in the original dataset, we have chosen 2000 users at random and connect them with the items that they have rated with rating 3 or higher. Resolution of the dataset’s time stamps is one day.

Yelp.com is a website where users can review and rate various businesses such as restaurants, doctors, and bars. This and several other datasets have been released by the company for its dataset challenges [40]. In the dataset, the users are connected with the businesses that they have rated with rating 3 or higher. Resolution of the dataset’s time stamps is one day.

In Digg.com, users can vote for any story of their interest. The Digg dataset is composed of news that were collected over the period of one month in 2009 [41]; only stories that were promoted to the web site’s front page are included. While the mechanism of promotion to the font page is not explicitly available, it is obvious that story popularity is an important factor (the lowest story degree in the dataset is 111). A link is created in the network representation between a user and a story when the user votes for the story. Resolution of the dataset’s time stamps is one minute.

**Results.**– Method calibration with respect to the time window length \( \tau \) is shown in fig. 3 (for methods with multiple parameters, we show here their best-performing choice). The recall-maximizing parameter values are then used in out-of-sample measurement \( (T_P \in [0.9 T_m, T_m - \Delta_P]) \) of the recommendation performance that is summarized in table 2. This table further includes recall values obtained with ProbS on the random probe. In agreement
Fig. 3: (Color online) Method calibration on time probes with $T_P \in [0.8T_m, 0.9T_m]$: recommendation recall, ranking score, and popularity of the recommended items as a function of the window length $\tau$ for the evaluated methods. Horizontal lines mark the performance of the time-unaware method with the highest recall (SimS with $\lambda = 0.3$ and $\theta = 1.25$ for Netflix, ProbS for Yelp, and HeatS for Digg).

Table 2: Method evaluation on time probes with $T_P \in [0.9T_m, T_m - \Delta_P]$: recall for recommendation list length $L = 50$ for ProbS on the random probe and for all evaluated methods on the time probe; we use here the optimal parameter values presented in fig. 3. “No time” stands for the best-performing time-unaware recommendation method (specified in the caption of fig. 3).

| Dataset | Random probe | Time probe |
|---------|--------------|------------|
|         | ProbS | ProbS | DI | No time | TProbS |
| Netflix | 0.24 | 0.08 | 0.15 | 0.10 | **0.18** |
| Yelp    | 0.18 | 0.09 | 0.06 | 0.10 | **0.12** |
| Digg    | 0.18 | 0.06 | 0.71 | 0.15 | **0.74** |

with our previous arguments about the inadequacy of the random probe for recommendation evaluation, we see that the recall values drop by a factor two or three once the time probe is used instead of the random one. The random probe gives an excessively optimistic view of the performance of network-based recommendation methods.

The main finding is that the use of time-aware recommendation substantially improves recall in all three datasets studied here. The biggest recall improvement is observed on the Digg data where aging is much faster than in Netflix and Yelp (the time $T_{50}$ in which the most popular items reach 50% of their final popularity on average is 16 hours in Digg as opposed to 680 days in Netflix and 690 days in Yelp). At the same time, recall obtained in Digg data with time-aware recommendation is only marginally higher than recall obtained with recent degree increase. This is due to the selective nature of this dataset which includes only the most popular Digg stories. Heterogeneous taste of users and personalized recommendation are thus limited in effect. We present the Digg dataset because of its quick aging dynamics that fully exposes the inappropriateness of traditional recommendation methods. By contrast, the gap between TProbS and DI is particularly big for the Yelp data, which may be due to geographical effects: even when a restaurant’s recent degree increase is high, recommending it to a user
who lives far away is unlikely to prove useful. By contrast, TProbS does not face the same problem because user location is reflected by their collection patterns. One can also note that the relative improvement achieved with TProbS on this dataset is smaller than it is on the Netflix and Digg data. An inspection of the time probe reveals that this is due to the fact that in the Yelp data, the aging dynamics is different as there are items of substantial age that are present in the probe often (perpetually popular items).

Besides significantly improving recommendation recall for all three datasets, the average degree of the recommended items is for TProbS considerably lower than for traditional methods, which demonstrates the method’s potential to increase recommendation diversity which is a long-standing issue [42–44]. In particular, when TProbS is used instead of ProbS, \(k_T(50)\) decreases by 38%, 50%, and 82% for the Netflix, Yelp, and Digg data, respectively. THybrid, which is a time-aware modification of the popular ProbS-HeatS hybrid recommendation method, further improves recall and lowers the average popularity of top-ranked items. Results for other recommendation list lengths \(L\) are qualitatively the same.

Figure 3 further shows that for the Netflix and Yelp data, the ranking score achieved by TProbS is nearly independent of \(\tau\), which is very different from recall that strongly changes with \(\tau\). This is an indication that the ranking score with its coverage of all probe items—including those that are ranked very badly in users’ recommendation lists—is a little reliable indicator of the recommendation performance. By contrast, recall is only sensitive to probe items appearing at the top \(L\) places of recommendation lists and thus has the potential to reflect the users’ benefit from recommendation. Note that the recall maxima in fig. 3 occur at \(\tau^*\) which roughly fulfills the rule of thumb \(\tau^* \approx T_{50}/10\) (here \(T_{50}\) is the half-time of popularity growth defined in the text). While the direct proportionality between \(\tau^*\) and \(T_{50}\) is expected, how to explain the proportionality factor remains an open question.

**Discussion.** — By contrast to other recommendation approaches [13–15,29], network-based recommendation methods do not pay attention to time stamps in the input data and are commonly evaluated on the random probe. We demonstrate that the evaluation on the random probe is fundamentally flawed as follows: 1) it essentially measures how a method reproduces the network topology of the input data and 2) it tends to overestimate the method’s performance in realistic settings. By contrast, the evaluation based on the time probe measures the method’s ability to reflect both network topology as well as the system’s natural growth patterns and the users’ shifting interests. Note that this contributes to a current line of research that questions the real use of recommender systems to users [45].

Facing the reduced recommendation performance of network-based methods under the new evaluation protocol, we propose to enhance the existing time-unaware methods with temporal features and show that even their simple combination dramatically improves the resulting recommendation accuracy. The improvement is also with respect to methods using only temporal features which suggests that to obtain the best performance, it is crucial to combine temporal features and personalized preferences. Besides improved accuracy, we find that time-aware methods tend to recommend items of smaller degree than traditional methods and therefore can be used to improve the degree of novelty and serendipity (the ability to produce unexpected, yet useful results) in recommendation [25,46]. The proposed multiplicative time-dependent term \(\Delta k_\alpha/k_\alpha\) can be used in combination with most network-based recommendation methods. How to combine it with the class of recommendation methods that aim at predicting the rating that a user would give to an item remains an open question. In summary, our work opens the door to the search for even more effective time-aware methods for recommendation and information filtering in general.

**REFERENCES**

[1] LAZER D. et al., Science, 323 (2009) 721.
[2] DONG Y. et al., Link prediction and recommendation across heterogeneous social networks, in Proceedings of the 12th IEEE International Conference on Data Mining (IEEE) 2012, pp. 181–190.
[3] BAO J., ZHENG Y., WILKIE D. and MOKIEL M., GeoInformatica, 19 (2015) 525.
[4] SCHAER J. B., KONSTANT J. and RIEDL J., Recommender systems in e-commerce, in Proceedings of the 1st ACM Conference on Electronic Commerce (ACM) 1999, pp. 158–166.
[5] BELL R. M. and KOREN Y., ACM SIGKDD Explor. Newslett., 9 (2007) 75.
[6] PATHAK B., GARFINKEL R., GOPAL R. D., VENKATESAN R. and YIN F., J. Manag. Inf. Syst., 27 (2010) 159.
[7] BOBADILLA J., ORTEGA F., HERNANDO A. and GUTIERREZ A., Knowledge-Based Syst., 46 (2013) 109.
[8] NEWMAN M., Networks: An Introduction (Oxford University Press) 2010.
[9] ZHANG Y.-C. et al., EPL, 80 (2007) 68003.
[10] ZHOU T., REN J., MEDO M. and ZHANG Y.-C., Phys. Rev. E, 76 (2007) 046115.
[11] ZHOU T. et al., Proc. Natl. Acad. Sci. U.S.A., 107 (2010) 4511.
[12] YU F., ZENG A., GILLARD S. and MEDO M., Physica A, 452 (2016) 192.
[13] DING Y. and LI X., Time weight collaborative filtering, in Proceedings of the 14th ACM International Conference on
