Rozenshtein, Polina; Gionis, Aristides

Temporal PageRank

Published in:
Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2016, Proceedings

DOI:
10.1007/978-3-319-46227-1_42

Published: 01/01/2016

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Rozenshtein, P., & Gionis, A. (2016). Temporal PageRank. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2016, Proceedings* (pp. 674-689). (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Vol. 9852 LNAI). Springer. https://doi.org/10.1007/978-3-319-46227-1_42

Temporal PageRank

Polina Rozenshtein and Aristides Gionis

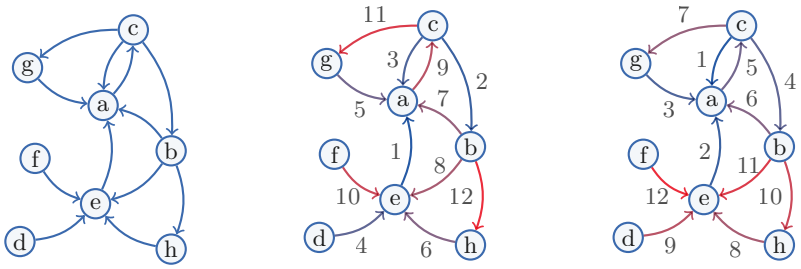
Helsinki Institute for Information Technology
Department of Computer Science
Aalto University, Finland
firstname.lastname@aalto.fi

Abstract. PageRank is one of the most popular measures for ranking the nodes of a network according to their importance. However, PageRank is defined as a steady state of a random walk, which implies that the underlying network needs to be fixed and static. Thus, to extend PageRank to networks with a temporal dimension, the available temporal information has to be judiciously incorporated into the model.
Although numerous recent works study the problem of computing PageRank on dynamic graphs, most of them consider the case of updating static PageRank under node/edge insertions/deletions. In other words, PageRank is always defined as the static PageRank of the current instance of the graph.
In this paper we introduce *temporal PageRank*, a generalization of PageRank for temporal networks, where activity is represented as a sequence of time-stamped edges. Our model uses the random-walk interpretation of static PageRank, generalized by the concept of *temporal random walk*. By highlighting the actual information flow in the network, temporal PageRank captures more accurately the network dynamics.
A main feature of temporal PageRank is that it adapts to concept drifts: the importance of nodes may change during the lifetime of the network, according to changes in the distribution of edges. On the other hand, if the distribution of edges remains constant, temporal PageRank is equivalent to static PageRank.
We present temporal PageRank along with an efficient algorithm, suitable for online streaming scenarios. We conduct experiments on various real and semi-real datasets, and provide empirical evidence that temporal PageRank is a flexible measure that adjusts to changes in the network dynamics.

Keywords: PageRank, graph mining, social-network analysis, dynamic graphs, time-evolving networks, interaction networks

1 Introduction

PageRank is a classic algorithm for estimating the importance of nodes in a network. It has been considered a success story on applying link analysis information seeking and ranking, and has been listed as one of the ten most influential data-mining algorithms [24]. PageRank has been applied to numerous settings



(a) (b) (c)

Fig. 1: (a) A static graph, in which hubs a and e have the highest static PageRank score; (b) and (c) represent two different temporal networks: in (b) the temporal PageRank score of nodes a and e are expected to be stable over time; in (c) node e becomes more important than a as the time goes by, and the temporal PageRank scores of a and e are expected to change accordingly.

and it has inspired a family of fixed-point computation algorithms, such as, TopicRank [6], TrustRank [8], SimRank [11], and more.

PageRank is defined to be the steady-state distribution of a random walk. As such, it is implied that the underlying network structure is fixed and does not change over time. Even though numerous works have studied the problem of computing PageRank on dynamic graphs, the emphasis has been given on maintaining PageRank efficiently under network updates [12, 19], or on computing PageRank efficiently in streaming settings [22]. Instead there has not been much work on how to incorporate temporal information and network dynamicity in the PageRank definition.

To make the previous claim more clear imagine that starting from an initial network G we observe k elementary updates in the network structure $e_1, \ldots, e_k$ (such as edge additions or deletions), resulting on a modified network G'. A typical question is how to compute the PageRank of G' efficiently, possibly by taking into consideration the PageRank of G, and the incremental updates. Nevertheless, the PageRank of G' is defined as a steady-state distribution and as the network G' would "freeze" at that time instance.

Our goal in this paper is to extend PageRank so as to incorporate temporal information and network dynamics in the definition of node importance. The proposed measure, called temporal PageRank, is designed to provide estimates of the importance of a node u *at any given time* t. If the network dynamics and the importance of nodes change over time, so does temporal PageRank, and it duly adapts to reflect these changes.

An example illustrating the concept of temporal PageRank, and presenting the main difference with classic PageRank, is shown in Figure 1. First, a static (directed) graph is shown in Figure 1(a). Vertices a and e are the hubs of the graph, and thus, the nodes with the highest static PageRank score. Figures 1(b)

and (c) show two *temporal networks*; the number next to each edge denotes the time-stamp that the edge arrives. In Figure 1(b) the in-coming edges of nodes a and e are arriving in an interleaving manner, so we expect that the importance of a and e will be stable over time, and that their temporal PageRank scores will be approximately equal to their static PageRank scores. On the other hand, in Figure 1(c) we are witnessing a *concept drift*: node a receives its in-coming edges in the initial phase, while node e receives its in-coming edges later on. Due to this change, node e becomes more important than a as time goes by. Accordingly the scores of temporal PageRank for a and e are changing over time reflecting the change in the network dynamics.

Note also that a dynamic algorithm for computing PageRank is required to report the same output (the static PageRank of the graph in Figure 1(a)) independently of whether it receives its input as in Figure 1(b) or (c).

As illustrated in the previous example, temporal PageRank is defined for *temporal networks* [9,18], i.e., networks with time-stamped edges. We generalize the random-walk interpretation of static PageRank by using *temporal random walks*, i.e., *time-respecting* random walks on the temporal network.

We provide a simple update algorithm for computing temporal PageRank. Our algorithm processes the graph edges in order of arrival and it is proven to converge to the correct temporal PageRank scores. We also prove that if the edge distribution remains constant, temporal PageRank converges to the static PageRank of the underlying graph that the edge distribution is drawn.

We conduct extensive experimental evaluation on various real and semi-real datasets, which support our theoretical results and provide empirical evidence that temporal PageRank is a flexible measure that adjusts to changes in the network dynamics.

2 Models

2.1 Static PageRank

Consider a static weighted directed graph $G_s = (V, E_s, w)$ with n nodes. Let P be the adjacency matrix of G_s, such that each row is normalized to unit sum. To avoid dangling nodes it is typically assumed that the all-zero rows of P are substituted by rows of $1/n$.

Given adjacency matrix $P \in \mathbb{R}^{n \times n}$ and a unit-normalized *personalization* row vector $\boldsymbol{h} \in \mathbb{R}^n$, we consider a random walk that visits the nodes of the graph G_s at discrete steps $i = 1, 2, \ldots$. At step $i = 1$ the random walk starts at a node $u \in V$ with probability $h(u)$. Given that at step i the random walk has visited a node u, at step $i + 1$ it visits a node v selected as follows: with probability $1-\alpha$ the node v is chosen according to the distribution $\boldsymbol{h}$, while with probability α the node v is chosen according to the distribution specified by the u-th row of P.

Consider now a *Markov chain* with nodes V as its *state space* and *transition matrix*

$$P' = \alpha P + (1 - \alpha)\mathbf{1}\boldsymbol{h},$$

where $\mathbf{1}$ is a unit column vector. This Markov chain models the random walk defined above. Assuming that the matrix P' is stochastic, aperiodic, and irreducible, by the Perron–Frobenius theorem there exists a unique row vector π, such that $\pi P' = \pi$ and $\pi\mathbf{1} = \mathbf{1}$. The vector π is the stationary distribution of the Markov chain, and it is also known as the *PageRank vector*. The u-th coordinate of π is the *PageRank score* of node u.

A closed-form expression for π can be derived as

$$\pi = (1-\alpha)\boldsymbol{h}(I-\alpha P)^{-1} = (1-\alpha)\boldsymbol{h}\sum_{k=0}^{\infty}\alpha^k P^k,$$

and the PageRank score of a node u can be written as

$$\begin{aligned}\pi(u) &= \sum_{v\in V} h(v)\sum_{k=0}^{\infty}(1-\alpha)\alpha^k \sum_{\substack{z\in Z(v,u)\\ |z|=k}}\prod_{(i,j)\in z} P(i,j)\\ &= \sum_{v\in V}\sum_{k=0}^{\infty}(1-\alpha)\alpha^k \sum_{\substack{z\in Z(v,u)\\ |z|=k}} h(v)\Pr[z \mid v]\\ &= \sum_{v\in V}\sum_{k=0}^{\infty}(1-\alpha)\alpha^k \sum_{\substack{z\in Z(v,u)\\ |z|=k}} \Pr[z], \end{aligned} \tag{1}$$

where $Z(v,u)$ is a set of all walks from v to u, and (i,j) is used to denote two consecutive nodes of a certain walk $z \in Z(v,u)$. The product $\prod_{(i,j)\in z} P(i,j) = \Pr[z \mid v] = \Pr[z]/h(v)$ expresses the probability that a random walk reaches node u, provided that it starts at node v and it follows only graph edges.

In the definition of PageRank, it is assumed that the transition probability matrix P' is given in advance, and it does not change. A number of works address the problem of computing PageRank *incrementally*, when nodes and edges are added or removed. However, PageRank is still defined by its static version, as the stationary distribution of the graph that contains all nodes and edges that are currently *active* [4, 5, 12, 19]. Here we propose another view of PageRank, where temporal information and network dynamics are explicitly incorporated in the underlying random walk that defines the PageRank distribution.

2.2 Temporal PageRank

Temporal PageRank extends static PageRank by incorporating temporal information into the random-walk model. Our model uses *temporal networks* [9, 13, 18, 21]. A temporal network $G = (V, E)$ consists of a set of n nodes V and a set of m timestamped edges (or interactions) E between pairs of nodes

$$E = \{(u_i, v_i, t_i)\}, \text{ with } i = 1, \ldots, m, \text{ such that } u_i, v_i \in V \text{ and } t_i \in \mathbb{R},$$

where t_i represents the timestamp when an interaction between u_i and v_i is taking place. For generality we assume that the edges of the temporal graph

are *directed*. We also assume more than one different edge may exist between a given pair of nodes, with different timestamps, representing multiple interactions in time between a pair of nodes.

Following previous studies on temporal networks [9, 18], given a temporal network G, we define a *temporal walk* on G, or a *time-respecting walk*, to be a sequence of edges (u_1, u_2, t_1), (u_2, u_3, t_2), ..., (u_j, u_{j+1}, t_j), such that $t_i \leq t_{i+1}$ for all $1 \leq i \leq j-1$.

Our extension of static PageRank to temporal PageRank is based on modifying the PageRank definition of Equation (1) so that only temporal walks are considered instead of all possible walks.

The intuition behind the idea can be illustrated by the example shown in Figure 1(c). Node a initially receives many in-links and it should be considered important. After time $t = 8$, however, it does not receive any more in-links and thus, its importance should diminish. By using time-respecting walks one can accurately model the fact that the probability of the random walk being at node a decreases as time increases beyond time $t = 8$. Essentially, the probability that a random walk being at node a after time $t = 8$ corresponds to the probability that the random walk has arrived at node a *before* time $t = 8$ and it has not left yet. Clearly this probablity decreases as time increases beyond $t = 8$.

We now define temporal PageRank more formally. Let $Z^T(v, u \mid t)$ be a set of all possible *temporal walks* that start at node v and reach node u before time t. We can compute the probability of a particular walk $z \in Z^T(v, u \mid t)$ as the number $c(z \mid t)$ of all such walks (starting at v and reaching u before time t) normalized by a number of all temporal walks that start at node v and have the same length

$$\Pr'\left[z \in Z^T(v, u \mid t)\right] = \frac{c(z \mid t)}{\sum_{\substack{z' \in Z^T(v,x|t) \\ x \in V,\ |z'|=|z|}} c(z' \mid t)}. \tag{2}$$

To compute the number $c(z \mid t)$ of temporal walks that start at v and reach u before time t one can consider the unweighted count of all possible temporal walks. Such a count implies that once reaching u at time t_1 the random walk selects uniformly at random one of the future interactions (u, x, t_2), with $t_2 > t_1$, to move out of u. This model is not very intuitive as it assumes that the random walk has knowledge of the future interations. Instead, once reaching u by an interaction (v, u, t_1) it is more likely to move out of u in one of the next interactions (u, x, t_2). Thus, we assume that the probability of taking (v, u, t_1) followed by (u, x, t_2) increases as the time difference $(t_2 - t_1)$ decreases.

To model this decreasing probability we consider an exponential distribution. Our motivation for this definition is the exponential-decay model in data-stream processing, which is commonly used. We define the probability that interaction (v, u, t_1) is followed by (u, x, t_2):

$$\Pr\left[(v, u, t_1), (u, x, t_2)\right] = \beta^{|(u,y,t') | t' \in [t_1, t_2],\ y \in V|}.$$

We will refer to β as transition probability. The weighted number of temporal walks is then defined as

$$c(z \mid t) = (1-\beta) \prod_{((u_{i-1},u_i,t_i),(u_i,u_{i+1},t_{i+1}))\in z} \beta^{|(u_i,y,t')|t'\in[t_i,t_{i+1}],\ y\in V|},$$

where $(1-\beta)$ is a normalization term. Note that $\beta = 1$ with omitted normalization corresponds to the unweighted case. In this case we view temporal network as a sequence of samples from some unknown and changing distribution P'.

By combining Equations (1) and (2), the *temporal PageRank* score of a node u at time t is defined as

$$\mathbf{r}(u,t) = \sum_{v\in V}\sum_{k=0}^{t}(1-\alpha)\alpha^k \sum_{\substack{z\in Z^T(v,u|t)\\ |z|=k}} \Pr'[z \mid t]\,. \tag{3}$$

Note that according to this definition, the temporal PageRank score of a node u is a function of time. Thus, although our definition is an adaptation of the path-counting formulation of static PageRank (Equation (1)), the temporal PageRank is not a limiting distribution as static PageRank.

Also note that the definition of temporal PageRank (Equation (3)) does not incorporate explicitly a personalization vector $\boldsymbol{h}$. Instead, in the temporal PageRank model presented above, the probability of starting a temporal walk at a node u is proportional to the number of temporal edges that start in u. The vector that contains the starting probabilities for all nodes is referred to as *walk starting probability vector* and it is denoted by $\boldsymbol{h'}$. The vector $\boldsymbol{h'}$ is learned from the data, in particular, for each node u, it is $h'(u) = \frac{|(u,v,t)\in E:\ \forall v\in V|}{|E|}$.

On the other hand, given a personalization vector $\boldsymbol{h}^*$, the personalized temporal PageRank is defined as

$$\mathbf{r}(u,t) = \sum_{v\in V}\sum_{k=0}^{t}(1-\alpha)\alpha^k \frac{h^*(v)}{h'(v)} \sum_{\substack{z\in Z^T(v,u|t)\\ |z|=k}} \Pr'[z \mid t] \tag{4}$$

Equation (4) assumes that the walk starting probability vector $\boldsymbol{h'}$ is known. In practice, $\boldsymbol{h'}$ can be learned by one scan of the edges of the temporal network.

3 Algorithms

3.1 Computing temporal PageRank

In order to compute temporal PageRank we need to process the sequence of interactions E and calculate the weighted number of temporal walks. When a new interaction (u,v,t) arrives it can be used to advance any of the temporal walks that end in u, or it can be the start of a new walk. To keep count of the number of walks ending at each node we use an *active mass* vector $\boldsymbol{s}(t) \in \mathbb{R}^{|V|}$, with $\boldsymbol{s}(u,t)$ being equal to the weighted count of walks ending at node u at time t. We also use a vector $\boldsymbol{r}(t) \in \mathbb{R}^{|V|}$ to keep temporal PageRank estimates,

Algorithm 1: stream processing

```
   input  : E, transition probability β ∈ (0, 1], jumping probability α
 1 r = 0, s = 0;
 2 foreach (u, v, t) ∈ E do
 3     r(u) = r(u) + (1 − α);
 4     s(u) = s(u) + (1 − α);
 5     r(v) = r(v) + s(u)α;
 6     if β ∈ (0, 1) then
 7         s(v) = s(v) + s(u)(1 − β)α;
 8         s(u) = s(u)β;
 9     else if β = 1 then
10         s(v) = s(v) + s(u)α;
11         s(u) = 0;
12 normalize r;
13 return r;
```

where $\boldsymbol{r}(u,t)$ stores the value of temporal PageRank (t-PR) of node u at time t. Algorithm 1 processes a sequence of interactions E, updates the counts $\boldsymbol{s}(t)$ and $\boldsymbol{r}(t)$ for each new interaction (u,v,t), and outputs $\boldsymbol{r}$ as a t-PR estimate.

Proposition 1. *Algorithm 1 computes temporal PageRank defined in Eq. (3).*

Proof. Algorithm 1 counts explicitly the weighted number of temporal walks. Lines 3 and 4 correspond to initiating a new walk with probability $1-\alpha$. With probability α the last interaction is chosen to continue active walks that wait in node u (line 5). Line 7 (or 10, depending on transition probability β) increments the active walks (active mass) count in the node v with appropriate normalization $1-\beta$. Line 8 (or 11) decrements the active mass count in node u. If the transition probability is $\beta = 1$, then the random walk chooses the first suitable arrived interaction to continue the walk. □

Algorithm 1 processes all interactions E in *one* pass and $\mathcal{O}(n)$ space. We need $\mathcal{O}(1)$ space per node, leading to total $\mathcal{O}(n)$ space, while every interaction initiates a constant number of updates, leading to $\mathcal{O}(1)$ update time per interaction.

To compute personalized temporal PageRank for a given personalization $\boldsymbol{h}^*$ we perform normalization, defined by Equation (4), and multiply terms $(1-\alpha)$ in lines 3 and 4 by $\frac{h^*(u)}{h'(u)}$. Unless we know the distribution of temporal edges in advance, we need to learn $\boldsymbol{h}'$. Thus, we obtain a 2-pass algorithm to calculate personalized temporal PageRank for a given personalization vector $\boldsymbol{h}^*$.

3.2 Temporal vs. static PageRank

Temporal PageRank is defined to handle network dynamics and concept drifts. An intuitive property that one may expect is that if the edge distribution of the

temporal edges remains constant, then temporal PageRank approximates static PageRank. In this section we show that indeed this is the case.

Consider a weighted directed graph $G_s = (V, E_s, w)$ and a time period $\mathcal{T} = [1, .., T]$. Without loss of generality assume $\sum_{e \in E_s} w(e) = 1$ and let $N_{\text{out}}(u)$ be the out-link neighbors of u. Let edges $e \in E_s$ be associated with a sampling distribution $\mathcal{S}_E : p[e = (u, v)] = w(e)$. A temporal graph $G = (V, E)$ is constructed by sampling T edges from G_s using $\mathcal{S}_E$ (probability to pick an edge into E is proportional to the weight of this edge in the static graph). We will consider a simple case of transition probability $\beta = 1$: a random walk takes the first suitable interaction to continue.

In the setting described above we can prove the following statement.

Proposition 2. *The expected values of temporal PageRank on graph $G = (V, E)$ converge to the values of static PageRank on graph $G_s = (V, E_s, w)$, with personalization vector $h(u) = \sum_{v \in N_{\text{out}}(u)} w(e = (u, v))$ (weighted out-degree).*

Proof. At any time moment t every vertex $u \in V$ has PageRank score $\boldsymbol{r}(u, t)$ and active mass (number of walkers that wait to continue) equal to $\boldsymbol{s}(u, t)$.

The expected value $\mathbb{E}(\boldsymbol{r}(v, T))$ of the PageRank count of node v at time T is a sum over expected increments of $\boldsymbol{r}(v)$ over time:

$$\mathbb{E}(\boldsymbol{r}(v, T)) = \sum_{t=1}^{T} \mathbb{E}(\Delta r(v, t)).$$

At time t the increment of $\boldsymbol{r}(v)$ can be caused by selecting an edge $e(t) = (v, q)$ with starting point in v and $q \in V$. In this case $\boldsymbol{r}(v)$ is incremented by $(1 - \alpha)$. Another possibility to increment $\boldsymbol{r}(v)$ is to select an edge $e(t) = (q, v)$ with u as an end point and $q \in V$. In this case $\boldsymbol{r}(v)$ is incremented by $\alpha \boldsymbol{s}(q, t)$, where $\boldsymbol{s}(q, t)$ is a value of active mass in node q at time t. Let $p(u)$ be a probability that sampled interaction has u as its start point. Note, that

$$p(u) = \frac{\sum_{j \in V} w(e = (u, j))}{\sum_{i \in V} \sum_{j \in V} w(e = (i, j))},$$

that is, the normalized out-degree of u. Thus, $\mathbb{E}(\Delta \boldsymbol{r}(v, t))$ can be written as

$$\mathbb{E}(\Delta \boldsymbol{r}(v, t)) = (1 - \alpha) p(v) + \alpha \sum_{u \in V} p(u) p(v|u) \mathbb{E}(\boldsymbol{s}(u, t)).$$

To calculate expected amount of active mass in $\boldsymbol{s}(u, t)$, notice that $\boldsymbol{s}(u, t)$ equals to total increments of $\boldsymbol{r}(u)$ happened between the time moment, when edge with starting point in u was selected to update, and t:

$$\mathbb{E}(\boldsymbol{s}(u, t)) = \Delta \boldsymbol{r}(u, t) p(u) + (\Delta \boldsymbol{r}(u, t) + \Delta \boldsymbol{r}(u, t-1)) p(u)(1 - p(u)) + \dots$$
$$\cdots + p(u)(1 - p(u))^{t-1} \sum_{t'=0}^{t-1} \Delta \boldsymbol{r}(u, t - t') = \sum_{t'=0}^{t-1} \mathbb{E}(\Delta \boldsymbol{r}(u, t - t')) p(u) \sum_{k=t'}^{t-1} (1 - p(u))^k$$

The inner sum is a geometric progression:

$$\mathbb{E}(\boldsymbol{s}(u,t)) = \sum_{t'=0}^{t-1} \mathbb{E}(\Delta\boldsymbol{r}(u,t-t'))p(u)\frac{1}{p(u)}\left[(1-p(u))^{t'} - (1-p(u))^{t}\right].$$

We sum $\mathbb{E}(\boldsymbol{s}(u,t))$ over time and consider the two summations separately:

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{E}(\boldsymbol{s}(u,t)) = &\sum_{t=1}^{T}\sum_{t'=0}^{t-1} \mathbb{E}(\Delta\boldsymbol{r}(u,t-t'))(1-p(u))^{t'} \\ &- \sum_{t=1}^{T}\sum_{t'=0}^{t-1} \mathbb{E}(\Delta\boldsymbol{r}(u,t-t'))(1-p(u))^{t}.\end{aligned}$$

The first summation term can be written as:

$$\sum_{t=1}^{T}\sum_{t'=0}^{t-1} \mathbb{E}(\Delta\boldsymbol{r}(u,t-t'))(1-p(u))^{t'} = \sum_{t=1}^{T} \mathbb{E}(\Delta\boldsymbol{r}(u,t)) \sum_{t'=0}^{T-t}(1-p(u))^{t}.$$

The second summation term is:

$$\sum_{t=1}^{T}\sum_{t'=0}^{t-1} \mathbb{E}(\Delta\boldsymbol{r}(u,t-t'))(1-p(u))^{t} = \sum_{t=1}^{T} \mathbb{E}(\Delta\boldsymbol{r}(u,t))(1-p(u))^{t} \sum_{t'=0}^{T-t}(1-p(u))^{t}.$$

Putting the parts together:

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{E}(s(u,t)) &= \sum_{t=1}^{T} \mathbb{E}(\Delta\boldsymbol{r}(u,t))(1-(1-p(u))^{t}) \sum_{t'=0}^{T-t}(1-p(u))^{t} \\ &= \sum_{t=1}^{T} \mathbb{E}(\Delta\boldsymbol{r}(u,t))(1-(1-p(u))^{t})\frac{1}{p(u)}(1-(1-p(u))^{T-t+1}).\end{aligned}$$

Now the expected total increment $\mathbb{E}(\boldsymbol{r}(v,T))$ can be expressed as:

$$\mathbb{E}(\boldsymbol{r}(v,T)) =$$
$$(1-\alpha)\sum_{t=1}^{T} p(v) + \alpha \sum_{u\in V} p(v|u) \sum_{t=1}^{T} \mathbb{E}(\Delta\boldsymbol{r}(u,t))(1-(1-p(u))^{t})(1-(1-p(u))^{T-t+1}).$$

We need to show that

$$\lim_{T\to\infty} \frac{\mathbb{E}(\boldsymbol{r}(v,T))}{T} = (1-\alpha)p(v) + \alpha \lim_{T\to\infty} \sum_{u\in V} p(v|u)\frac{\mathbb{E}(\boldsymbol{r}(u,T))}{T}. \tag{5}$$

Let us upper-bound $\mathbb{E}(\Delta\boldsymbol{r}(v,t))$. Consider a time moment $t' \le t$. A value of mass introduced to the system at t' is $(1-\alpha)$. This mass can arrive to the node v at time moment t through a sequence of $t-t'$ steps of transmission (when a node u, which currently holds this mass, was chosen for action) or retainment (a node u was not chosen for action and the mass remains in u). Transmission

happens with probability $p(u)\alpha$; the probability of retainment is $1-p(u)$. Define $p = \max_{v \in V}\{1-p(v), \alpha p(v)\}$. Then the expected value remained from this mass is upper-bounded by $(1-\alpha)p^{(t-t')}$. The sum of all introduced bits of mass is an upper-bound for the active mass expected to enter node v at time t:

$$\mathbb{E}(\Delta \boldsymbol{r}(v,t)) \leq \sum_{t'=1}^{T}(1-\alpha)p^{t'} \leq (1-\alpha)\frac{p(1-p^t)}{1-p} \leq \frac{1}{1-p}$$

Now we need to show that the following limit goes to 0:

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}(\Delta \boldsymbol{r}(u,t))((1-p(u))^{T+1} - (1-p(u))^{t} - (1-p(u))^{T-t+1})$$
$$= \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}(\Delta \boldsymbol{r}(u,t))(1-p(u))^{T+1} - \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}(\Delta \boldsymbol{r}(u,t))(1-p(u))^{t}$$
$$- \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}(\Delta \boldsymbol{r}(u,t))(1-p(u))^{T-t+1}$$

Consider three limits separately. The first one:

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}(\Delta \boldsymbol{r}(u,t))(1-p(u))^{T+1} = \lim_{T\to\infty}\frac{(1-p(u))^{T+1}}{T}\sum_{t=1}^{T}\mathbb{E}(\Delta \boldsymbol{r}(u,t))$$
$$\leq \lim_{T\to\infty}\frac{p^{T+1}}{T}\sum_{t=1}^{T}\frac{1}{1-p} = 0$$

The second one:

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}(\Delta \boldsymbol{r}(u,t))(1-p(u))^{t} \leq \lim_{T\to\infty}\frac{1}{T(1-p)}\sum_{t=1}^{T}p^{t} = \lim_{T\to\infty}\frac{1}{T(1-p)}\frac{p-p^{T+1}}{1-p} = 0$$

The third one:

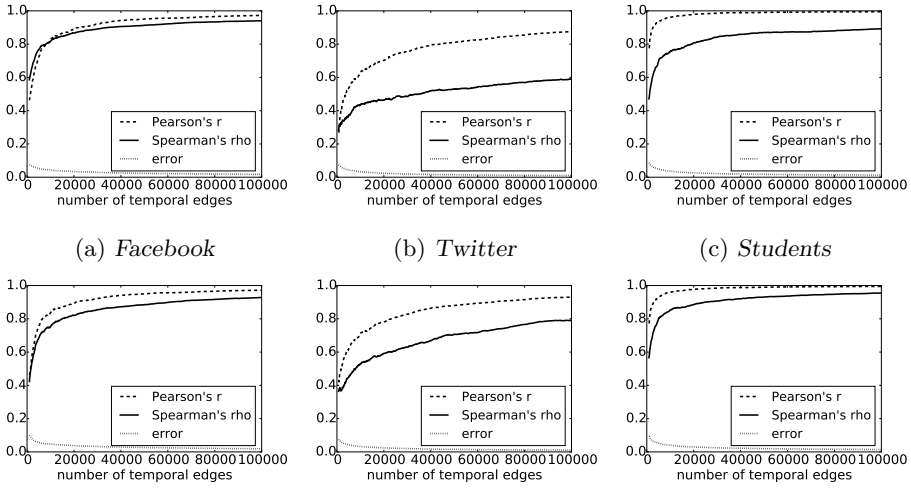
$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}(\Delta \boldsymbol{r}(u,t))(1-p(u))^{T-t+1} \leq \lim_{T\to\infty}\frac{p^{T+1}}{T(1-p)}\sum_{t=1}^{T}p^{-t} = \lim_{T\to\infty}\frac{p^{T+1}}{T(1-p)}\frac{p^{-T}-1}{1-p}$$
$$\leq \lim_{T\to\infty}\frac{p}{T(1-p)^2} = 0$$

It follows that Expression (5) is true.

Now, if we define $pr(v) = \lim_{T\to\infty}\frac{1}{T}\mathbb{E}(r(v,T))$, then Expression (5) can be written as personalized PageRank in a steady state:

$$pr(v) = (1-\alpha)p(v) + \alpha\sum_{u\in V}p(v|u)pr(u)$$

□



(a) *Facebook* (b) *Twitter* (c) *Students*

(d) *Facebook* (e) *Twitter* (f) *Students*

Fig. 2: Convergence of Temporal PageRank to static PageRank. The first row (2a, 2b, 2c) corresponds to degree personalization, the second row (2d, 2e, 2f) corresponds to random personalization, given a priori.

4 Experimental evaluation

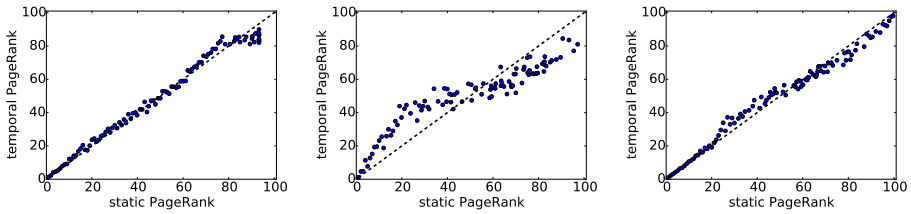
To further support our theoretical analysis, we provide an empirical evaluation of temporal PageRank. The implementation of all algorithms and scripts are publicly available.[1] We first describe our experimental setup.

Datasets. We consider semi-real temporal networks, constructed by using real-world directed networks with edge weights equal to the frequency of corresponding interaction. In particular, we consider the following networks: *Facebook*, *Twitter* and *Students*. For each such network we extract static subgraphs $G_s = (V, E_s, w)$ with $n = 100$ nodes, obtained by BFS from a random node. We normalize edge weights w to sum to 1. Then we sample a sequence of temporal edges E, such that each edge $e \in E_s$ is sampled with probability proportional to its weight $w(e)$; the distribution of sampled edges is denoted by $\mathcal{S}_{E(w)}$. The number of temporal edges E is set to $m = 100\,\mathrm{K}$.

The *Facebook* dataset is a 3-month subset of Facebook activity in a New Orleans regional community [23]. The dataset contains an anonymized list of wall posts (interactions). The *Twitter* dataset tracks activity of Twitter users in Helsinki during 08.2010–10.2010. As interactions we consider tweets that contain mentions of other users. The *Students* dataset[2] is an activity log of a student on-

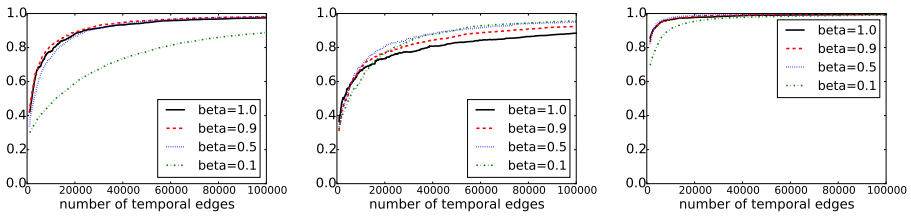
[1] `https://github.com/polinapolina/temporal-pagerank`

[2] `https://toreopsahl.com/datasets/online_social_network`



(a) *Facebook* (b) *Twitter* (c) *Students*

Fig. 3: Comparison of temporal PageRank ranking with static PageRank ranking, degree personalization is used.



(a) *Facebook* (b) *Twitter* (c) *Students*

Fig. 4: Rank quality (Pearson corr. coeff.) and transition probability β.

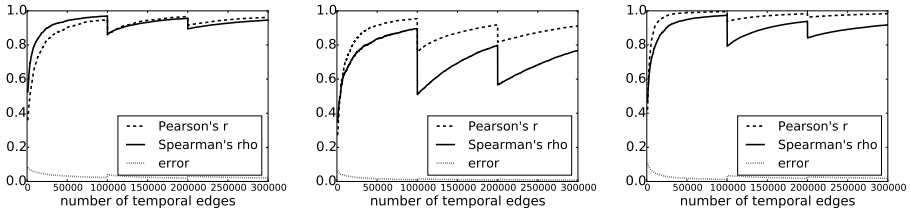
line community at the University of California, Irvine. Nodes represent students and edges represent messages.

Measures. To evaluate the settings in which temporal PageRank is expected to converge to the static PageRank of a corresponding graph, we compare temporal and static PageRank using three different measures: we use (i) Spearman's ρ to compare the induced rankings, we also use (ii) Pearson's correlation coefficient r, and (iii) Euclidean distance ϵ on the PageRank vectors.

All the reported experimental results are averaged over 100 runs. Damping parameter is set of $\alpha = 0.85$. Waiting probability β for temporal PageRank is set to 0 unless specified otherwise.

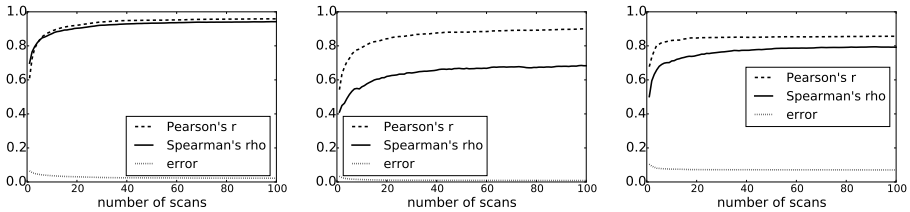
4.1 Results

Convergence. In the first set of experiments we test how fast the temporal PageRank algorithm converges to corresponding static PageRank. In this setting we process datasets with m temporal edges and compare the temporal PageRank ranking with the corresponding static PageRank ranking. In the plots of Figure 2 we report Pearson's r, Spearman's ρ and Euclidean error ϵ. The first column corresponds to the calculation of temporal PageRank without



(a) *Facebook* (b) *Twitter* (c) *Students*

Fig. 5: Adaptation for the change of sampling distribution.



(a) *Facebook* (b) *Twitter* (c) *Students*

Fig. 6: Convergence to static PageRank with increasing number of random scans of edges.

any a priori knowledge of personalization vector. Thus, the resulting temporal PageRank corresponds to the static PageRank with out-degree personalization: $\boldsymbol{h}(u) = \sum_{v \in N_{\text{out}}(u)} w(u, v)$, where $N_{\text{out}}(u)$ are out-link neighbors of u. The second column shows convergence in the case when the personalization vector $\boldsymbol{h}^*$ is given and appropriate renormalization of t-PR counts is taking place.

The plots in Figure 2 show that in both variants of personalization the behavior is similar: in most cases the correlation of the PageRank counts reaches high values already after 20 K temporal edges. Pearson's r is remarkably high, while Spearman's ρ is typically lower. This can be explained by the large number of discordant pairs in the tail of ranking — due to producing a power-law distribution PageRank is known to give robust rankings only at the top of the ranking list. The Euclidean error ϵ also decreases to near-zero values fast.

In Figure 3 we show direct comparison between rankings, obtained by static and temporal PageRanks after processing all temporal edges. We observe that the rank correlation is high for top-ranked nodes and decreases towards the tail of ranking.

Transition probability β. In this experiment we evaluate the dependence of the resulting ranking and the speed of convergence on the transition probability β. The plots in Figure 4 show that lower transition probability β corresponds

to slower convergence rate. On the other hand, smaller values of β produce better correlated rankings. This behavior is intuitive, as a lower value for β implies accumulation of more information regarding the possible walks, which in turn implies a slower convergence rate.

Adaptation to concept drifts. In this experiment we test whether temporal PageRank is adaptive to concept drifts. We start with a temporal network sampled from some static network $G_s^1 = (V, E_s, w_1)$. After sampling m temporal edges E_1, we change the weights of the static graph and sample another m temporal edges E_2 from $G_s^2 = (V, E_s, w_2)$. A final sequence of m edges E_3 is sampled from $G_s^3 = (V, E_s, w_3)$. We run our algorithm on the concatenated sequence $E = \langle E_1, E_2, E_3 \rangle$, without a priori personalization. On Figure 5 we report correlation with the corresponding ground-truth static PageRank. The transition probability β is set to 0.5. In all cases, temporal PageRank is able to adapt to the changing distribution quite fast. Note however, that the previous history is not completely eliminated and for each change of the distribution an increasing number of edges is required to reach a certain correlation level.

Random scans. In the last experiment, given a static graph $G_s = (V, E_s, w)$ we generate a sequence of temporal E by scanning the edges E_s in random order several times. Figure 6 shows that as the number of scans increasing, our estimate for temporal PageRank converges to the static PageRank of the graph. We see that the correlation obtains high values even after a few (around 10) scans. This experiments suggests a very simple and efficient algorithm to compute the static PageRank of a graph, by running our algorithm on a small number of linear scans (randomly ordered) on the graph edges.

5 Related work

PageRank is one of the most popular measures for ranking the nodes of a network according to their importance. The original idea was introduced by Page and Brin [20] for application to web search, and since then it is widely used as a graph-mining tool. As the size of typical networks has increased significantly over the last years, and as networks tend to grow and evolve fast, research on designing scalable algorithms for computing PageRank is still active [16].

A different line of research is dedicated to efficient approaches for updating PageRank in dynamic and/or online scenarios [4, 5, 12, 19, 22]. The term "dynamic" is typically used to refer to the model of edge additions and deletions. However, we discussed in the introduction, even in these dynamic settings PageRank is defined as a stationary distribution over a static graph (the current graph). Another research direction uses temporal information to calculate weights of edges of a static graph [10, 17].

On the contrary, temporal PageRank intends to capture the continuous interaction between individuals. Temporal PageRank is defined over temporal networks [9, 18], where each edge has an associated time-stamp recording an interaction at that point. To our knowledge there is no published work, which considers temporal generalization of PageRank. The closest work is dedicated

to Bonacich's centrality [15]. It focuses on empirical study of a citation network with coarse snapshots, aggregated over a year. In contrast, we are interested in theoretical relation between temporal and static PageRanks and test our methods on several networks with fine granularity.

The static Pagerank definition has multiple interpretations, extensively discussed in a survey by Langville et al. [14]. Our definition of temporal PageRank has a random walk-based interpretation inspired by the one given for static PageRank [3]. Methodologically, the closest papers to our work, are Monte-Carlo simulation algorithms [2] and PageRank calculation by local updates [1, 7].

6 Concluding remarks

We proposed a generalization of static PageRank for the case of temporal networks. The novelty of our approach relies on the fact that we explicitly take into account the exact time that nodes interact, which leads to more accurate ranking. The main feature of the generalization is that it takes into account structural network changes, and models the fact that the importance of nodes may change during the lifetime of the network, according to changes in the distribution of edges. Additionally, we showed that if the distribution of edges remains stable, the temporal PageRank converges to the static PageRank. We provided an efficient algorithm to calculate temporal PageRank and demonstrated its quality and convergence rate through multiple experiments on diverse datasets.

Acknowledgements. This work is partially supported by the Academy of Finland project "Nestor" (286211) and the EC H2020 RIA project "SoBigData" (654024).

References

1. R. Andersen, F. Chung, and K. Lang. Local graph partitioning using pagerank vectors. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 475–486. IEEE, 2006.
2. K. Avrachenkov, N. Litvak, D. Nemirovsky, and N. Osipova. Monte carlo methods in pagerank computation: When one iteration is sufficient. *SIAM Journal on Numerical Analysis*, 45(2):890–904, 2007.
3. R. Baeza-Yates, P. Boldi, and C. Castillo. Generalizing pagerank: Damping functions for link-based ranking algorithms. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 308–315. ACM, 2006.
4. B. Bahmani, A. Chowdhury, and A. Goel. Fast incremental and personalized pagerank. *Proceedings of the VLDB Endowment*, 4(3):173–184, 2010.
5. B. Bahmani, R. Kumar, M. Mahdian, and E. Upfal. Pagerank on an evolving graph. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 24–32. ACM, 2012.
6. I. Berlocher, K.-i. Lee, and K. Kim. Topicrank: bringing insight to users. In *SIGIR*, 2008.

7. M. Bianchini, M. Gori, and F. Scarselli. Inside pagerank. *ACM Transactions on Internet Technology (TOIT)*, 5(1):92–128, 2005.
8. Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *VLDB*, 2004.
9. P. Holme and J. Saramäki. Temporal networks. *Physics reports*, 519(3):97–125, 2012.
10. W. Hu, H. Zou, and Z. Gong. Temporal pagerank on social networks. In *International Conference on Web Information Systems Engineering*, pages 262–276. Springer, 2015.
11. G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *KDD*, 2002.
12. K. S. Kim and Y. S. Choi. Incremental iteration method for fast pagerank computation. In *Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication*, page 80. ACM, 2015.
13. R. Kumar, T. Calders, A. Gionis, and N. Tatti. Maintaining sliding-window neighborhood profiles in interaction networks. In *Machine Learning and Knowledge Discovery in Databases*, pages 719–735, 2015.
14. A. N. Langville and C. D. Meyer. Deeper inside pagerank. *Internet Mathematics*, 1(3):335–380, 2004.
15. K. Lerman, R. Ghosh, and J. H. Kang. Centrality metric for dynamic networks. In *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*, pages 70–77. ACM, 2010.
16. P. A. Lofgren, S. Banerjee, A. Goel, and C. Seshadhri. Fast-ppr: Scaling personalized pagerank estimation for large graphs. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1436–1445. ACM, 2014.
17. B. Manaskasemsak, P. Teerasetmanakul, K. Tongtip, A. Surarerks, and A. Rungsawang. Computing personalized pagerank based on temporal-biased proximity. In *Information Technology Convergence*, pages 375–385. Springer, 2013.
18. O. Michail. An introduction to temporal graphs: An algorithmic perspective. In *Algorithms, Probability, Networks, and Games*, pages 308–343, 2015.
19. N. Ohsaka, T. Maehara, and K.-i. Kawarabayashi. Efficient pagerank tracking in evolving networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 875–884. ACM, 2015.
20. L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: bringing order to the web. 1999.
21. P. Rozenshtein, N. Tatti, and A. Gionis. Discovering dynamic communities in interaction networks. In *Machine Learning and Knowledge Discovery in Databases*, pages 678–693, 2014.
22. A. D. Sarma, S. Gollapudi, and R. Panigrahy. Estimating pagerank on graph streams. *Journal of the ACM (JACM)*, 58(3):13, 2011.
23. B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi. On the evolution of user interaction in facebook. In *Proceedings of the 2nd ACM workshop on Online social networks*, pages 37–42. ACM, 2009.
24. X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip, et al. Top 10 algorithms in data mining. *KAIS*, 2008.