Efficient algorithm to compute Markov transitional probabilities for a desired PageRank

Gábor Berend^{1,2*}

*Correspondence:

berendg@inf.u-szeged.hu

¹Institute of Informatics, University of Szeged, Árpád tér 2., Szeged, Hungary

²SZTE-MTA Research Group on Artificial Intelligence, Tisza Lajos krt. 103., Szeged, Hungary

Abstract

We propose an efficient algorithm to learn the transition probabilities of a Markov chain in a way that its weighted PageRank scores meet some predefined target values. Our algorithm does not require any additional information about the nodes and the edges in the form of features, i.e., it solely considers the network topology for calibrating the transition probabilities of the Markov chain for obtaining the desired PageRank scores. Our experiments reveal that we can reliably and efficiently approximate the probabilities of the transition matrix, resulting in the weighted PageRank scores of the nodes to closely match some target distribution. We demonstrate our findings on both quantitative and qualitative evaluations by reporting experimental results on web traffic (the English Wikipedia and a Hungarian news portal) and the bicycle sharing network of New York City.

Keywords: Markov chains; PageRank scores; Random walk; Wikipedia

1 Introduction

Complex networks provide a ubiquitous and natural way to model real-world processes and phenomena including but not limited to social interactions [1–6], natural language [7–9] or biomedicine [10, 11], inter alia. In the steadily evolving area of network science, PageRank [12] and its vast number of variants, including [8, 11, 13, 14], admittedly belong to the most popular approaches of network analysis.

Given a network with its topology, PageRank algorithm determines a probability distribution over its nodes by calculating the stationary distribution of a random walk which is assumed to make the decision about which node to visit next based on local vicinity information of nodes in a *uniform* manner, i.e., inversely proportional to the number of its neighboring nodes.

From a real-world web browsing perspective, this assumption implies that users are believed to click on hyperlinks located at some website with equal probability, which most likely is not the case in reality. Substantial research has been conducted on biasing the transition probabilities of random walk models in an application-specific manner in order to obtain modified PageRank scores for the nodes of certain types of networks [15–21].

© The Author(s) 2020. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

In this work, we take a different approach as our goal is not to devise such transition probabilities which yield useful PageRank scores for a particular application. Our goal instead is to set the weights of the transition matrix such that it converges to a predefined PageRank vector provided as input. This latter problem has been in the focus of [22, 23]. In this paper, we propose an efficient algorithm—being linear in the number of edges of the input graph—for the above task, which we shall refer as the reverse PageRank problem.

From an application-oriented point of view, finding such transition probabilities which results in a predefined PageRank distribution for some Markov chain can help in designing better routing protocols. For instance, one might want to spread around some resource over a communication or social network with the additional (soft) constraint that all the nodes should have an even (or similar) amount of access to the resource in the long run. In such a scenario, we are interested in predicting the transition probabilities of the network such that the desired distribution of the PageRank scores over the nodes is as uniform as possible, i.e., the random walk visits nodes independently of their centrality within the graph.

Another potential use case for the investigated problem is to gain insights into the underlying principles of some partially observable Markov chain. Placing the problem in the original context of the PageRank algorithm, even if we have information about the relative popularity of the individual websites, it is not necessary that we also have access to the actual flow of network traffic between pairs of websites (hence partial observability). We might, however, want to still be able to approximate click-through rates between websites in the absence of access to web server logs with referral statistics. In such a scenario, solving for the reverse PageRank problem, we can approximate the transition probabilities between websites such that the stationary distribution over them closely matches our initial beliefs for the relative popularity of the websites. Our large scale experiment on English Wikipedia is precisely of this form, i.e., we trained our model such that it attempts to predict click-through rates between pairs of Wikipedia articles. In order to do so, we were merely relying on the overall popularity of the articles.

The main contributions of this paper can be summarized in the followings:

- we propose an efficient algorithm to calculate the approximate gradient for our problem, which can be calculated in linear time as a function of the number of possible transitions of the Markov chain,
- we illustrate the effectiveness of the proposed algorithm via rigorous quantitative and qualitative experiments on three real-world complex networks, i.e., the English Wikipedia, a Hungarian news portal and the bicycle sharing network of New York City,
- we release our source code at <https://github.com/begab/reversePageRank> in order to foster reproducibility and additional experimentation.

2 Illustration of the problem

This section provides a small example which demonstrates the problem and its difficulty. Suppose that we have a graph $G = (V, E)$ as displayed in Fig. 1. Let us further assume that there is some resource traversing the network, and this resource should be present at vertices A through F with probabilities proportional to the radii of the solid circles marking them, i.e., 0.26, 0.24, 0.12, 0.09, 0.16 and 0.13, respectively. Now the question that we face is the following: how should we guide the resource, so that its expected presence over the nodes (marked by dashed circles) match the desired input distribution (solid circles)?

Figure 1(a) illustrates that letting the resource to choose its upcoming destination uniformly at random from the directly accessible neighboring states—as it is done in vanilla PageRank algorithm—can provide some states with excess importance (e.g., states *C* and *D*) and leave some of the states under-represented at the same time (cf. states *A* and *E*).

An alternative to the uniform routing scheme could be if we chose succeeding states proportional to the desired stationary distribution of the neighboring states. As Fig. 1(b) depicts, however, such a strategy might not provide any real benefit over uniform routing.

Figure 1(c) demonstrates the behavior of our proposed algorithm. Note that in this figure the radii of the dashed and solid circles practically concur as opposed to Fig. 1(a) and Fig. 1(b). This means that performing resource routing according to the proposed transition probabilities would result in a stationary distribution closely matching the desired one.

For this previous motivating example our algorithm is able to find a close to perfect solution, however, we should note that in the general case it is possible that for a certain network topology and expected PageRank scores no perfect solution exists even in theory. We deal with the general solvability of our problem in Sect. 5.3.

3 Related work

There have also been previous work with similar motivation to ours [22, 23], i.e., to determine transition weights in some complex network. The core motivation in [22] and [23] are identical to ours, however, they employ different approaches.

Inspired by Luce's choice theory [22–24] learn weights for the vertices of some network in an iterative fashion and infer transitional probabilities for the edges based on them. The reliance on Luce's choice theory lets [22, 23] to learn only $O(|V|)$ parameters for some network $G = (V, E)$ and use those parameters for inferring the $O(|E|) > O(|V|)$ transition probabilities. Note that the classical Metropolis–Hastings algorithm [25, 26] is also applicable to obtain state transition probabilities of a Markov chain for some desired stationary distribution, however, as stated in [22], “very little is known about it from a formal point of view, especially, its mixing time and rate of convergence”. ChoiceRank [23] is based on and involves an iterative EM-type inference algorithm for trying to recover the marginal popularity of the nodes in the network provided as an input to the algorithm. Unlike [22], the authors of [23] do not require the input network to be strongly connected.

There exist further work on the estimation of transition probabilities in Markov chains that assume temporal access to observing transitions from the network [27, 28]. [29] proposed the *inverse Markov chain* problem, in which the authors try to predict the number of transitions in a transportation network in a partially observable setting. The problem formulation in [29] is similar to ours, but also differs in important aspects. [29] assumes that the visit frequencies for a subset of vertices $V_o \subset V$ are observed, whereas our algorithm requires a distribution (the PageRank vector) defined for all the vertices. Due to the different setting, the loss function employed in [29] also differs from the one we utilize. Partial observability in [29] also apply to the edge transition frequencies over $V_o \times V_o$, i.e., the algorithm has an additional (and optional) term in the loss function regarding the actual edge transition frequencies over a subset of the edges in the network. Another difference is that the objective function applied in [29] was optimized using the natural gradient, whereas we used a quasi-Newton optimization technique, which suites better problems of the scale our experiments involved. Perhaps most importantly, the calculation of the gradient of the objective in [29] is quadratic in the number of (observable) vertices, which makes its application prohibitive even for moderate sized networks.

The Supervised Random Walk approach [2] aims at learning an appropriate weighting function over the edges of complex networks by minimizing the sum of margin-based soft losses between positively and negatively labeled nodes. The margin-based loss employed in [2] includes the stationary distribution of random walks with restarts from multiple source nodes. [2] assumes that each edge (u, v) can be described by a low dimensional feature vector ψ_{uv} , whereas our model does not require access to any additional meta-data in order to calculate edge features. Since we do not regard edge weights as a parameterized function of feature values, our proposed approach is also applicable in scenarios when edge features, such as *edge age*—an example feature used in [2]—are not accessible.

Various studies have been conducted recently related to Wikipedia about trying to predict missing links [30, 31], improving and analyzing navigability [32–35]. More generally, the field of network tomography [36] focuses on statistical methods that can be used for making inferences about unknown edge properties of (typically computer) networks that could be based on node-level aggregated information. Network tomography applications vary vastly, i.e., the measurements may be collected passively by monitoring natural traffic

flows or actively by generating probe traffic. Our setting belongs to the former category of passive approaches. Typical network tomography approaches, such as [37, 38] for recovering the unobserved random variables operate via Expectation Maximization and Markov Chain Monte Carlo Methods.

4 Notation and background

In this section, we briefly describe the original PageRank algorithm and also introduce our notation for the rest of the paper. Markov chains can be used to define stochastic processes by providing the state space $\mathcal{S} = \{S_1, \dots, S_n\}$ and a row stochastic transition matrix $P \in \mathbb{R}^{n \times n}$, the p_{ij} element of which can be interpreted as the probability of observing state S_j in time t given that state S_i was observed at time $t - 1$. The stationary distribution of a Markov chain characterized by transition matrix P can be determined by calculating the left principal eigenvector of P .

Given a graph $G = (V, E)$, the PageRank algorithm determines a distribution of importance over the nodes in the network that we denote by π . We use subscripts for indexing a particular element of a vector. That is, π_v denotes the PageRank score for node v .

The PageRank model assumes that the prestige of any node depends on the importance of those nodes which link to it in a recursive manner. Formally, π_v can be defined as

$$\pi_v = \sum_{(u,v) \in E} p_{uv} \pi_u, \quad (1)$$

where p_{uv} denotes the probability of traversing from state u to v . For some node u , p_{uv} is uniformly set to $\frac{1}{d_u}$, d_u denoting the outdegree of node u . According to the matrix formulation, the PageRank vector is the stationary distribution of the Markov chain parameterized by P , which is guaranteed to uniquely exist for ergodic Markov chains. In order to handle non-ergodic networks as well, Page et al. [12] suggests the following calculation for the PageRank scores:

$$\pi_v = \frac{\beta}{|V|} + (1 - \beta) \sum_{(u,v) \in E} \frac{1}{d_u} \pi_u, \quad (2)$$

where the damping factor β is typically chosen from the interval $[0.1, 0.2]$. This extension ensures the Markov chain to be aperiodic and irreducible as a non-zero probability (proportional to the damping factor) is reserved for direct transitioning between any pair of nodes in the network. Equation (2) reveals that all the outgoing edges of a particular node are still weighted in a uniform manner, i.e., proportional to the outdegree of that node. In the following, we introduce our model for learning edge-specific transition weights for a given target distribution over the network.

5 Reverse PageRank algorithm

We next formulate the model utilized for learning the transition weights of Markov processes with predefined PageRank scores. We coin our approach as the Reverse PageRank algorithm that we shall refer as REVERSEPR for short. Throughout the rest of the paper, we denote the desired PageRank vector we wish our weighted random walk with restarts to converge by π^* .

In the following, we denote the parameters we optimize for and the particular PageRank scores the nodes receive after convergence by Θ and $\pi(\Theta)$, respectively. The edge weight associated for a particular edge $(u, v) \in E$ is going to be proportional to $\exp(\theta_{uv})$. More concretely, the probability of directly traversing between any pair of nodes $(u, v) \in V \times V$ is given by

$$p_{uv}(\Theta) = \begin{cases} \frac{\exp(\theta_{uv})}{\sum_{(u,k) \in E} \exp(\theta_{uk})}, & \text{if } (u, v) \in E, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

As the above formulation is invariant to a constant additive factor, that is $p_{uv}(\Theta) = p_{uv}(\Theta + c)$ for any $c \in \mathbb{R}$, infinitely many Θ can yield the same p_{uv} values. In order to overcome this and to decrease the number of variables in the model by $O(|V|)$, we ensure that for each node u , there is one parameter in θ_{u*} , i.e., among the parameters controlling for the transition probabilities starting from node u , which is constantly kept at the value of zero (cf. the comment in line 11 of Algorithm 1).

Our overall goal is thus to find a feasible value of Θ for which $\pi(\Theta)$ lies as close to π^* as possible. More formally, given a graph $G = (V, E)$, the objective function we seek to minimize is

$$\min_{\Theta} \text{KL}(\pi^* \parallel \pi(\Theta)) = \min_{\Theta} \sum_{v \in V} \pi_v^* \log \frac{\pi_v^*}{\pi_v(\Theta)}, \quad (4)$$

i.e., the Kullback–Leibler (KL) divergence between distributions π^* and $\pi(\Theta)$.

As $\text{KL}(\pi^* \parallel \pi(\Theta)) = H(\pi^*, \pi(\Theta)) - H(\pi^*)$ and the second term—being the entropy of the target PageRank vector—does not depend on Θ , we alternatively opt for the equivalent optimization problem

$$\max_{\Theta} -H(\pi^*, \pi(\Theta)) = \max_{\Theta} \sum_{v \in V} \pi_v^* \log \pi_v(\Theta), \quad (5)$$

with $H(\pi^*, \pi(\Theta))$ denoting the cross entropy between π^* and $\pi(\Theta)$.

Algorithm 1 Efficient computation of the approximate gradient of Eq. (5)

```

1: function CALCULATEAPPROXIMATEGRADIENT( $G, \Theta, \beta$ )
2:   for all  $(u, v) \in E$  do
3:      $g[(u, v)] \leftarrow 0$  ▷ initialization
4:    $\pi \leftarrow \text{WEIGHTEDPAGERANK}(G, \Theta, \beta)$  ▷  $\beta$  denotes the damping factor
5:   for all  $u \in V$  do
6:      $T \leftarrow 0$ 
7:     for all  $v \in \text{ADJACENTVERTICES}(G, u)$  do
8:        $T \leftarrow T + p_{uv}(\Theta) \frac{\pi_v^*}{\pi_v}$ 
9:      $i \leftarrow 0$ 
10:    for all  $v \in \text{ADJACENTVERTICES}(G, u)$  do
11:      if  $i > 0$  then ▷ we keep one of the  $\theta_{u*}$  values as 0
12:         $g[(u, v)] \leftarrow \pi_u p_{uv}(\Theta) (\frac{\pi_v^*}{\pi_v} - T)$ 
13:    return  $i \leftarrow i + 1$ 

```

In case of Eq. (5), the partial derivative with respect to a single parameter θ_{ij} has the form

$$\frac{\partial -H(\pi^*, \pi(\Theta))}{\partial \theta_{ij}} = \sum_{v \in V} \frac{\pi_v^*}{\pi_v(\Theta)} \frac{\partial \pi_v(\Theta)}{\partial \theta_{ij}}, \quad (6)$$

revealing that we need to sum over all the vertices in order to calculate a single partial derivative. Furthermore, since $\pi_v(\Theta)$ is recursively entangled with θ_{ij} , the partial derivative within the summation would require the application of a PageRank-style iterative approach [2, 39]. As a consequence, the gradient of Eq. (5) can be calculated in $O(|E| \cdot (|E| + |V|))$ time, implying that the exact calculation of the gradient is beyond practical applicability even for moderate sized graphs.

In order to circumvent the difficulty of calculating the gradient of Eq. (5), we propose an alternative approximate gradient which can be computed in a highly effective $O(|E|)$ manner. As stated earlier, the main bottleneck during the computation of the gradient of Eq. (5) arises from the presence of $\frac{\partial \pi_v(\Theta)}{\partial \theta_{ij}}$ in Eq. (6). Due to the recursive formulation of PageRank, $\pi_v(\Theta)$ can be expressed as $\sum_{(u,v) \in E} p_{uv}(\Theta) \pi_u(\Theta)$, which means that

$$\frac{\partial \pi_v(\Theta)}{\partial \theta_{ij}} = \sum_{(u,v) \in E} \pi_u(\Theta) \frac{\partial p_{uv}(\Theta)}{\partial \theta_{ij}} + p_{uv}(\Theta) \frac{\partial \pi_u(\Theta)}{\partial \theta_{ij}}. \quad (7)$$

In our approximation, we make the simplifying assumption that the π_u values are not dependent on Θ , canceling this way out the (otherwise computationally still expensive) second term from Eq. (7). Together with the fact that the transition probabilities parameterized by Θ have the partial derivative

$$\frac{\partial p_{uv}(\Theta)}{\partial \theta_{ij}} = \begin{cases} -p_{ij}(\Theta) \cdot p_{ij}(\Theta) + p_{ij}(\Theta), & \text{if } u = i \text{ and } v = j, \\ -p_{ij}(\Theta) \cdot p_{iv}(\Theta), & \text{if } u = i \text{ and } v \neq j, \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

the partial derivative from Eq. (6) can be conveniently approximated as

$$\sum_{(u,v) \in E} \frac{\pi_v^*}{\pi_v(\Theta)} \pi_u(\Theta) \frac{\partial p_{uv}(\Theta)}{\partial \theta_{ij}} = \pi_i(\Theta) p_{ij}(\Theta) \left(\frac{\pi_j^*}{\pi_j(\Theta)} - \sum_{(i,v) \in E} p_{iv}(\Theta) \frac{\pi_v^*}{\pi_v(\Theta)} \right). \quad (9)$$

The nice property of Eq. (9) is that its value can be calculated in $O(1)$ amortized time. This is due to the fact that the last sum in Eq. (9) can be reused for the calculation of multiple partial derivatives—i.e., for any of the variables θ_{i*} . This observation implies that the entire gradient of the objective function can be calculated in $O(|E|)$ time as also illustrated in Algorithm 1.

5.1 Details on the optimization

A key property of optimization algorithms is their time needed for convergence. As opposed to first order methods, second order methods are known to have the ability of converging faster to an optimum. The application of second order methods for optimizing Eq. (5), however, would be strictly prohibitive. This is because second order methods rely on

the inverse of the Hessian of the objective function which has $O(|E|)$ parameters according to our problem formulation—with $|E|$ potentially being in the range of 10^9 s for large real-world networks.

In order our algorithm to be applicable for large-scale networks as well, we applied the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) [40] quasi-Newton optimization technique. The usage of L-BFGS is advantageous since it tries to mimic the behavior and the fast convergence of second order methods, while only requiring the computation and storage of the gradients of the objective function up to a fixed number of previous iterations.

Similar to first-order optimization methods, L-BFGS also has hyperparameters, such as the step size for standard gradient descent, however, we did not investigate tuning these hyperparameters for obtaining better results. Instead, we relied on the default hyperparameter settings in this regard of the MALLET package that we used for performing the optimization.

Confirmed by our preliminary experiments, relying on the calculation of the proposed approximate gradient not only sped up the optimization process by orders of magnitude, but we also managed to obtain better solution in terms of our objective function introduced in (5). Our conjecture for the latter phenomenon is that the proposed approximate gradient is strictly smaller than the actual gradient, thus missing good local optima during the optimization of the non-convex objective function was less likely.

5.2 The non-convexity of the objective

For an illustration of the non-convexity of the objective function already with a single parameter, see Fig. 2. Note that the non-convexity of the objective implies that different initializations for Θ can result in convergence to different local optima. According to our empirical evidences—that we discuss in more details in Sect. 6.3.2—the quality of the local optima according to multiple evaluation criteria did not differ substantially.

Furthermore, we have experienced that initializing Θ homogeneously with all zeros can be regarded as a generally robust way to initialize model parameters. We should also add that the optimization procedure we employ is deterministic in the sense that once initialized identically, it always converges to the same (local) optimum when applying the same hyperparameters for L-BFGS (that we kept constant at their default values throughout our experiments).

5.3 The general solvability of the problem

We finally comment on the existence and the general solvability for the proposed problem formulation. A trivial solution always exists—i.e., choosing p_{uv} proportional to π_v^* —for fully connected networks with π^* obeying the inequality $\frac{\beta}{|V|} \leq \min_{v \in V} \pi_v^*$. Note that when the inequality is violated, then the node corresponding to the smallest value in π^* already receives a larger than desired amount of distribution based on its share from the damping factor ($\frac{\beta}{|V|}$), hence a perfect solution cannot be given. On the other hand, for any fully connected network obeying the above inequality, it suffices to set all the rows of its corresponding transition matrix identically proportional to $\pi^* - \frac{\beta}{|V|}$ in order to obtain an objective value of zero.

In the more common scenario, when we are not given a fully connected network, an optimal solution with a KL-divergence of zero might not exist for certain input distributions. It can be easily verified that whenever node u has a single outgoing edge towards node v , $\pi_v(\Theta) \geq \pi_u(\Theta)$ should always follow. If the desired distribution behaves contradictory for such a pair of vertices, that is $\pi_v^* < \pi_u^*$, then a perfect solution in terms of KL-divergence cannot be given. Figure 3 illustrates a further example for which no perfect solution can be determined, even though it obeys the previously stated necessity condition.

5.4 Comparison to existing algorithms

REVERSEPR differs from [22, 23] in that it treats the transition probability of every edge in the network as a separately tuneable parameter. As the number of edges are typically larger than the number of vertices, it makes the problem formulation of REVERSEPR underdetermined. [22] and [23] follows a different route, as they learn a separate parameter for each node in the network and they infer the edge transition probabilities based on the node-level parameters they learn.

Another difference between ChoiceRank and REVERSEPR is that the former receives the aggregate incoming traffic and outgoing traffic of each individual node as input, whereas for the REVERSEPR algorithm it suffices to have access solely to π^* . Unlike [29], ChoiceRank and REVERSEPR share the common trait that they both rely on the full topology, i.e., the entire edge structure are assumed to be observable (without the actual traffic flowing over them).

Finally, the algorithm in [22] assumed that the input network is strongly connected. Note that REVERSEPR does not require such an assumption as it conveniently turns any input network into ergodic by employing a damping factor similar to the PageRank algorithm.

6 Experiments

This section contains our experimental results conducted over the English Wikipedia, a Hungarian news portal and the bicycle sharing network of New York City. We used the machine learning package MALLET [41] to perform the L-BFGS optimization over our objective function.

As already mentioned in Sect. 4, PageRank implementations often employ a damping factor β to overcome difficulties arising with non-ergodic networks. Unless stated otherwise, we conducted all our experiments using the default damping factor of $\beta = 0.01$. We purposely chose such a small damping factor in order not to severely alter the random walk applied over the *original* network, while ensuring ergodicity.

6.1 Baselines

In order to measure the effectiveness of the proposed algorithm, we compare it to several strong baselines. As our algorithm makes use of the topology of the network (E) and the expected PageRank scores (π^*), we derived various baselines relying on these sources of information. The different approaches utilized in our experiments are summarized in Table 1.

The *Random* strategy is the most simplistic of all as it assigns transition probabilities in an uniform manner when determining it for some node. The *Indegree* strategy assigns the transition probabilities based on the number of indegrees some node has. As the number of incoming edges can be a good proxy to the hubness of the nodes, we anticipated this kind of heuristic to work particularly for transportation networks in particular, such as the Citibike network.

The heuristic dubbed *Jaccard* takes into consideration the extent to which the neighborhood of some potential target node v overlaps with the source node u when providing predictions for p_{uv} . Note that this strategy can only assign some nonzero probability between pairs of nodes which have at least one neighbor in common. Another property of the Jaccard heuristic is that it assigns higher values for pairs of nodes for which there is little difference in their direct neighbors. This strategy seems less useful for transportation networks, i.e., from a real world perspective one would expect low traffic between airports from which the directly accessible destinations are highly similar. For networks, where semantics matter more, such as the Wikipedia network, the Jaccard baseline is expected to provide more accurate predictions, since high similarity in the neighborhood of nodes can be an indication towards high topical relatedness.

The *Traffic* baseline simply chooses the next node to visit based on the marginal popularity of the nodes, proportional to π^* . Obviously, the choice which is the most popular in general is not necessarily the most popular one in every situation.

Table 1 Overview of the different approaches for approximating p_{uv}

Approach	
Random	$p_{uv} \propto 1$
Indegree	$p_{uv} \propto \{(k, v) \in E\} $
Jaccard	$p_{uv} \propto \frac{ \{(k, v) \in E \wedge (u, k) \in E\} }{ \{(k, v) \in E\} }$
Traffic	$p_{uv} \propto \pi_v^*$
(unweighted) PageRank	$p_{uv} \propto \pi_v$
ChoiceRank [23]	$p_{uv} = \frac{\lambda_v}{\sum_{(u, v') \in E} \lambda_{v'}} \quad (\forall \lambda_i \in \mathbb{R}_{>0})$ [24]
Reverse PageRank	$p_{uv} \propto \exp(\theta_{uv})$

Predictions for the *PageRank* approach are determined based on the PageRank scores of such a random surfer, which performs random walk in an unweighted manner, i.e., following the next node to visit uniformly at random.

Our baselines also include *ChoiceRank* [23], a recent algorithm building upon Luce's choice theorem [24] for identifying network traffic preferences. We relied on the publicly available open-source implementation of the algorithm^a during our experiments. ChoiceRank operates on the marginal indegree and outdegree sequence of the nodes and tries to recover them by calibrating a score for each node in the network. As our proposed algorithm has sole access to a desired distribution over the nodes, we treated the indegree and outdegree of some node u as $\sum_{(v,u) \in E} \pi_v^*$ and $\sum_{(u,v) \in E} \pi_v^*$, respectively.

Although the model proposed in [22] is also intended to solve our task, we did not consider it as a potential baseline, since its assumption on the strong connectedness of the input network does not generally hold for the networks used in our experiments.

6.2 Evaluation metrics

For some node u , we measure the quality of the predicted distribution over its outgoing edges by comparing it to the ground truth transition probability distribution from that node. We rely on four distinct evaluation measures that we describe below.

Following [23], we apply the normalized link displacement scores and KL-divergence between the ground truth and the predicted distributions for evaluation. Normalized link placement for some node u is calculated as

$$\frac{1}{|\{v : (u, v) \in E\}|^2} \sum_{(u,v) \in E} |\sigma_u(v)^* - \sigma_u(v)|, \quad (10)$$

where $\sigma_u(v)^*$ and $\sigma_u(v)$ returns the ordered rank of node v within the neighbors of node u according to the ground truth and the predicted transition probabilities, respectively. The KL-divergence that we use for evaluating the quality of the learned transition probabilities gets calculated for some node u in the network as

$$\sum_{\{v : (u,v) \in E\}} p_{uv}^* \log \frac{p_{uv}^*}{p_{uv}}. \quad (11)$$

An important property of (11) is that it is only defined for such cases when $p_{uv}^* > 0$ also implies $p_{uv} > 0$ to be true. That is, we cannot measure the KL-divergence when there is a non-zero ground truth transition probability, but a model assigns zero probability for it. This is hence a major shortcoming for the Jaccard similarity based heuristic that it can only assign a non-zero transition probability between pairs of nodes which have at least one neighbor in common. Since this criterion was not fully met for all the nodes involved in our experiments, we were not able to calculate the KL-divergence based evaluation for the Jaccard baseline.

Besides the aforementioned scores, we also measure the quality of the predicted transition probability distributions by quantifying their root mean squared error (RMSE) as in [22]. For some node u , the RMSE of the predicted transitions is calculated according to

$$\sqrt{\frac{1}{|\{v : (u, v) \in E\}|} \sum_{\{v : (u,v) \in E\}} (p_{uv}^* - p_{uv})^2}.$$

Additionally, we devise a fourth evaluation criterion, namely mean reciprocal rank (MRR) having its roots in the information retrieval community. This criterion is intended to measure the extent to which a model has the ability of ranking those transitions high that are actually the most frequent ones according to the ground truth data. That is, for a given node u , we calculate the value

$$\frac{1}{\sigma_u(\arg \max_{\{v:(u,v) \in E\}} p_{uv}^*)},$$

with $\sigma(\cdot)$ denoting the same function as in (10), i.e., it returns the rank of the node in its argument relative to the predicted transition probabilities with respect to u . MRR is then obtained by averaging the individual scores for the different nodes.

6.3 Experiments on the link structure of Wikipedia

For these experiments, we take the hyperlink structure of the English Wikipedia and try to estimate its click-through rates solely based on the distribution that we derive from the page visit statistics of the individual articles.

We conducted our experiments on the 20160305 dump of the English Wikipedia,^b which consist of more than $1.23 \cdot 10^7$ articles and $3.87 \cdot 10^8$ hyperlinks. Figure 4 illustrates the typical long tail degree distribution of the Wikipedia network we had access to.

As an additional resource in our evaluation, we relied on the publicly available raw page visit statistics^c for calculating π^* . That is, we derived the desired distribution over the pages based on the individual page visit statistics of the Wikipedia articles.

The running time of REVERSEPR was approximately 40 minutes on a 2.00 GHz Intel Xeon E7-4820 processor for the above described Wikipedia network. For ease of reproducibility, we make the Wikipedia network as well as the π^* vector that we based our experiments on readily accessible.^d

6.3.1 Predicting the most likely cross-article clicks

For evaluation purposes, we utilize the Wikipedia Clickstream dataset [42] which includes referral statistics for March 2016 over the entire English Wikipedia. This valuable source of information makes it possible to calculate p_{uv}^* , i.e., the ground truth click-through rate between a pair of Wikipedia articles (u, v) .

Obviously, predicting the most likely article to visit next is much easier (purely by chance) for articles with a low outdegree as opposed to articles with a large number—potentially thousands—of neighboring articles to choose from. For this reason, we report our performance metrics as a function of the outdegree of the articles.

Results are reported in Fig. 5 according to the metrics already introduced in Sect. 6.2. It can be seen in Fig. 5, that the rankings produced by the REVERSEPR algorithm is dominantly better compared to the alternative baselines.

The baseline which relies on the popularity of articles can only produce reliable predictions for articles with extremely low outdegree and mostly for those evaluations that focus more on the ranking capabilities of the methods instead of the accurateness of the actual transition probabilities.

In particular, the popularity-based *Traffic* baseline delivers the best performance in terms of its replacement and MRR scores for articles not having more than five outgoing links. Its performance, however fades out quickly for articles containing more hyperlinks. Inspecting Fig. 4 reveals that only a minority of Wikipedia articles belong to the category on which the popularity baseline could give reliable predictions.

It is the Jaccard similarity-based approach alone which shows a similar overall ranking ability to REVERSEPR. When approximating p_{uv}^* , the Jaccard baseline compares the extent to which the set of hyperlinks to be found on Wikipedia article u overlap with those included on article v . As a consequence, the Jaccard similarity-based approach makes much less reliable predictions for Wikipedia articles with a small number of outgoing links. The inability of the Jaccard similarity-based heuristic to give accurate predictions for nodes with low degree is most pronouncedly showcased by the RMSE results. The fact that the KL-divergence evaluation cannot be calculated for the Jaccard baseline illustrates another

weakness of that baseline, i.e., it sometimes assigns zero transition probabilities to transitions for which there is a nonzero ground truth transition probability in reality. Figure 5 demonstrates that even random guessing (i.e., the Random baseline) can outperform the Jaccard approach for articles with an extremely low number of outgoing links. We can notice that a large fraction of the Wikipedia articles fall into the category for which Jaccard baseline performs poorly, i.e., half of the Wikipedia articles had an outdegree of at most 61 for the investigated snapshot of Wikipedia (cf. Fig. 4).

6.3.2 Sensitivity analysis of REVERSEPR

Our algorithm is deterministic in the sense, that when initialized by the same parameters, it converges to the same solution. There could nonetheless still be some variability in the results depending on how we initialize our parameters Θ . This potential variability comes from non-convexity of our objective (cf. Sect. 5.2).

As such we cannot guarantee that every initialization of Θ would converge to the same—or even to an equally good—solution. Our default edge transition initialization strategy—that we applied in all our previously reported experiments—is to set the node transition distributions according to the vanilla PageRank algorithm, i.e., we deterministically set all edge probabilities uniformly. This means that initial transition probabilities for some node with outdegree d were initialized to have a value $\frac{1}{d}$. Practically, this means that all parameters of Θ got initialized to zero by our default strategy (that we denote as *Initialization 0* hereon).

In order to assess the sensitivity of REVERSEPR towards the initial choice for Θ , we repeated our experiments by applying random initializations for Θ as well. This strategy simply set the parameters in Θ independently to random values sampled uniformly from $[0, 1]$.

Besides the previously mentioned default initialization (*Initialization 0*), we executed REVERSEPR nine additional times, having Θ randomly initialized as described above. Figure 6 contains the evaluation scores for all the initializations, suggesting that random initialization does not play a crucial role in the performance of REVERSEPR in terms of RMSE and KL-divergence. Furthermore, we can see practically no variability in the results for any of the evaluations between the different *random* initializations of Θ .

Even though the default initialization performs similarly to random initializations for the KL-divergence and RMSE evaluations, they dominantly perform better for the remaining two ranking-oriented metrics, i.e., we can see a clear benefit from initializing Θ according to the all-zeros default strategy in terms of the displacement and MRR scores. Table 2 includes the aggregated numeric metrics for the default initialization strategy as well as the summary statistics (mean and standard deviation) of the metrics for 9 random initializations.

To summarize, random initialization of the parameters we optimize for does not account for a large variability in the quality of the solution REVERSEPR converges to. Furthermore, our deterministic strategy to initialize all parameters to zero seems to provide as good or even better quality in all evaluation aspects, for which reason we return to the usage of our default initialization strategy for all our upcoming experiments.

6.3.3 The effects of damping factor

Another aspect of REVERSEPR which could potentially affect the quality of the solution it converges to is the choice of the damping factor β . We have mentioned earlier that initially,

Table 2 Comparison of the default initialization strategy and alternative random initializations on the final overall evaluation metrics

	Default	Avg. of 9 random initializations	Std. of 9 random initializations
displacement	0.3075	0.3314	4.4e-05
KL	2.8656	2.8673	4.0e-05
RMSE	0.0997	0.0999	6.1e-06
MRR	0.2573	0.1204	8.6e-05

we set the damping factor to 0.01 so that the random walk does not deviate considerably from the true topology of the network. Here we report our results for REVERSEPR when evaluated using four substantially different damping factors, $\beta \in \{0.01, 0.05, 0.1, 0.2\}$.

Figure 7 illustrates the results that we recorded for our experiments when we applying different damping factors. We can draw similar conclusions for we did earlier for the effects of random initializations in Sect. 6.3.2. That is, the choice of the damping factor does not seem to have a decisive impact on the results obtained, REVERSEPR behaves rather robustly for the different β values.

We can nonetheless observe some small differences again for the two metrics which focus on the quality of the rankings and not the actual transition probabilities, i.e., the displacement and the MRR scores. Choices for larger values of β tend to slightly favor low degree nodes, whereas higher degree nodes benefit more from the application of smaller values of β . This can be explained by the fact that nodes with large degree are more likely to be a core member of the network, whereas lower degree nodes tend to be located at the periphery of the network. These are the nodes on the periphery which benefit more from a more frequent restart of the random walk, which corresponds to a higher value

Table 3 The effects of applying different damping factors β

	$\beta = 0.01$	$\beta = 0.05$	$\beta = 0.1$	$\beta = 0.2$
displacement	0.3075	0.3116	0.3158	0.3229
KL	2.8656	2.8656	2.8656	2.8656
RMSE	0.0997	0.0997	0.0997	0.0997
MRR	0.2573	0.2558	0.2507	0.2406

of β . We report the aggregated metrics that we obtained for the different choices of β in Table 3, corroborating that smaller values of β seem to perform better for the ranking-based metrics and that the other two metrics (KL and RMSE) behaved mostly insensitive to the different choices of β .

6.3.4 Qualitative assessment of the different approaches

In this section, we evaluate the different approaches from a qualitative perspective. While these examples are not intended for drawing far-reaching conclusions, they can complement the more rigorous quantitative evaluation and possibly provide additional insights into the behavior of the different approaches.

Since the Jaccard similarity-based approach is heavily relying on the local vicinity of articles, it makes the Jaccard approach less capable for adapting to the changes in temporal user preferences compared to REVERSEPR. This phenomenon is illustrated in Table 4 which lists the titles of Wikipedia articles when following the most likely transitions up to two hops, starting out from the Wikipedia article about *Machine learning*, *Stephen Hawking* and *Brie Larson*.

The articles predicted by REVERSEPR—starting at the article *Machine learning* and traversing through *Deep learning*—illustrates its ability to reflect temporal dynamics of

Table 4 Articles most likely to be visited in a 2-hop session and their respective in starting from three different Wikipedia articles according to the various approaches

(a) Starting from the article entitled 'Machine Learning'

	Hop #1 article	Indegree	Hop #2	Indegree
Ground truth	Artificial neural network	597	Backpropagation	88
Random	Databricks	9	O'Reilly Media	725
Indegree	International Standard Book Number	574,335	Integrated Authority File	200,124
Jaccard	Random forest	108	Out-of-bag error	4
Traffic	Stanford University	12,462	United States	547,915
PageRank	International Standard Book Number	574,335	Digital object identifier	574,335
ChoiceRank	International Standard Book Number	574,335	Canada	167,535
REVERSEPR	Deep learning	154	AlphaGo	13

(b) Starting from the article entitled 'Stephen Hawking'

	Hop #1	Indegree	Hop #2	Indegree
Ground truth	Jane Hawking	19	Lucy Hawking	37
Random	National Health Service	2268	Gibraltar Health Authority	11
Indegree	International Standard Book Number	574,335	Integrated Authority File	200,124
Jaccard	Roger Penrose	637	Alexander Friedmann	251
Traffic	Facebook	20,403	United States	547,915
PageRank	International Standard Book Number	574,335	Digital Object Identifier	169
ChoiceRank	World War II	122,312	United States	547,915
REVERSEPR	The Big Bang Theory	1125	The Big Bang Theory (season 9)	60

(c) Starting from the article entitled 'Brie Larson'

	Hop #1	Indegree	Hop #2	Indegree
Ground truth	Alex Greenwald	57	Phantom Planet	215
Random	Meryl Streep	2060	Tracey Ullman	590
Indegree	United States	54,7915	Geographic coordinate system	962,772
Jaccard	Greer Garson	409	Jennifer Jones	312
Traffic	United States	547,915	Ronald Reagan	9079
PageRank	United States	547,915	International Standard Book Number	547,335
ChoiceRank	United States	547,915	Daylight saving time	204,750
REVERSEPR	Room (2015 film)	142	List of accolades received by Room	2

page visits since the *deep learning*–empowered *AlphaGo* versus Lee Sedol match took place exactly during the investigated time period (March 2016). This increased temporal interest towards *AlphaGo* is also reflected in the prediction of REVERSEPR.

The most likely transition provided by REVERSEPR starting from the article about *Stephen Hawking* can also be explained by temporal dynamics, as Stephen Hawking had a cameo in the U.S. sitcom *The Big Bang Theory* aired on February 25, 2016, which likely increased the respective click-through rate during March 2016. Similarly, the most likely two-hop session predicted by REVERSEPR starting from the Wikipedia article on *Brie Larson* reflects the fact that Brie Larson was awarded as the Best Actress on the Academy Awards on February 28, 2016 for her role in the 2015 movie called *Room*.

Further inspection of Table 4 reveals that multiple baseline approaches favor transitions towards nodes with high indegree centrality, such as *International Standard Book Number*. Not pruning such articles from the network was a deliberate choice in order to preserve the original topology of the network—instead of applying any pragmatic (but somewhat arbitrary) removal of vertices from the network. Table 4 also suggests that REVERSEPR seems to treat nodes with high indegree centrality but low interest better than the baseline approaches.

6.3.5 Experiments on the Clickstream subnetwork of Wikipedia

In our experiments reported so far, we were relying on the entire Wikipedia network from March 2016. Our motivation for not performing any filtering of the vertices or the hyper-link structure was to preserve the “natural topology” of Wikipedia.

The Wikipedia network used in [23] covers the same period of time (March 2016), however, it contains only that subnetwork of the entire Wikipedia that is included in the Clickstream dataset [42]. As a consequence, the Wikipedia network included in [23] is approximately one order of magnitude smaller compared to the network we have reported our experiments earlier. The subnetwork that is purely based on the Clickstream dataset consists of approximately 2.32×10^6 nodes and 1.32×10^7 hyperlinks, whereas the entire Wikipedia network from the same period consists of 1.23×10^7 articles and 3.87×10^8 hyperlinks, respectively.

For the sake of better comparability with the experimental protocol included in [23], we additionally report another set of experiments on the English Wikipedia. This time we restricted our (sub)network to be the one that is included in the Clickstream dataset in the exact same manner how it was done in [23].

The gold standard transition probabilities for our previous experiments also relied on the same Clickstream data, however, the network itself could also include such nodes and edges that were not necessarily part of the Clickstream dataset. For our Clickstream experiments, we determined π^* proportional to the traffic that was included in the Clickstream dataset.

We include the results according to the different metrics obtained for the Clickstream network in Fig. 8. Comparing Fig. 8 with Fig. 5—which includes the same metrics over the large Wikipedia network—we can see similar trends. That is, we can observe that RE-

VERSEPR either performs better than alternative approaches by a fair margin or it is competitive to the rest of the approaches. The metrics are in general better compared to the ones received for the entire Wikipedia network, which can be explained by the fact that the vertices in the Clickstream network have fewer connections on average and making predictions for a smaller-sized neighborhood is an easier task in general.

6.4 Experiments on the Kosarak dataset

The Kosarak dataset^e includes anonymized click-stream data of a Hungarian on-line news portal including 41,001 vertices and 974,560 edges based on 7,029,013 clicks. Kosarak is similar to Wikipedia in the sense that it also contains web traffic data, but it also differs from the Wikipedia in important aspects.

On the one hand, the size of the network induced from the Kosarak dataset is substantially smaller from the hyperlink network of Wikipedia. Additionally, in the case of Kosarak dataset we see a much higher density of edges in contrast to the Wikipedia network, i.e., the fraction of edges and the number of potential edges in the network is $1 * 10^{-6}$ and $4 * 10^{-8}$ for Kosarak and Wikipedia, respectively.

Figure 9 contains a comparison of all the evaluated approaches regarding their performances on the various evaluation metrics on the Kosarak dataset. Figure 5 contains the same evaluation metrics for the Wikipedia network, which lets us compare the two results.

Besides the resemblances, we can also spot differences between Fig. 5 and Fig. 9. We can see for instance that the Traffic baseline achieves the best MRR score on the Kosarak dataset. In contrast, the Traffic baseline was only performant up to articles with medium outdegree for Wikipedia. Another difference is that, even though the Jaccard baseline was

among one of the best performing baselines for Wikipedia, it fails to deliver reasonable performance on the Kosarak network.

The differences can potentially be explained by the different nature of the two networks, i.e., Wikipedia has a different link structure due to its encyclopedic nature compared to the Kosarak network which was obtained from a news portal. A further important difference is that the Kosarak network contains only such edges that were included in at least one browsing session. In contrast when experimenting with Wikipedia, we could exploit the entire network structure (including those links which received no actual traffic). To this end, we regard the Wikipedia-based evaluation more meaningful and realistic.

Interestingly, the performance of the Jaccard approach is similarly good for the Kosarak dataset when evaluated in terms of displacement. What it means that the transition probabilities inferred from the Jaccard heuristic is generally applicable, even though it is not so reliable for determining the single most likely transition compared to other approaches.

The Traffic baseline behaves oppositely on the Kosarak network, i.e., it has the highest displacement scores even though it was performing best in terms of MRR. Again, what it means, that the setting the transition probabilities could help us find the most likely transition, but in general the ranking it provides might be quite dissimilar from reality.

ChoiceRank provides a steady performance on the Kosarak dataset regarding those evaluations which solely focus on the ranking of the neighboring vertices, as it obtains one of the highest MRR and a medium displacement score. REVERSEPR has lower MRR and higher displacement values, however, it outperforms all alternative approaches in terms of the KL-divergence and RMSE metrics by a fair margin. These metrics can be viewed more informative for the Kosarak dataset, where providing a good ranking is much easier in most of the cases, since the median outdegree in this network is only 4. This peculiarity of the dataset explains the good performance of the Random baseline as well.

6.5 Experiments on Citibike network

We additionally considered the dataset of the bicycle sharing system of New York City.^f The dataset contains sessions of bike rentals in the form of pick-up and drop-off station pairs with additional meta-data such as the date and duration for the rides.

We conducted our evaluation on the same network as in [23]. That is, we relied on those source–target destination pairs from year 2015 for which the duration of the ride was no more than one hour and which had an average daily ride frequency of at least one. In the end, we were dealing with a network of 489 nodes and 5209 edges from 3.38 million bike rentals in total.

We performed the same quality assessment on the Citibike network as for the previously analyzed networks. The result of our evaluation is depicted in Fig. 10. Note that REVERSEPR refers to our algorithm using its default version, i.e., when Θ gets initialized to all zeros and the damping factor β equals 0.01.

Figure 10 illustrates that REVERSEPR scored the best for all evaluation metrics when evaluated on the Citibike network. ChoiceRank obtains the second best results for all evaluation criteria, whereas the Jaccard baseline was hardly able to beat even the Random baseline for this network. This observation verifies our anticipation for the Jaccard baseline being less suitable for transportation networks (cf. Sect. 6.1).

By looking at Fig. 10, we can additionally notice that the simple strategy of predicting transition probabilities proportional to the indegree of the neighboring vertices provides

a strong baseline for the Citibike data. Recall that the same baseline had one of the poorest performances when evaluated on the Wikipedia network (cf. Fig. 5).

We can thus conclude that the dynamics driving the bike rental network makes different baselines perform better in predicting the transition probabilities compared to web traffic networks.

In summary, the relative performance of the individual alternative baseline approaches vary remarkably based on the peculiarities of the analyzed networks, whereas REVERSEPR always performed close to or better than the best performing baseline approach in all of our experiments.

6.6 Summary of the experimental results

We finally provide a thorough comparison between REVERSEPR and ChoiceRank in Table 5 both in terms of their speed and overall performance on the three notably different network structures we conducted our experiments on. The median outdegree was 61 for the Wikipedia network and 4 in the case of Kosarak and Citibike datasets.

Table 5(a) summarizes the most important statistics of the different input networks alongside with the amount of time needed for the different approaches to approximate the transition probabilities. We can see that the size and density of the three input networks differ substantially, and that REVERSEPR managed to find a solution at least 1.5–2 times faster than ChoiceRank. We find it remarkable—even if part of the speedup could be accounted for the usage of different programming languages for implementation (Python for ChoiceRank and Java for REVERSEPR)—since the optimization problem solved by ChoiceRank involves only $|V|$ parameters, whereas REVERSEPR optimizes over $O(|E|) \gg |V|$ parameters.

Table 5 A comparative summary of ChoiceRank and REVERSEPR (abbreviated as CR and RPR, respectively)

(a) Network properties and the running times of the algorithms

	$ V $	$ E $	Density ($\frac{ E }{ V ^2}$)	CR running time	RPR running time
Wikipedia	$\approx 1 * 10^7$	$\approx 4 * 10^8$	$3 * 10^{-6}$	5500 seconds	2400 seconds
Clickstream	$\approx 2 * 10^6$	$\approx 1 * 10^7$	$3 * 10^{-6}$	240 seconds	30 seconds
Kosarak	$\approx 4 * 10^4$	$\approx 1 * 10^6$	$6 * 10^{-4}$	4 seconds	2.5 seconds
Citibike	$\approx 5 * 10^2$	$\approx 5 * 10^3$	$2 * 10^{-2}$	0.3 seconds	0.05 seconds

(b) The mean values of the different evaluation criteria for the different networks. Bold means better result and significant differences ($p \ll 0.01$) are marked with †

	Displacement		KL divergence		RMSE		MRR	
	CR	RPR	CR	RPR	CR	RPR	CR	RPR
Wikipedia	0.325	0.307 †	3.005	2.866 †	0.101	0.099 †	0.094	0.257 †
Clickstream	0.299	0.276 †	0.242	0.241 †	0.0979	0.0978 †	0.447	0.517 †
Kosarak	0.332 †	0.359	0.885	0.480 †	0.150	0.074 †	0.554 †	0.431
Citibike	0.267	0.264	0.073	0.070	0.047	0.045	0.439	0.447

(c) The median values of the different evaluation criteria for the different networks. Bold means better result and significant differences ($p \ll 0.01$) are marked with †

	Displacement		KL divergence		RMSE		MRR	
	CR	RPR	CR	RPR	CR	RPR	CR	RPR
Wikipedia	0.324	0.309 †	2.946	2.776 †	0.074	0.073 †	0.045	0.111 †
Clickstream	0.326	0.281 †	0.1433	0.1431	0.076	0.076	0.333	0.5 †
Kosarak	0.344 †	0.378	0.703	0.210 †	0.116	0.046 †	0.5 †	0.333
Citibike	0.283	0.267	0.059	0.058	0.030	0.028	0.333	0.333

Table 5(b) and Table 5(c) compares the mean and median of the evaluation measures that were calculated over all the vertices of the different networks with an out-degree larger than one. We can see that other than for the displacement and MRR scores on the Kosarak dataset, the performance of REVERSEPR is as good as that of ChoiceRank for every combination of evaluation criteria and input network.

We additionally performed paired t-test and Mood's median test to see whether the differences in the mean and median scores of the two approaches differ statistically significantly. We indicate it in Table 5(b) and Table 5(c) when the difference was found to be significant (with $p \ll 0.01$). Using the non-parametric Wilcoxon sign test for checking whether the differences of the means over the vertices differ significantly also yielded the same results. We can see from Table 5(b) and Table 5(c) that all differences are significant for the click-stream networks, i.e., the Wikipedia, Clickstream and Kosarak datasets. ChoiceRank performed significantly better for the displacement and MRR metrics in the case of the Kosarak, whereas the metrics obtained by REVERSEPR were significantly better in all the other cases. For the smallest network, the mean and median evaluation metrics did not differ significantly between the two algorithms, which was likely due to the small size of the network.

As a final comparative assessment, we provide the distribution of the per vertex evaluation scores for all the datasets as a box plot in Fig. 11. We can observe that REVERSEPR is indeed among the best performing models for all networks and evaluation metrics, both in terms of its median and average scores. The advantage of REVERSEPR is the most pronounced for the KL-divergence and RMSE metrics, i.e., in those cases when the actual transitional probabilities are also considered not just the ranking they specify.

The primary benefit of REVERSEPR resides in its efficiency. As also highlighted in Table 5(a), REVERSEPR delivered a 2–8-fold speedup compared to ChoiceRank, with its evaluation metrics being competitive or even better than those of ChoiceRank. We achieved the above-mentioned speedup despite of the fact that our approach has $O(|E|)$ parameters, whereas ChoiceRank operates with $O(|V|) \ll O(|E|)$ parameters.

As our model involves $O(|V|)$ constraints, i.e., one for each element of π^* , having $O(|E|)$ variables makes our problem formulation overparameterized. This overparameterization allows us more flexibility in determining the p_{uv} transitional probabilities, however, it also means that multiple configuration of edge weights could yield the same objective value. A potential solution for mitigating this phenomenon could be to incorporate various regularization terms into the objective for encouraging the transitional probabilities to have a high entropy, or to the contrarily incentivize finding highly skewed transitional probabilities. The kind of inductive bias conveyed by the regularizers could be determined in an application specific manner. We treat the incorporation of various regularizes as a potential future extension of our proposed algorithm.

7 Conclusion

We introduced an efficient algorithm for estimating the transition probabilities between the nodes of complex networks given some desired PageRank distribution over its vertices. We managed to solve the problem by formulating an approximate gradient of our objective

function which can be calculated in linear time relative to the number of edges in the graph.

We illustrated the effectiveness of our proposed approach by evaluating it towards different clickstream datasets (the entire English Wikipedia, its filtered version according to the Clickstream dataset and a Hungarian news portal) and the bicycle sharing network of New York City.

Our proposed algorithm can be naturally extended to other networks, such as citation and collaboration networks, as it does not require any additional meta-data behind the network topology and a desired PageRank distribution.

Acknowledgements

The author is grateful for the funding agencies. The author is also grateful for Gyula Kovács (funded by EFOP-3.6.3-VEKOP-16-2017-0002) for providing code snippets for processing Wikipedia dumps.

Funding

This work was supported by the National Research, Development and Innovation Office of Hungary through the Artificial Intelligence National Excellence Program (grant no.: 2018-1.2.1-NKP-2018-00008). This research was supported by the project "Integrated program for training new generation of scientists in the fields of computer science", no EFOP-3.6.3-VEKOP-16-2017-0002. The project has been supported by the European Union and co-funded by the European Social Fund. This research was supported by grant TUDFO/47138-1/2019-ITM of the Ministry for Innovation and Technology, Hungary. Publication costs were covered by the University of Szeged Open Access Fund (application id. 4100).

Abbreviations

REVERSEPR, Reverse PageRank; KL-divergence, Kullback–Leibler divergence; L-BFGS, limited-memory Broyden–Fletcher–Goldfarb–Shanno; RMSE, Root Mean Squared Error; MRR, Mean Reciprocal Rank.

Availability of data and materials

The [source code](#) and the datasets including the [Wikipedia dump](#), [Wikipedia Clickstream](#) dataset, the [Kosarak](#) dataset and the [Wikipedia network](#) we were using, as well as the [Citibike data](#) used in our experiments are available on-line and also from the author upon request.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

GB designed the study and experiments, implemented the proposed algorithm and wrote the paper. All authors read and approved the final manuscript.

Endnotes

- ^a <https://github.com/lucasmaystre/choix>
- ^b <https://archive.org/details/enwiki-20160305>
- ^c <https://dumps.wikimedia.org/other/pagecounts-raw/>
- ^d http://rgai1.inf.u-szeged.hu/~berend/wikipedia_data/
- ^e <http://fimi.uantwerpen.be/data/>
- ^f The dataset is available from <https://www.citibikenyc.com/system-data>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 22 July 2019 Accepted: 14 July 2020 Published online: 29 July 2020

References

1. Matakos A, Terzi E, Tsaparas P (2017) Measuring and moderating opinion polarization in social networks. *Data Min Knowl Discov* 31(5):1480–1505. <https://doi.org/10.1007/s10618-017-0527-9>
2. Backstrom L, Leskovec J (2011) Supervised random walks: predicting and recommending links in social networks. In: *Proceedings of the fourth ACM international conference on web search and data mining. WSDM '11*. ACM, New York, pp 635–644. <https://doi.org/10.1145/1935826.1935914>
3. Leskovec J, Huttenlocher D, Kleinberg J (2010) Predicting positive and negative links in online social networks. In: *Proceedings of the 19th international conference on world wide web. WWW '10*. ACM, New York, pp 641–650. <https://doi.org/10.1145/1772690.1772756>
4. Newman MEJ (2001) The structure of scientific collaboration networks. *Proc Natl Acad Sci USA* 98(2):404–409. <http://www.pnas.org/content/98/2/404.full.pdf+html>. <https://doi.org/10.1073/pnas.98.2.404>

5. Hasan MA, Chaoji V, Salem S, Zaki M (2006) Link prediction using supervised learning. In: Proc. of SDM 06 workshop on link analysis, counterterrorism and security
6. Tasnádi E, Berend G (2015) Supervised prediction of social network links using implicit sources of information. In: Proceedings of the 24th international conference on world wide web. WWW '15 companion. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, pp 1117–1122. <https://doi.org/10.1145/2740908.2743037>
7. Erkan G, Radev DR (2004) LexRank: graph-based lexical centrality as salience in text summarization. *J Artif Intell Res* 22(1):457–479
8. Mihalcea R, Tarau P (2004) TextRank: bringing order into texts. In: Lin D, Wu D (eds) Proceedings of EMNLP 2004. Association for Computational Linguistics, Barcelona, pp 404–411. <http://www.aclweb.org/anthology/W04-3252>
9. Toutanova K, Manning CD, Ng AY (2004) Learning random walk models for inducing word dependency distributions. In: Brodley CE (Ed) ICML. ACM international conference proceeding series, vol 69. ACM, New York
10. Ji M, He Q, Han J, Spangler S (2015) Mining strong relevance between heterogeneous entities from unstructured biomedical data. *Data Min Knowl Discov* 29(4):976–998. <https://doi.org/10.1007/s10618-014-0396-4>
11. Wu G, Xu W, Zhang Y, Wei Y (2013) A preconditioned conjugate gradient algorithm for GeneRank with application to microarray data mining. *Data Min Knowl Discov* 26(1):27–56. <https://doi.org/10.1007/s10618-011-0245-7>
12. Page L, Brin S, Motwani R, Winograd T (1998) The PageRank citation ranking: bringing order to the web. In: Proceedings of the 7th international world wide web conference, Brisbane, Australia, pp 161–172. <citeseer.nj.nec.com/page98pagerank.html>
13. Gyöngyi Z, García-Molina H, Pedersen J (2004) Combating web spam with TrustRank. In: Proceedings of the thirtieth international conference on very large data bases, VLDB '04, vol 30. VLDB Endowment, Toronto, pp 576–587. <http://dl.acm.org/citation.cfm?id=1316689.1316740>
14. Jeh G, Widom J (2002) Simrank: a measure of structural-context similarity. In: Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining. KDD '02. ACM, New York, pp 538–543. <https://doi.org/10.1145/775047.775126>
15. Richardson M, Domingos P (2002) The intelligent surfer: probabilistic combination of link and content information in PageRank. *Advances in neural information processing systems*, vol 14. MIT Press, Vancouver
16. Hwang H, Hristidis V, Papakonstantinou Y (2006) ObjectRank: a system for authority-based search on databases. In: Proceedings of the ACM SIGMOD international conference on management of data, Chicago, Illinois, USA, June 27–29, 2006, pp 796–798. <https://doi.org/10.1145/1142473.1142593>
17. Nemirowsky D, Avrachenkov K (2008) Weighted PageRank: cluster-related weights. In: Voorhees EM, Buckland LP (eds) TREC, vol. Special publication 500-277, National Institute of Standards and Technology (NIST), Gaithersburg. <http://dblp.uni-trier.de/db/conf/trec/trec2008.html#NemirowskyA08>
18. Tsoi AC, Morini G, Scarselli F, Hagenbuchner M, Maggini M (2003) Adaptive ranking of web pages. In: Proceedings of the twelfth international world wide web conference, WWW 2003, Budapest, Hungary, May 20–24, 2003, pp 356–365. <https://doi.org/10.1145/775152.775203>
19. Haveliwala TH (2002) Topic-sensitive PageRank. In: Proceedings of the 11th international conference on world wide web. WWW '02. ACM, New York, pp 517–526. <https://doi.org/10.1145/511446.511513>
20. Ding Y (2011) Applying weighted PageRank to author citation networks. *J Am Soc Inf Sci Technol* 62(2):236–245. <https://doi.org/10.1002/asi.21452>
21. Xing W, Ghorbani A (2004) Weighted PageRank algorithm. In: Communication networks and services research, 2004. Proceedings. Second annual conference on, pp 305–314. <https://doi.org/10.1109/DNSR.2004.1344743>. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1344743&tag=1
22. Kumar R, Tomkins A, Vassilvitskii S, Vee E (2015) Inverting a steady-state. In: Proceedings of the eighth ACM international conference on web search and data mining. WSDM '15. ACM, New York, pp 359–368. <https://doi.org/10.1145/2684822.2685310>
23. Maystre L, Grossglauser M (2017) Choicerank: identifying preferences from node traffic in networks. In: Proceedings of the 34th international conference on machine learning. ICML 2017, Sydney, NSW, Australia, 6–11 August 2017, pp 2354–2362. <http://proceedings.mlr.press/v70/maystre17b.html>
24. Luce RD (1977) The choice axiom after twenty years. *J Math Psychol* 15(3):215–233
25. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953) Equation of state calculations by fast computing machines. *J Chem Phys* 21(6):1087–1092. <https://doi.org/10.1063/1.1699114>
26. Hastings WK (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57(1):97–109. <http://biomet.oxfordjournals.org/cgi/reprint/57/1/97.pdf>. <https://doi.org/10.1093/biomet/57.1.97>
27. Bai DS (1975) Efficient estimation of transition probabilities in a Markov chain. *Ann Stat*, 3:1305–1317
28. Barsotti F, De Castro Y, Espinasse T, Rochet P (2014) Estimating the transition matrix of a Markov chain observed at random times. *Stat Probab Lett* 94:98–105
29. Morimura T, Osogami T, Ide T (2013) Solving inverse problem of Markov chain with partial observations. In: Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ (eds) Advances in neural information processing systems, vol 26. Curran Associates, Red Hook, pp 1655–1663
30. Adafre SF, de Rijke M (2005) Discovering missing links in Wikipedia. In: Proceedings of the 3rd international workshop on link discovery. LinkKDD '05. ACM, New York, pp 90–97. <https://doi.org/10.1145/1134271.1134284>
31. Paranjape A, West R, Zia L, Leskovec J (2016) Improving website hyperlink structure using server logs. In: Proceedings of the ninth ACM international conference on web search and data mining. WSDM '16. ACM, New York, pp 615–624. <https://doi.org/10.1145/2835776.2835832>
32. Lamprecht D, Dimitrov D, Helic D, Strohmaier M (2016) Evaluating and improving navigability of Wikipedia: a comparative study of eight language editions. In: Proceedings of the 12th international symposium on open collaboration. OpenSym '16. ACM, New York, Article no 17, pp 1–10. <https://doi.org/10.1145/2957792.2957813>
33. Singer P, Lemmerich F, West R, Zia L, Wulczyn E, Strohmaier M, Leskovec J (2017) Why we read Wikipedia. In: Proceedings of the 26th international conference on world wide web, pp 1591–1600. International World Wide Web Conferences Steering Committee
34. West R, Leskovec J (2012) Human wayfinding in information networks. In: Proceedings of the 21st international conference on world wide web. WWW '12. ACM, New York, pp 619–628. <https://doi.org/10.1145/2187836.2187920>

35. West R, Pineau J, Precup D (2009) Wikipedia: an online game for inferring semantic distances between concepts. In: Proceedings of the 21st international joint conference on artificial intelligence. IJCAI'09. Morgan Kaufmann, San Francisco, pp 1598–1603. <http://dl.acm.org/citation.cfm?id=1661445.1661702>
36. Lawrence E, Michailidis G, Nair V, Xi B (2006). Network tomography: a review and recent developments
37. Vardi Y (1996) Network tomography: estimating source-destination traffic intensities from link data. *J Am Stat Assoc* 91(433):365–377
38. Castro R, Coates M, Liang G, Nowak R, Yu B (2004) Network tomography: recent developments. *Stat Sci* 19(3):499–517. <https://doi.org/10.1214/088342304000000422>
39. Andrew AL (1978) Convergence of an iterative method for derivatives of eigensystems. *J Comput Phys* 26(1):107–112. [https://doi.org/10.1016/0021-9991\(78\)90102-X](https://doi.org/10.1016/0021-9991(78)90102-X)
40. Liu DC, Nocedal J (1989) On the limited memory bfgs method for large scale optimization. *Math Program* 45(3):503–528. <https://doi.org/10.1007/BF01589116>
41. McCallum AK (2002) MALLET: a machine learning for language toolkit. <http://www.cs.umass.edu/~mccallum/mallet>
42. Wulczyn E, Taraborelli D (2017) Wikipedia Clickstream. <https://doi.org/10.6084/m9.figshare.1305770.v22>. https://figshare.com/articles/Wikipedia_Clickstream/1305770

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)